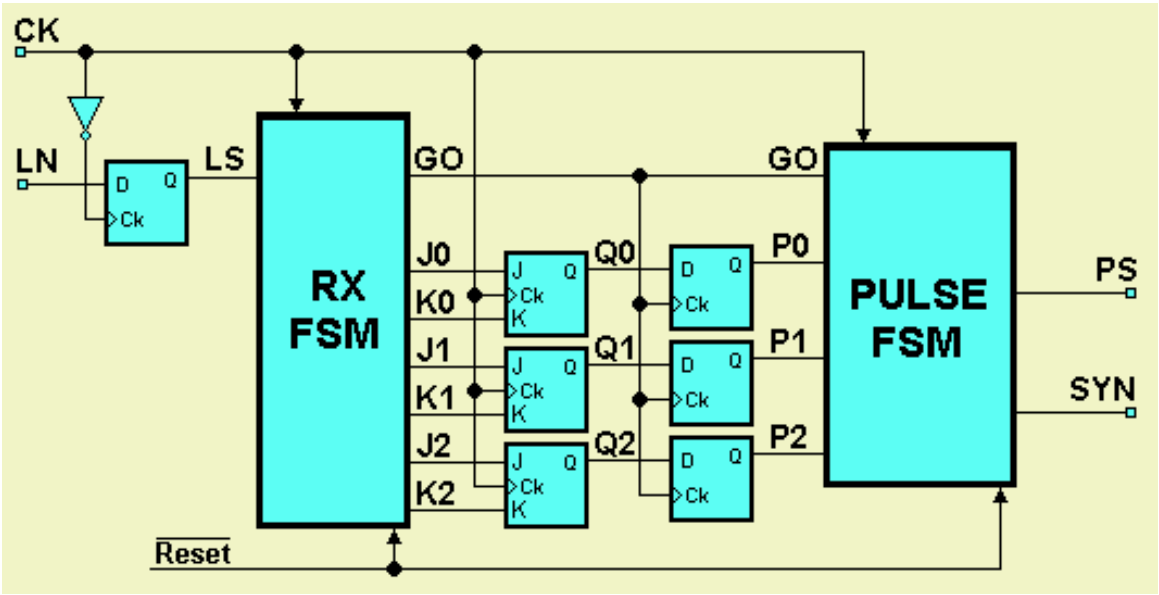
University of Genoa
Dibe Biophysical and Electronic Engineering Department
Electronic Systems and Networking Group esng

Using the *Deeds* Learning Materials



Deeds - Digital Electronics Education and Design Suite

(Feb 2004)

*Edited by **Giuliano Donzellini** and **Domenico Ponta***

Using the Deeds Learning Materials

Deeds - Digital Electronics Education and Design Suite

Index

The Learning Materials **P. 4**

Example 1: interaction between Deeds browsers and d-DcS **P. 9**

Example 2: interaction between Deeds browsers and d-FsM **P. 14**

Example 3: interaction between Deeds browsers and d-McE **P. 21**

Index of Figures

	Page #
Fig. 1: The header of the Deeds Learning Materials page, in the Deeds web site (DIBE, University of Genoa, Italy).	4
Fig. 2: A particular of the index of the Deeds Learning Materials, in the Deeds web site (DIBE, University of Genoa, Italy)	5
Fig. 3: This page contains all the Deeds learning materials, organized as single exercises and classified by topics (DIBE, University of Genoa, Italy)	6
Fig. 4: The page related to the course “Electronic System Design 1 (IN-TL-BM)”, targeted to students of the first year of information and electronics engineering (DIBE, University of Genoa, Italy)	7
Fig. 5: The Assistant opened aside of the main browser, showing a page with a problem assignment	8
Fig. 6: A list of laboratory assignments, opened in the Deeds main browser.	9
Fig. 7: The specific laboratory assignment, opened in the Assistant browser.	10
Fig. 8: The Digital Circuit Simulator, opened by a click on the web page. The circuit template has been automatically downloaded from the courseware site.	11
Fig. 9: The timing simulation of the circuit, once completed by the student.	11
Fig. 10: The student can download the report template to speed up its compilation and delivering.	12
Fig. 11: The report template for this laboratory assignment assignment.	13
Fig. 12: A list of laboratory assignments, with use of d-FsM, opened in the Deeds main browser.	14
Fig. 13a: The specific laboratory assignment, opened in the Assistant browser (first page).	15
Fig. 13b: The specific laboratory assignment, opened in the Assistant browser (second page).	16
Fig. 14: The downloaded ASM diagram, template of the solution.	16
Fig. 15a,b,c: The three pages of the Input/Output dialog window, used to define inputs, outputs and state variables .	17
Fig. 16: The property window, displaying the properties of the ‘a’ state.	17
Fig. 17: The property window, displaying the properties of a condition block.	18
Fig. 18: The finished ASM diagram, and its timing simulation, in the d-FsM.	18
Fig. 19: The finished d-DcS schematic, and the timing simulation of the component, in the d-DcS.	19
Fig. 20: Also in this case, the student will download the report template to speed up its compilation and delivering.	20
Fig. 21: The report template for this laboratory assignment.	20
Fig. 22: A list of laboratory assignments, opened in the Deeds main browser.	21
Fig. 23a: The specific laboratory assignment, opened in the Assistant browser (first part).	22
Fig. 23b: The specific laboratory assignment, opened in the Assistant browser (second part).	23
Fig. 24: The Micro Computer Emulator, opened by a click on the web page. The editor shows the trace of the solution, automatically downloaded from the courseware site.	24
Fig. 25: The DMC8 “architecture”, as shown by the help-system.	25
Fig. 26: An example of the ‘on line’ instruction set documentation: the Arithmetic and Logic instructions.	25
Fig. 27: Another example of the ‘on line’ instruction set documentation: the Shift and Rotate instructions.	26
Fig. 28: The program under test in the interactive debugger of the d-McE: a Warning has be sent to the user.	27
Fig. 29: Port addresses can be modified in the “I/O Ports Address Decoding” dialog window.	28
Fig. 30: Port addresses can be modified by a mouse click on the simulated ‘on board’ dip switches.	28
Fig. 31: The student can download the report template to speed up its compilation and delivering.	28
Fig. 32: The simple template provided on the web page, that the student can download.	29
Fig. 33: A partial view of a ‘final’ student report.	30

The Learning Materials

Learning digital electronics: this is the target of the Deeds environment. Deeds is conceived as a common resource for all introductive courses in digital electronics. Deeds learning material has been developed keeping in mind this target. The learning material that we produced has been tested in various pilot courses, and it is centred on different technical subjects, in the format of exercises and lab assignments, to be delivered at different student levels. In Fig. 1, the main page of the learning material is displayed, opened in the main browser of the Deeds.

An exercise or a lab assignment based on Deeds appears as a set of HTML pages with text and figures. The page aspect and layout are similar to 'normal' web pages. But figures, visual objects and hyperlinks are "actives", because they are connected to the editing and simulation tools of Deeds. For example, let's suppose that an exercise presents a certain digital circuit, visualising its schematics in a picture. When the user clicks on the picture, Deeds launches the corresponding simulator, opening that schematic: then, simulator will allow to "animate" the circuit, i.e. to explore its functionality interactively.

The target of traditional exercises is to help understanding theory, applying it to simple cases and providing a feedback to the teacher through the delivery of the solutions. The role of the developed Deeds learning material is to allow students to check the correctness of the solutions obtained manually and to provide graphical tools for editing solutions, until they are satisfied with their work and can deliver their reports through the network.

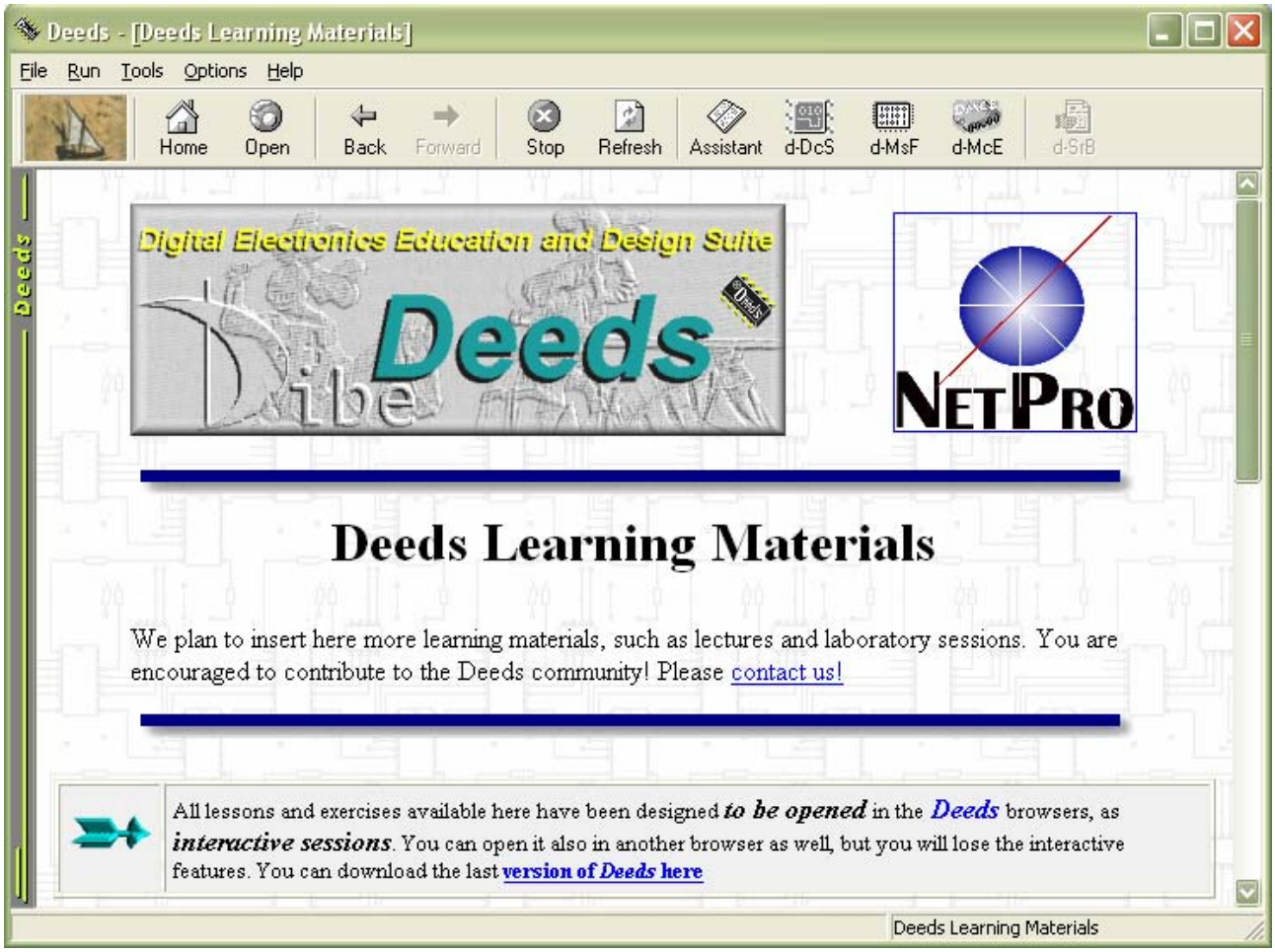


Fig. 1: The header of the Deeds Learning Materials page, in the Deeds web site (DIBE, University of Genoa, Italy).

The Deeds learning material are based also on a different approach to the structure of the exercises. In fact, with the simulators, students may be tempted to skip manual analysis. Exercises, therefore, has been targeted more to the real understanding of the issues than to the execution of repetitive tasks.

Students use Deeds to download the assignments from a web page. A Deeds assignment consists of a functional description and a set of specification of the system that students must design. The approach is meant to replicate the features of a professional environment, within the guidelines suggested by the educational purposes. Project development phases are guided by help and instructions supplied through the Assistant Browser. In Fig. 2 you see the general index of the learning material developed at our site (ESNG, DIBE, University of Genoa, Italy).
The address of the main page of the site is: http://esng.dibe.unige.it/Netpro/Deeds, and it is necessary to click on the “Learning Materials” link, on the left of the page.

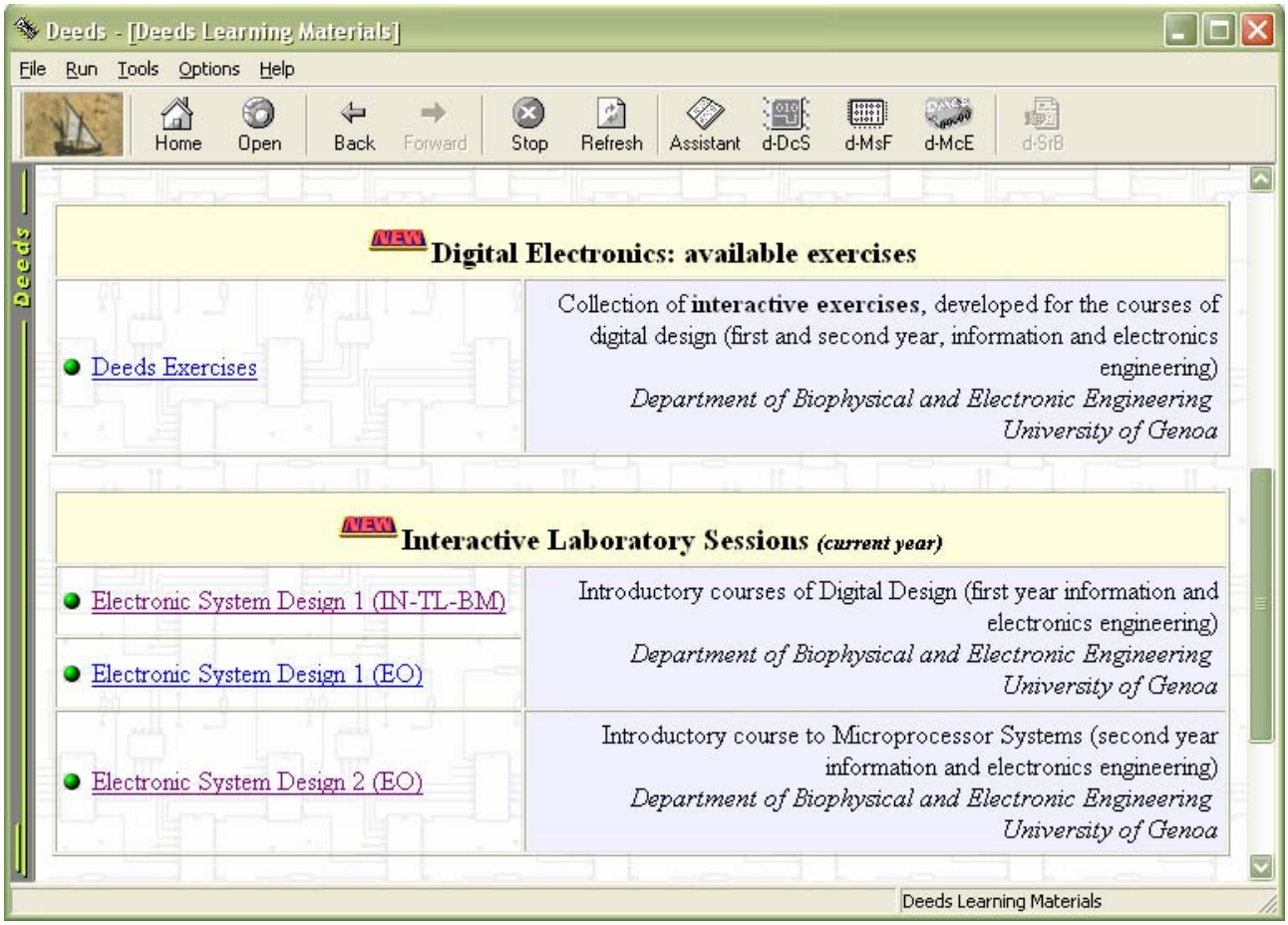


Fig. 2: A particular of the index of the Deeds Learning Materials, in the Deeds web site (DIBE, University of Genoa, Italy).

In the general index some links are provided. The first link visible in Fig. 2 (“Deeds Exercises”) points to a page that contains all the interactive material, under the form of Deeds exercises, classified by topics (see Fig. 3).

The other links refer to the specific NetPro pilot courses, organized at DIBE (Department of Biophysical and Electronic Engineering, University of Genoa).

In Fig. 4 you can see, as an example, a view of the page related to the course “Electronic System Design 1 (IN-TL-BM)”, targeted to students of the first year of information and electronics engineering.

Fig. 3: This page contains all the Deeds learning materials, organized as single exercises and classified by topics (DIBE, University of Genoa, Italy).

The material is classified by topic. Each exercise is available in two ways.

The hyperlinks highlighted on the left point directly to each Deeds exercise, to use them ‘on line’. This is the preferred mode, when students works in a institution laboratory, constantly connected to the global network.

The links evidenced with a number, on the right, instead, point to zipped files, suitable to be downloaded from home or another location where the connection to the network is realized by a modem. In this case, the student downloads the file, then un-compresses locally it and, finally, opens the respective ‘Index’ page in the Deeds.

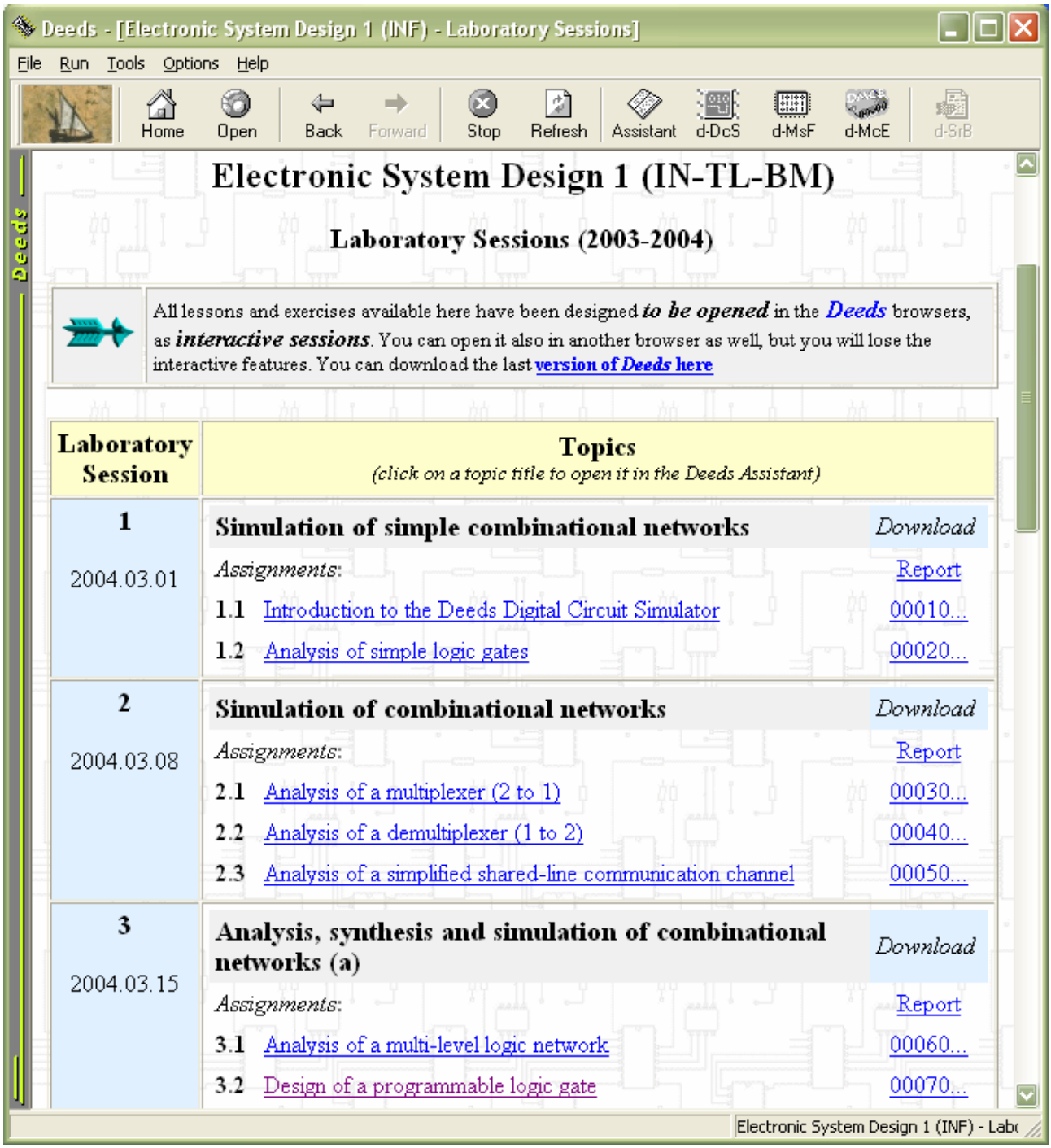


Fig. 4: The page related to the course "Electronic System Design 1 (IN-TL-BM)", targeted to students of the first year of information and electronics engineering (DIBE, University of Genoa, Italy).

In this page (Fig. 4), the material is ordered by laboratory assignment. Each assignment includes links to the exercises that the student must solve, and another link, used to download a "report template", alias a format of the file used to write the solutions and deliver them to the teacher, through the network.

As in the previous page, the material is available in two ways: the first to use it 'on line' (links on the left), and the second to download it from distance.

When the student click on a link (chosen among those on the left), the Deeds activates the secondary browser: the "Assistant".

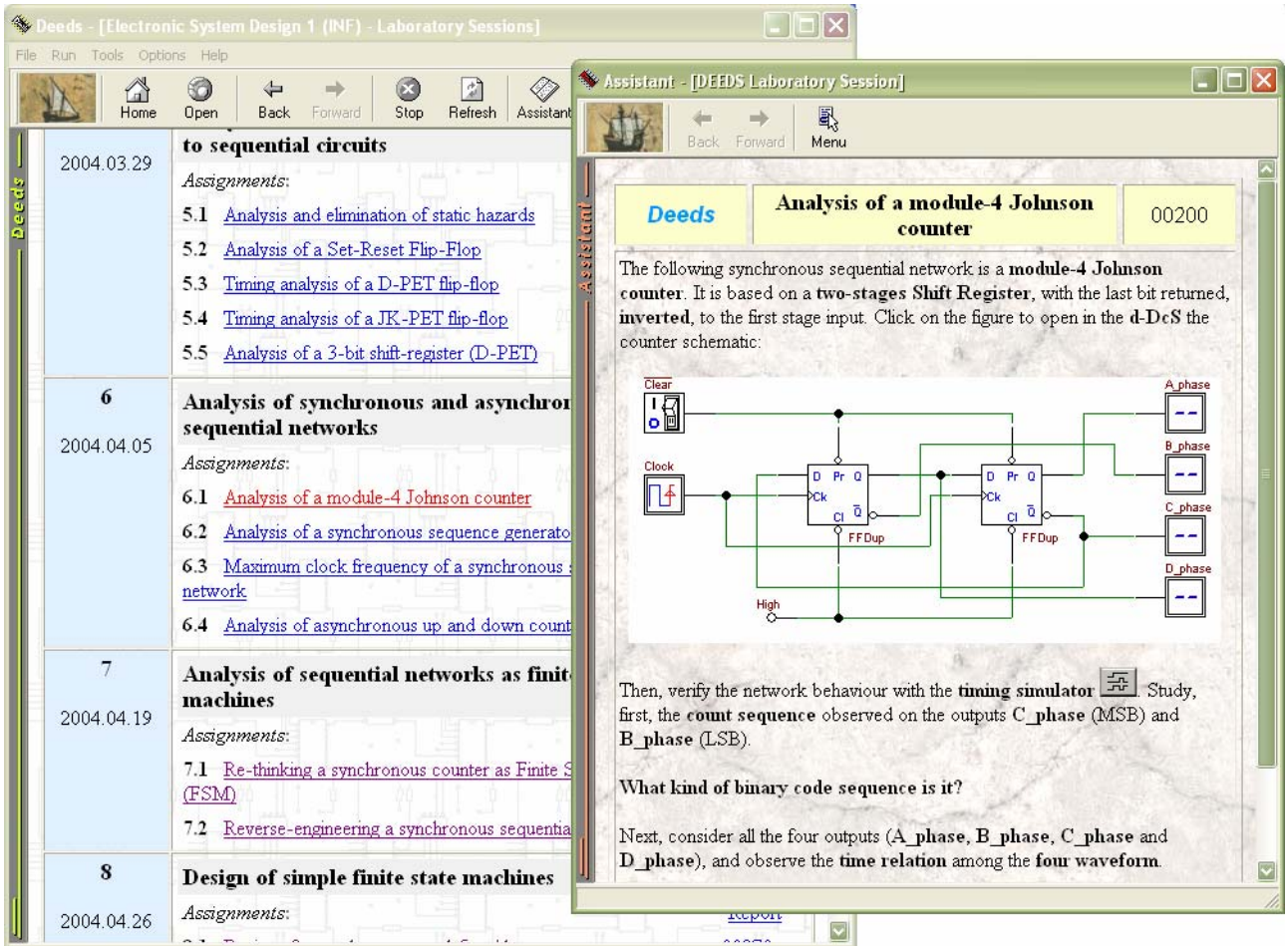


Fig. 5: The Assistant opened aside of the main browser, showing a page with a problem assignment.

The “Assistant” browser has characteristics similar to those of the main browser, but it is specialized to assist students, side by side, in their work (Fig. 5). This is the browser used to open lessons, exercises and laboratory assignments.

In the following, some example about the usage of the learning material is proposed.

Example 1: interaction between Deeds browsers and d-DcS

In Fig. 6, a list of assignments is opened in the Deeds main browser. Suppose that the student has to attend the assignment # 2.1: *"Analysis of a demultiplexer (1 to 2)",* from the "Electronic System Design 1" NetPro pilot course.

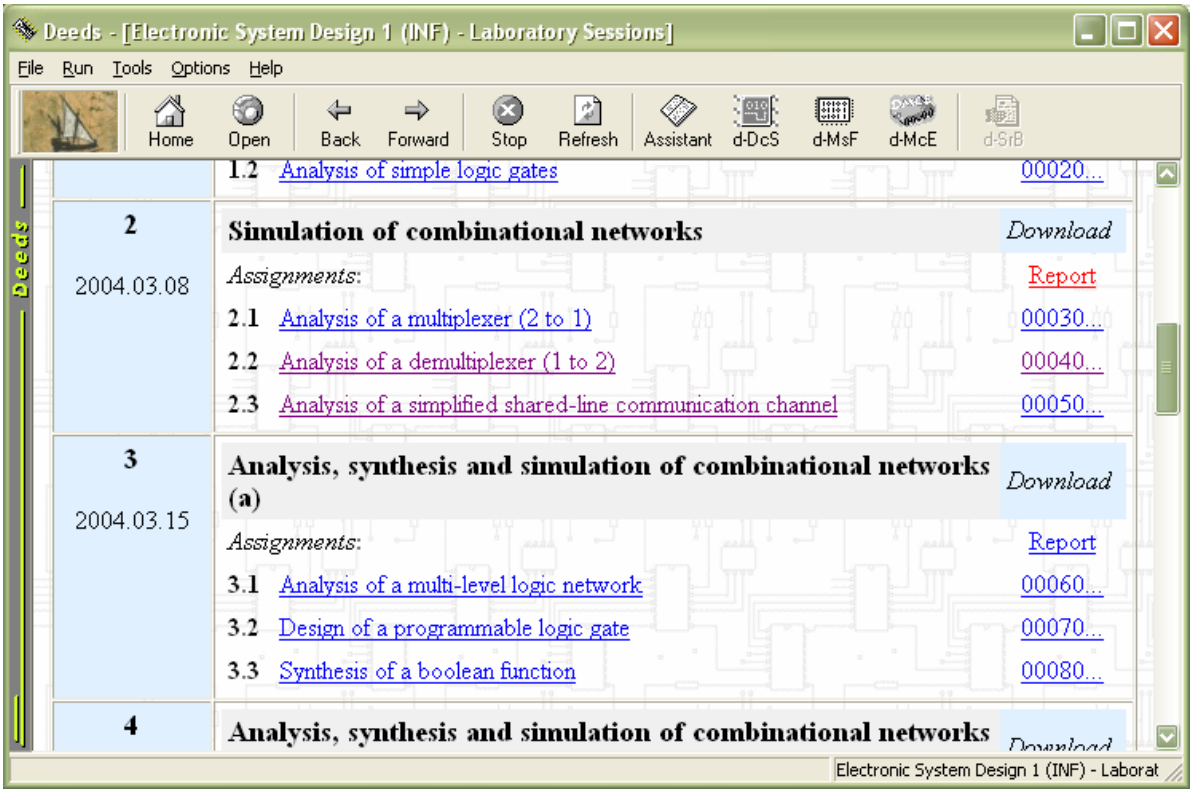


Fig. 6: A list of laboratory assignments, opened in the Deeds main browser.

Than, he or she clicks on the link, and the assignment will open in the Assistant (see Fig. 7).

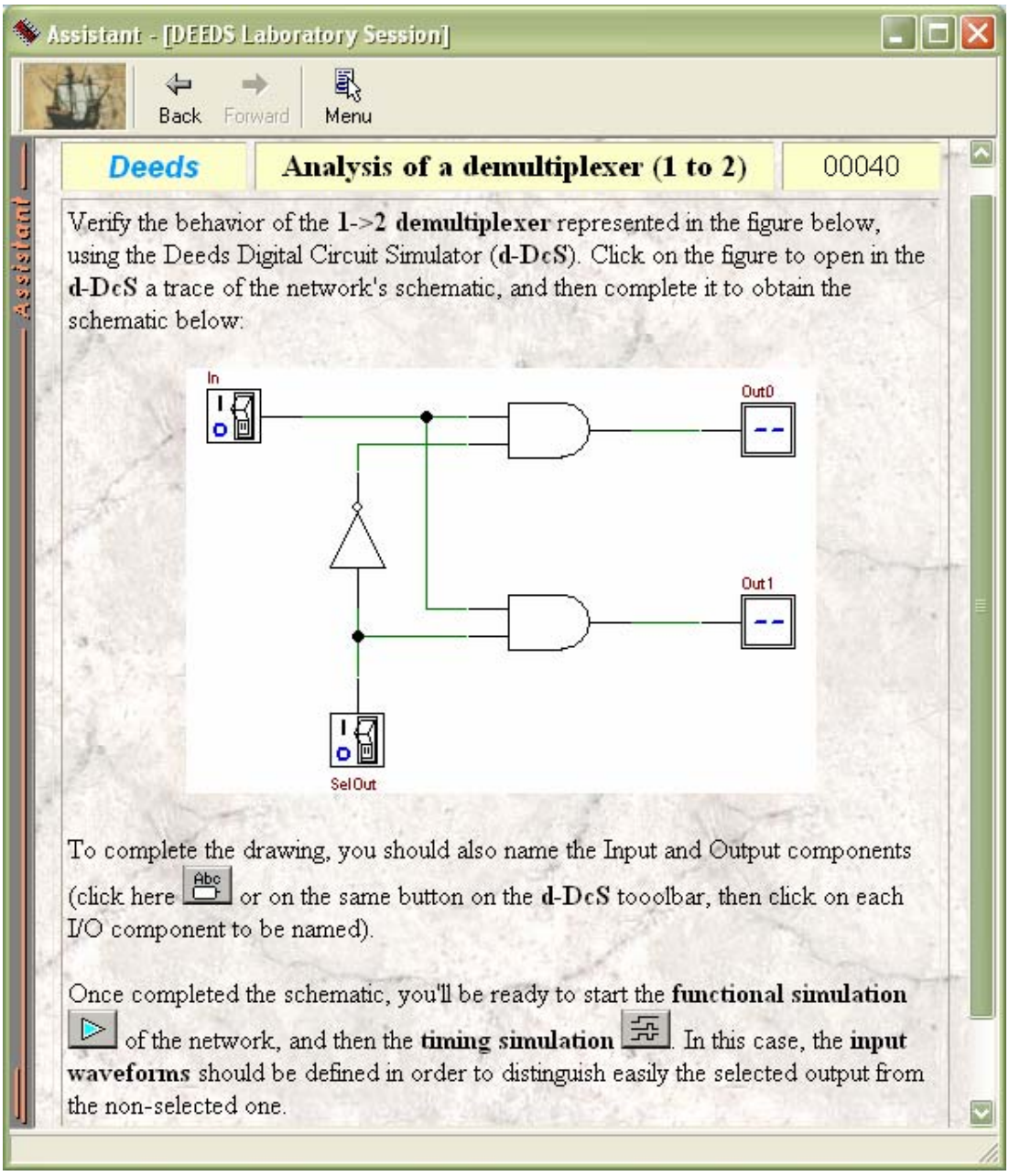


Fig. 7: The specific laboratory assignment, opened in the Assistant browser.

The assignment asks the user to verify the behavior of the 1->2 demultiplexer represented in the figure, using the Deeds Digital Circuit Simulator). The text suggests to click on the figure to open in the d-DcS a trace of the network's schematic, and then to complete it.

In this example, you see that it is necessary only a simple click on the figure to activate the simulator and to download from the web site a ‘template’ of the solution. This approach aims to simplify user operation, avoiding to spend time in no useful and distracting tasks.

The user will see the Digital Circuit Simulator, and the file downloaded in it, as in Fig. 8.

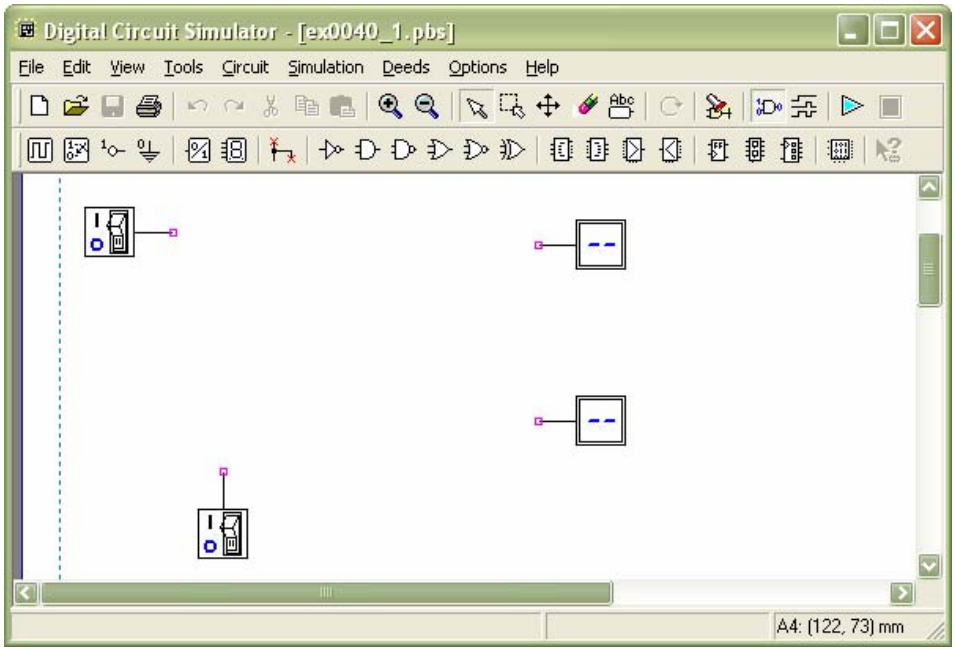


Fig. 8: The Digital Circuit Simulator, opened by a click on the web page. The circuit template has been automatically downloaded from the courseware site.

The assignment suggests now to complete the drawing, and also to activate a few useful simulator commands directly from the web page, with a simple click.

Once completed the schematic, also the simulation can be started, directly from the Deeds web page. In Fig. 9 you can see the results expected from the student work.

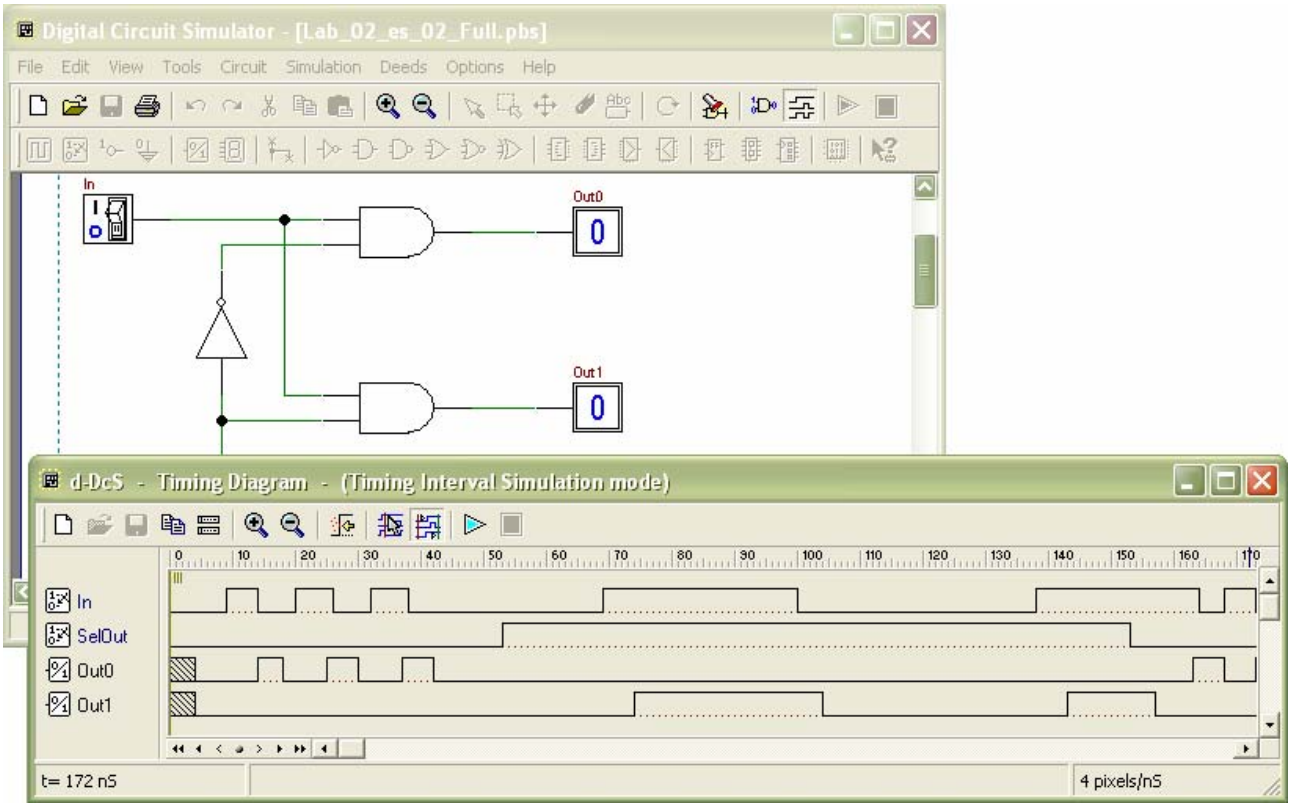


Fig. 9: The timing simulation of the circuit, once completed by the student.

Now is the time for the student to compile and deliver a good report. In the Deeds assignment page, a link is prepared to download and edit a report template file (Fig. 10).

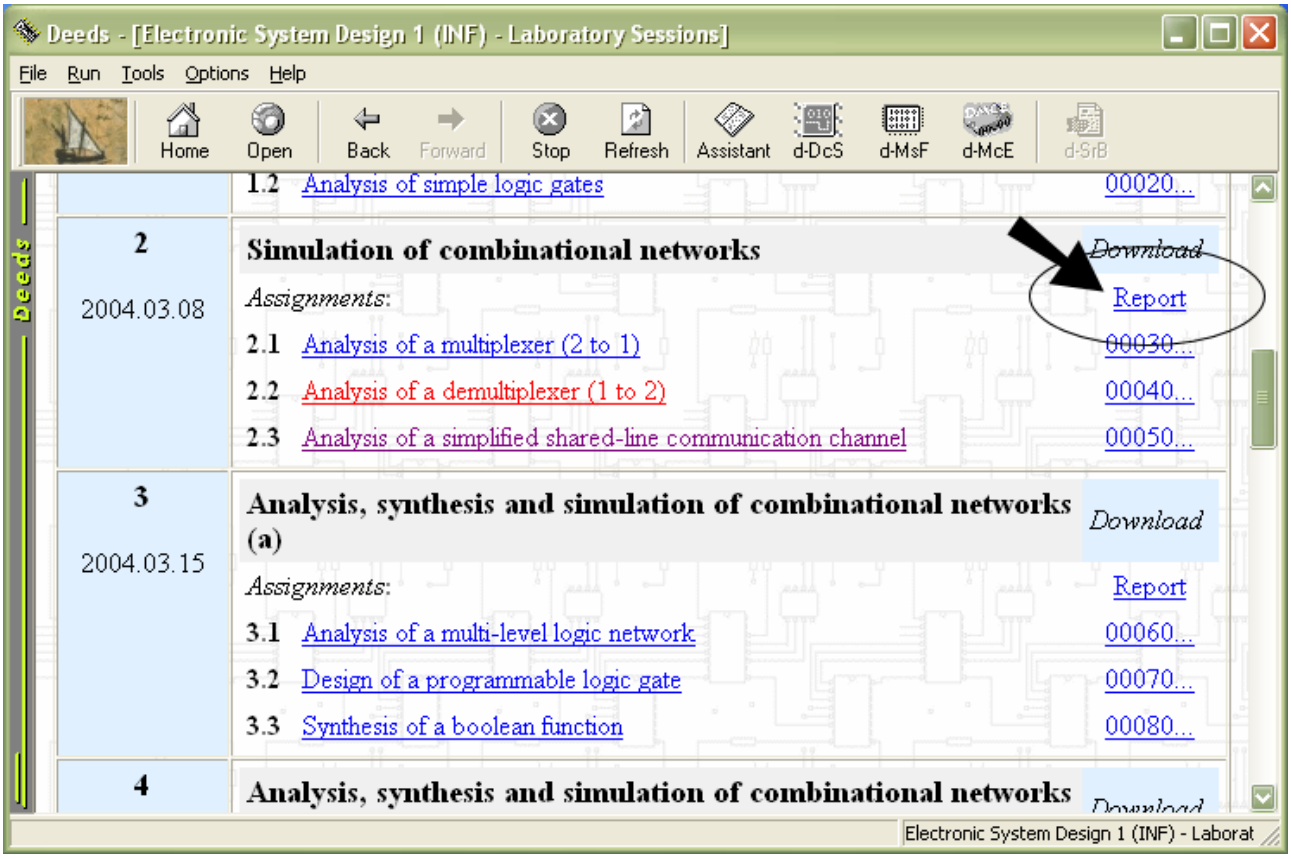


Fig. 10: The student can download the report template to speed up its compilation and delivering.

This has been previewed to uniform the report styles, making easier the teacher task, especially when the number of student is valuable. But the availability of a report template is very useful also to the student, because it saves a lot of time, speeding up the student work and leaving more time to concentrate on the arguments to learn.

This is the report template for this laboratory assignment (Fig. 11).

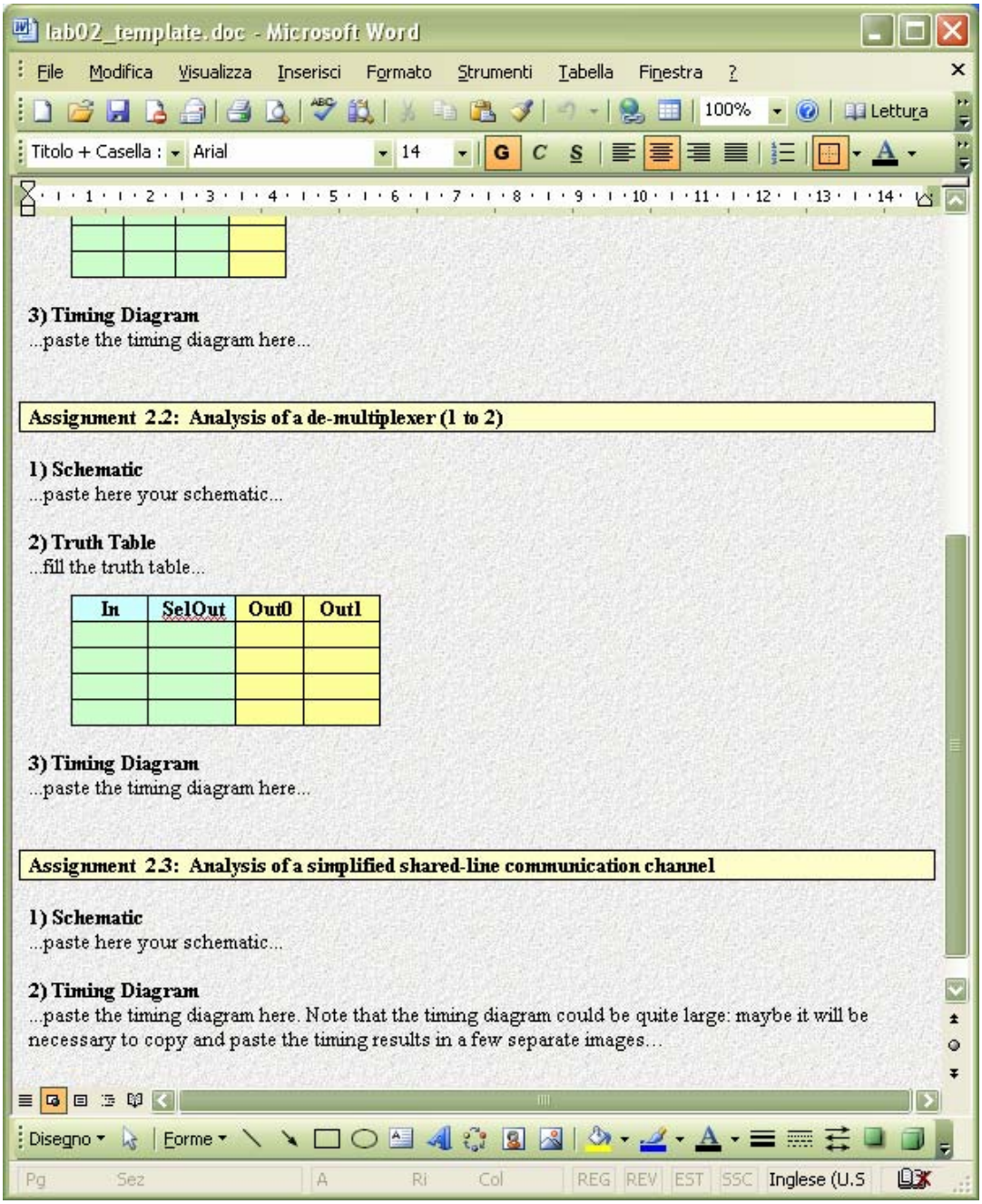


Fig. 11: The report template for this laboratory assignment assignment.

Example 2: interaction between Deeds browsers and d-FsM

As in the example applied to the Deeds with the d-DcS, in Fig. 12, a list of laboratory assignments is opened in the Deeds main browser, from the “Electronic System Design 1” NetPro pilot course.

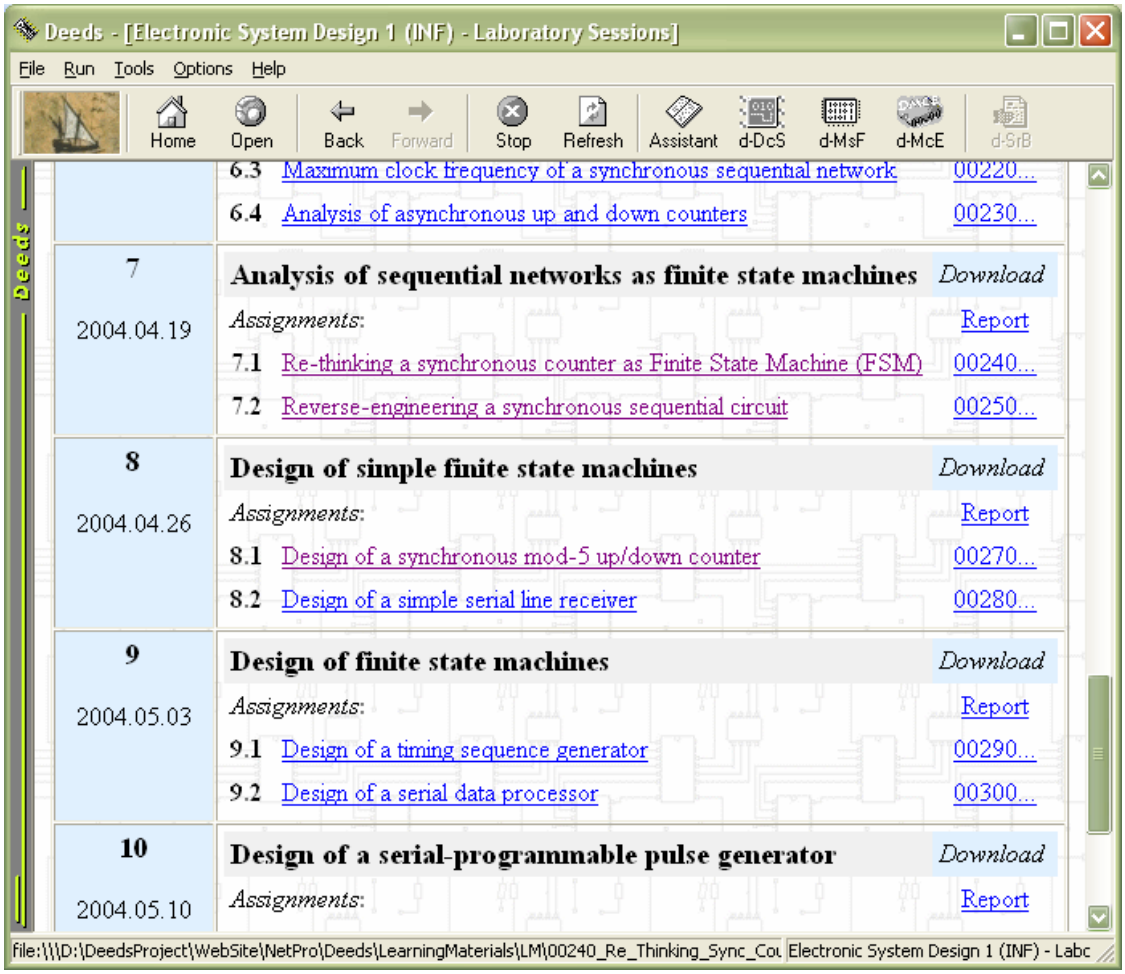


Fig. 12: A list of laboratory assignments, with use of d-FsM, opened in the Deeds main browser.

The student executes the assignment # 8.1: *“Design of a synchronous mod-5 up/down counter”*. As in the example related to the d-DcS, with a click of the user on the link, the specific assignment will be opened in the Assistant (Fig. 13a and 13b).

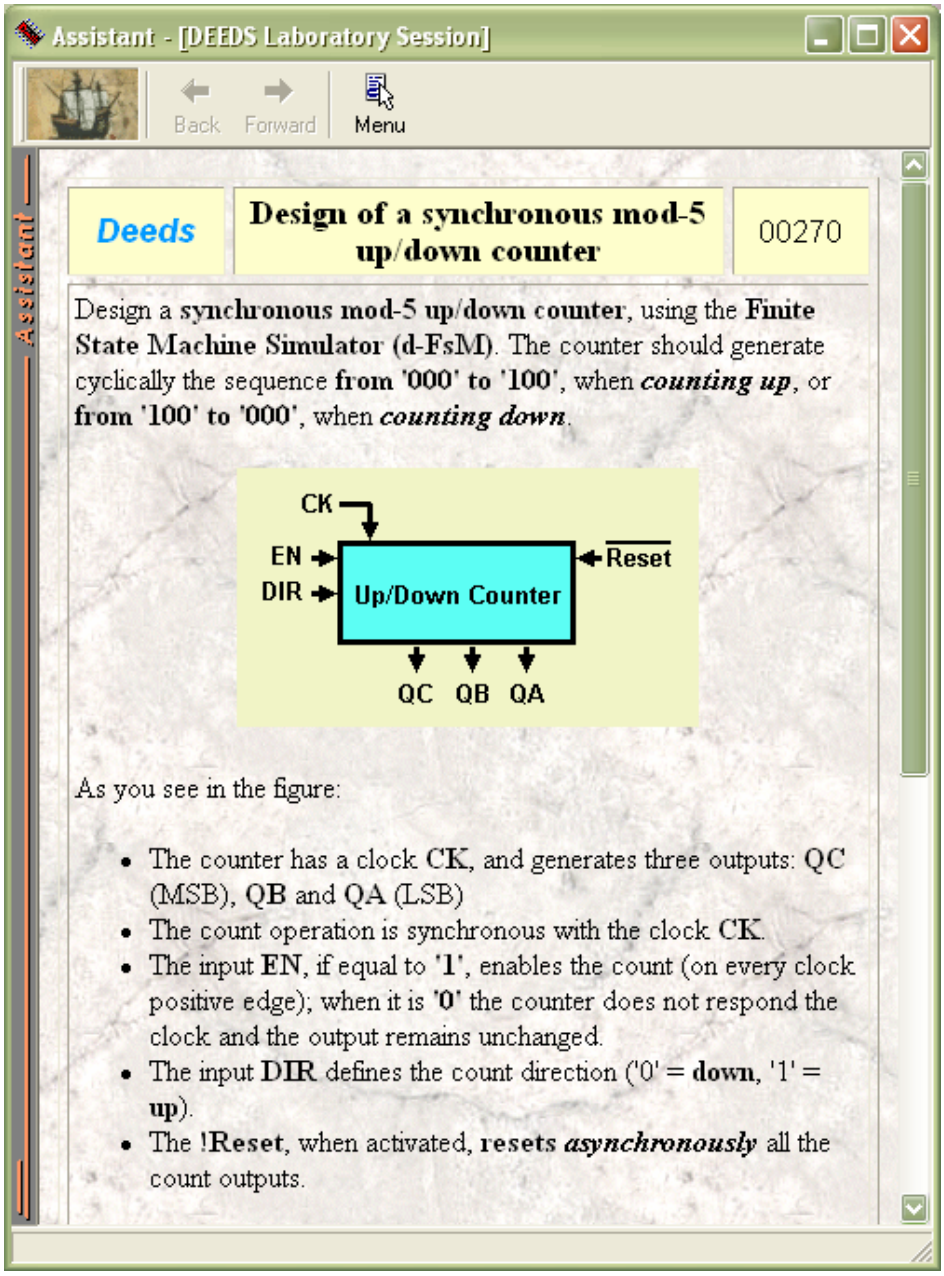


Fig. 13a: The specific laboratory assignment, opened in the Assistant browser (first page).

The assignment asks the user to design a synchronous mod-5 up/down counter, using the Finite State Machine Simulator.

In the laboratory assignment (Fig. 13a) is explained that the counter should generate a numerical sequence on the outputs QC, QB and QA, depending from the line input EN and DIR. The counter is synchronous with the clock CK and it is initialized by an asynchronous Reset input. In particular, the input DIR defines the count direction (up or down), and the input EN enables the count operation, that will take place on every clock positive edge.

In Fig. 13b, the assignment continues with a suggestion: to download an ASM diagram template, to be guided toward the solution. If the student use this option, he or she could concentrate better on the argument, instead of build from scratch the solution, bothering with the simulator details and spending time in less useful and distracting tasks. The option is not mandatory, however, and the student can freely activate the simulator without using the template.

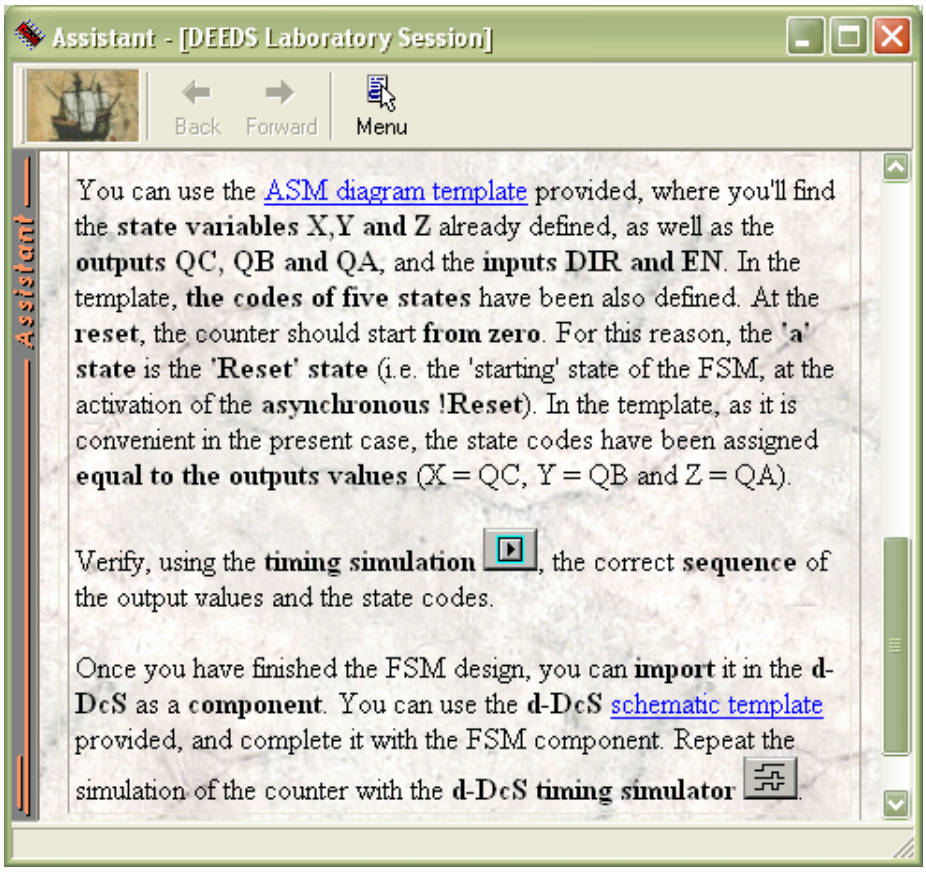


Fig. 13b: The specific laboratory assignment, opened in the Assistant browser (second page).

To download the template, it is necessary only a simple click on the link in the text. The d-FsM will be activated, and the file downloaded from the web site, automatically. In Fig. 14 you see the suggested template, as downloaded in the simulator.

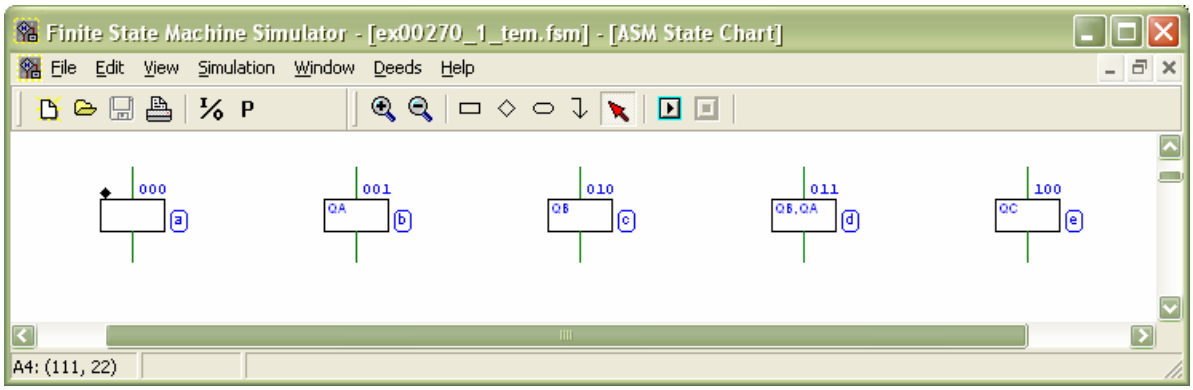


Fig. 14: The downloaded ASM diagram, template of the solution.

In the template, as the text of the assignment explains, the student will find some important definition already set: the state variables X,Y,Z, the outputs QC, QB, QA and the inputs DIR and EN. The necessary five state blocks are already drawn.

In Fig. 15a,b,c are displayed the pre-defined properties, as they appear in the Input/Output dialog windows, that the user activates with the tool bar command  .

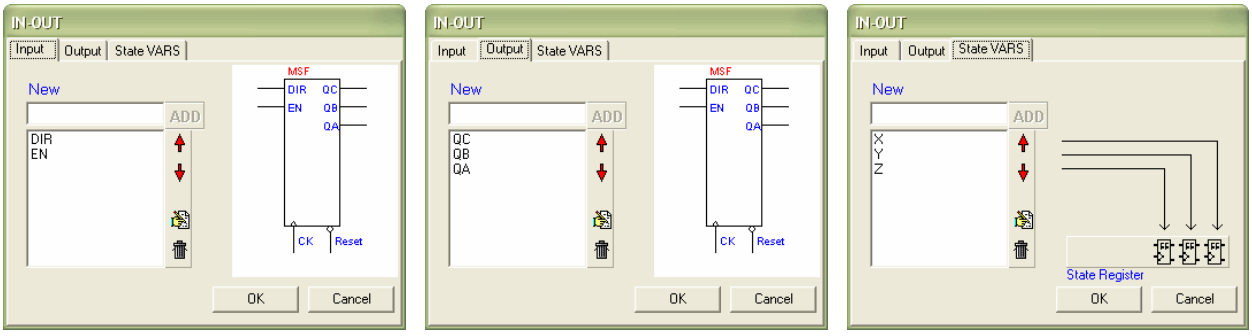


Fig. 15a,b,c: The three pages of the Input/Output dialog window, used to define inputs, outputs and state variables .

Note that the specification requires that the 'a' state will be the 'Reset' state, i.e. the 'starting' state of the component at the activation of the asynchronous !Reset. Also this characteristic has been pre-defined in the template, as the ‘a’ state appears in the drawing with a *little diamond* placed on it.

Actually, all the states properties have been pre-defined in the template. The user can modify this properties opening the Property Window. This can be left aside to the editor, during the operations (to open it, press the tool bar button ). In Fig. 16 you see the Property Window, as it appears when the user select the ‘a’ state block (with a mouse click on it).

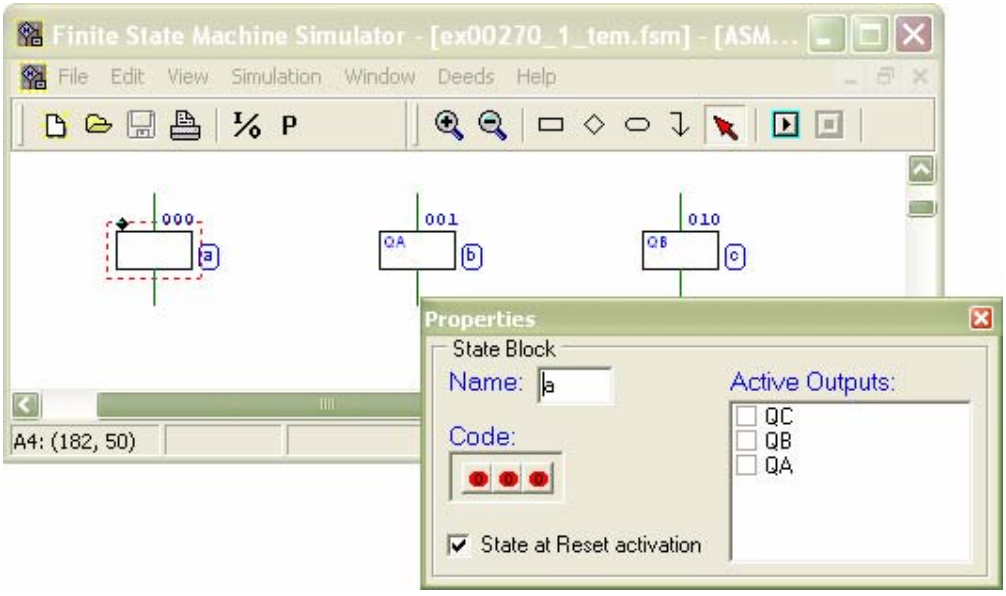


Fig. 16: The property window, displaying the properties of the ‘a’ state.

For a state block, the user can set or change the symbolic name (‘a’ in the present case), the state code (‘000’, here), and the active outputs (none, in the example). The check box on the left imposes this one as ‘Reset State’.

The user is asked to complete the ASM diagram and, using the timing simulation integrated in the d-FsM, to verify the correct sequence of output values and state codes. The user will start drawing, adding path lines and diamonds, as required by the requested functionality.

In Fig. 17 you see the Property Window, as it appears when the user select a condition block. The user can change the orientation of the diamond connections and the condition, chosen among the input variables ('DIR' in this example).

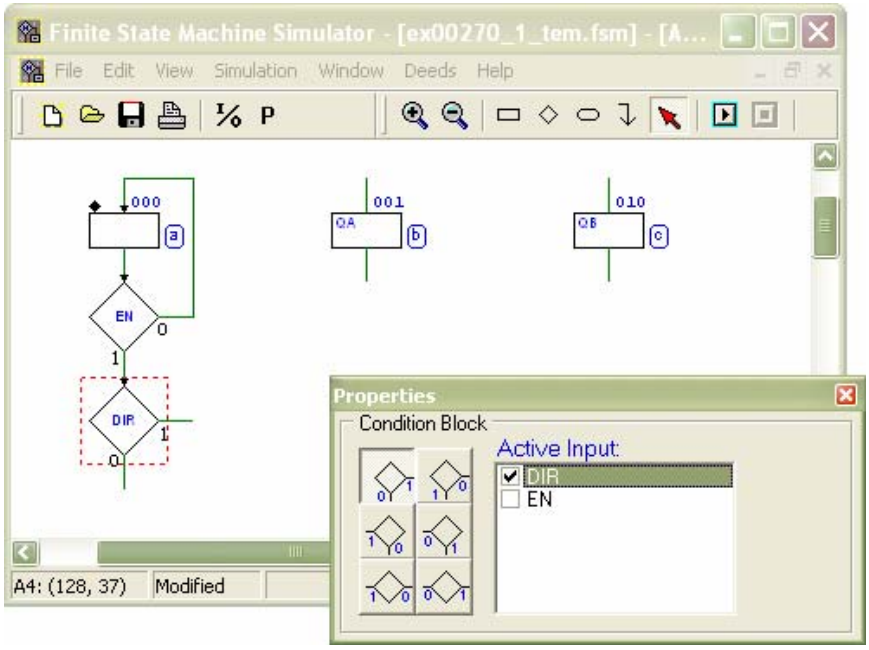


Fig. 17: The property window, displaying the properties of a condition block.

Once the student have finished the design, the next step required is to verify the behaviour of the counter with the timing simulator of the d-FsM itself (Fig. 18).

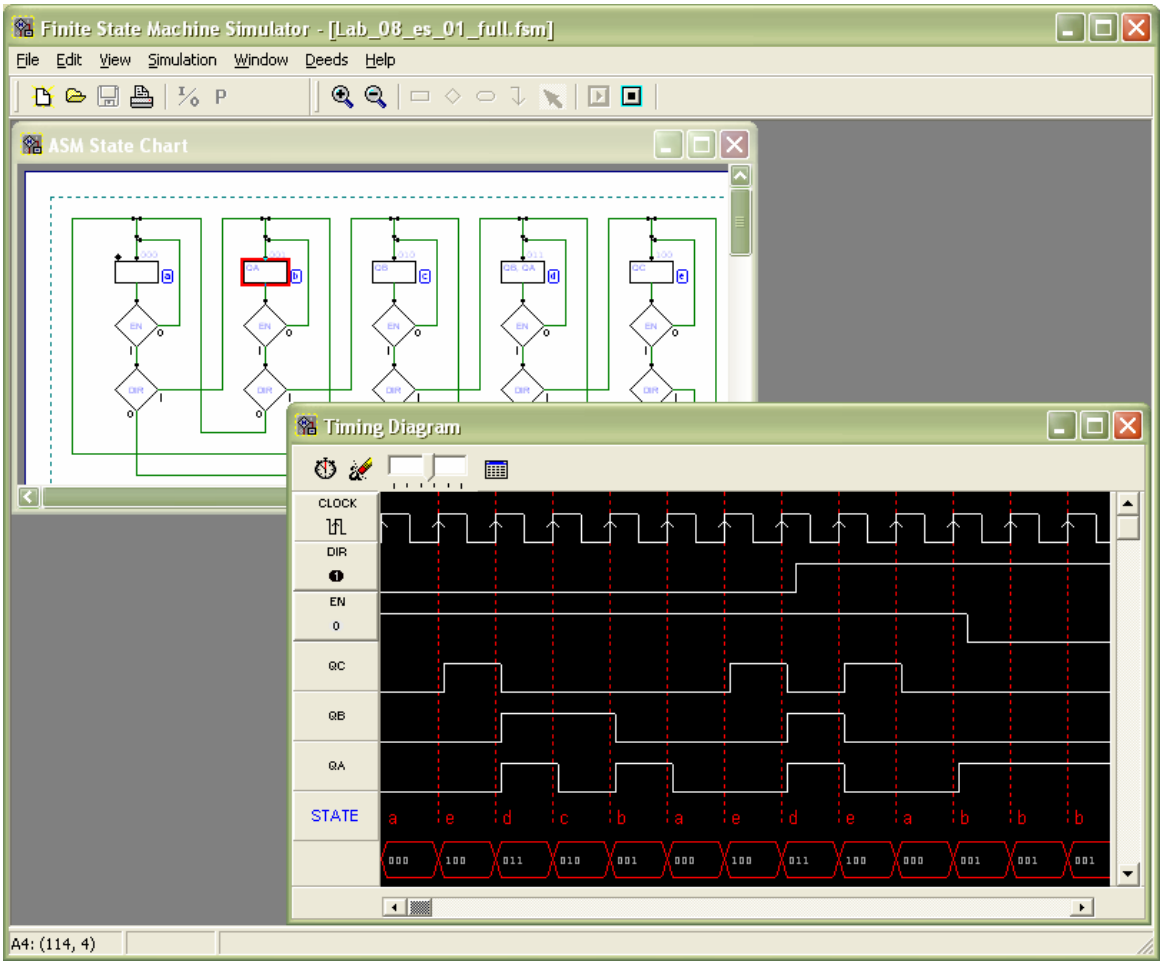


Fig. 18: The finished ASM diagram, and its timing simulation, in the d-FsM.

When the user clicks on the ‘Clock’ button, the internal simulator evaluates next state and outputs (according to the current input values) and displays the results on the time diagram.

At the same time, in the editor window, the corresponding new state is *highlighted* (with a coloured frame around it, see Fig. 18). This is an important feature, because a major difficulty, for a beginner, is to understand the correspondence between states and events time sequence.

Finally, when the behaviour of the component satisfies all the required specifications, the component could be imported in the d-DcS (see the assignment, Fig. 13b). Also in this case, a simple d-DcS schematic template is provided, to speed up the operations; it can be easy downloaded and opened in the d-DcS with a click on the hyperlink in the text. Once completed the schematic, the simulation of the counter could be repeated in the d-DcS timing simulator (Fig. 19).

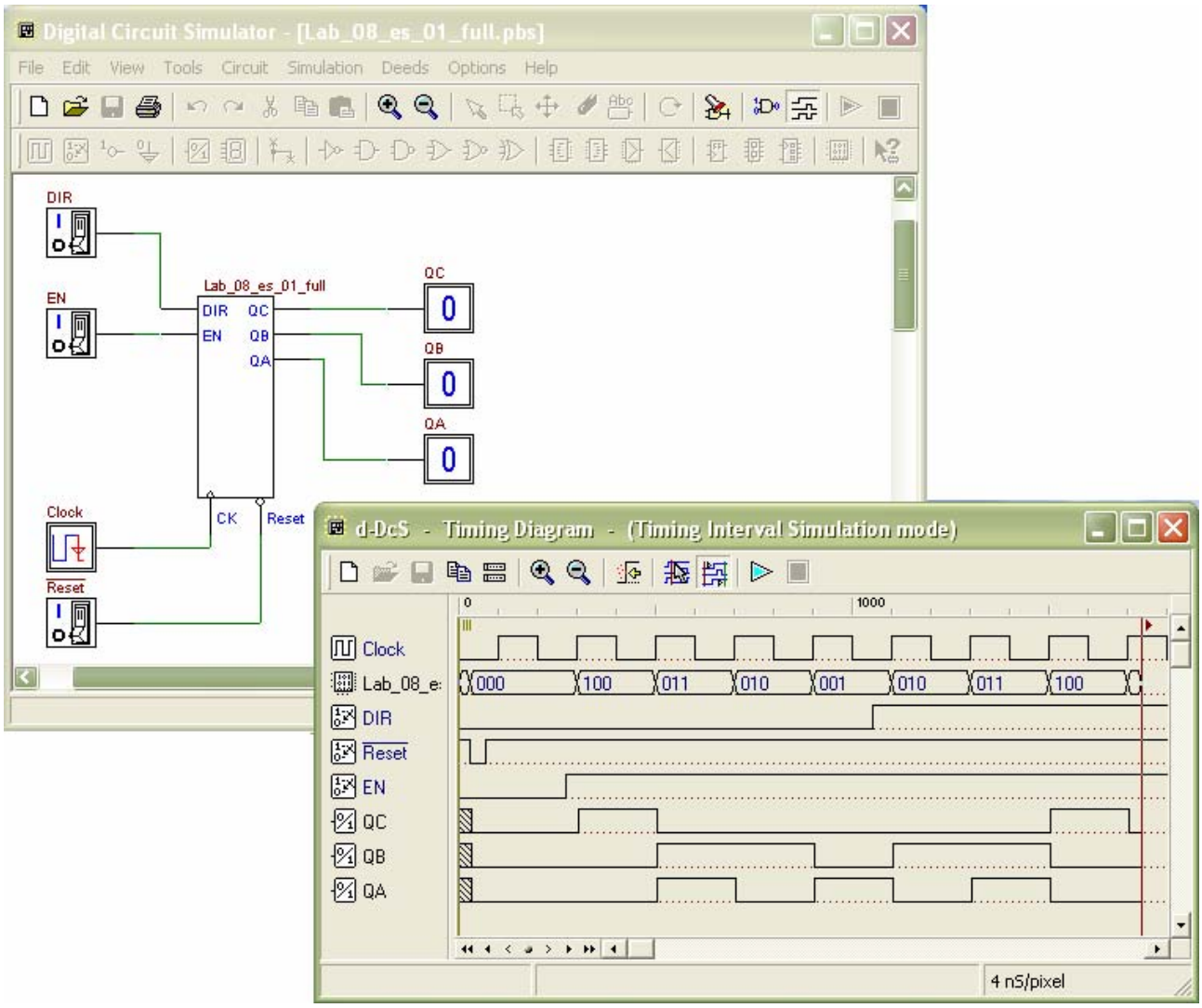


Fig. 19: The finished d-DcS schematic, and the timing simulation of the component, in the d-DcS.

As in the example related to the d-DcS, at this point the student will compile and deliver a report about its work. As already seen, in the assignments page, a link is set to download a report template file (Fig. 20).

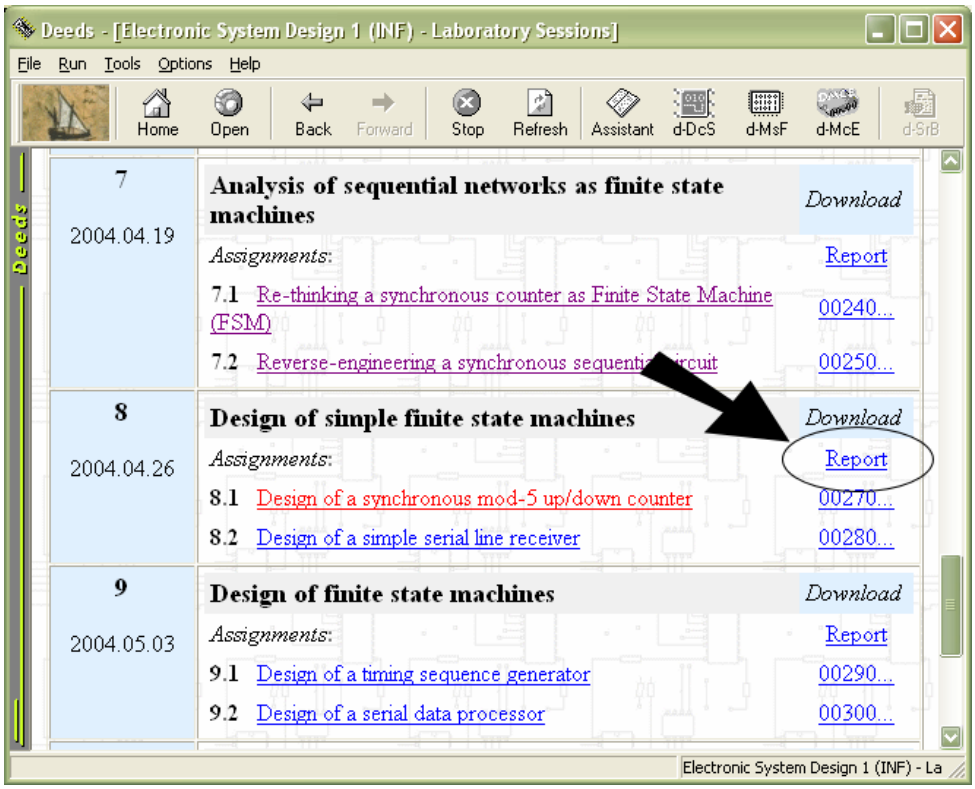


Fig. 20: Also in this case, the student will download the report template to speed up its compilation and delivering.

In Fig. 21 is displayed the report template prepared for this laboratory assignment, downloaded and ready to be edited.

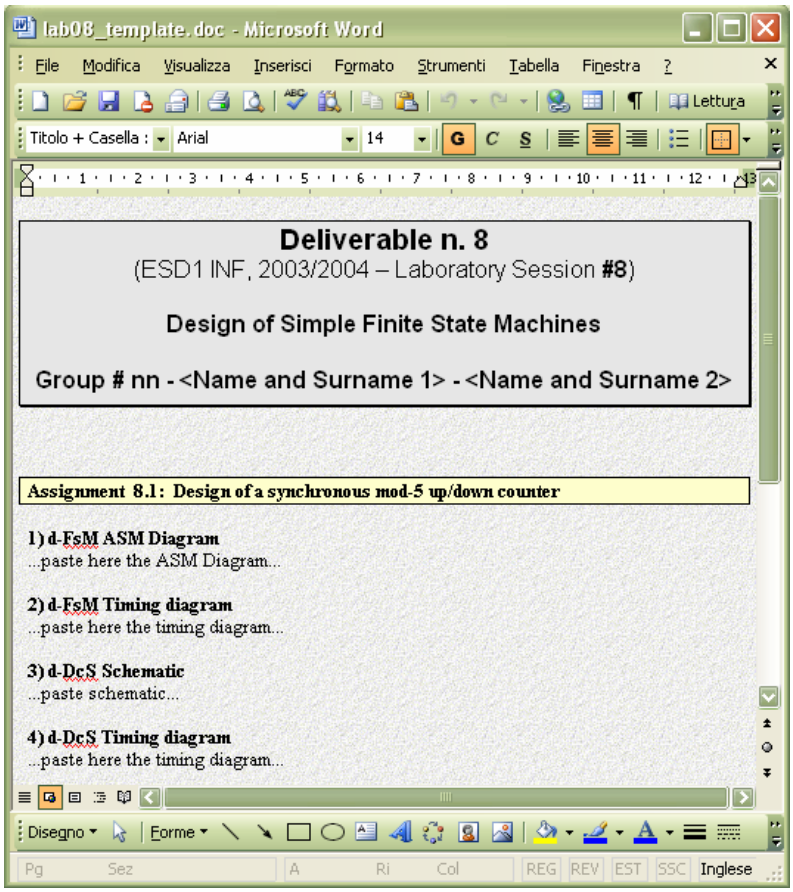


Fig. 21: The report template for this laboratory assignment.

Example 3: interaction between Deeds browsers and d-McE

In Fig. 22 a list of laboratory assignments is opened in the Deeds main browser. The student has to attend the assignment # 4.1: *"Asynchronous serial communication"* (from the course on microcomputer: "Electronic System Design 2" NetPro pilot course).

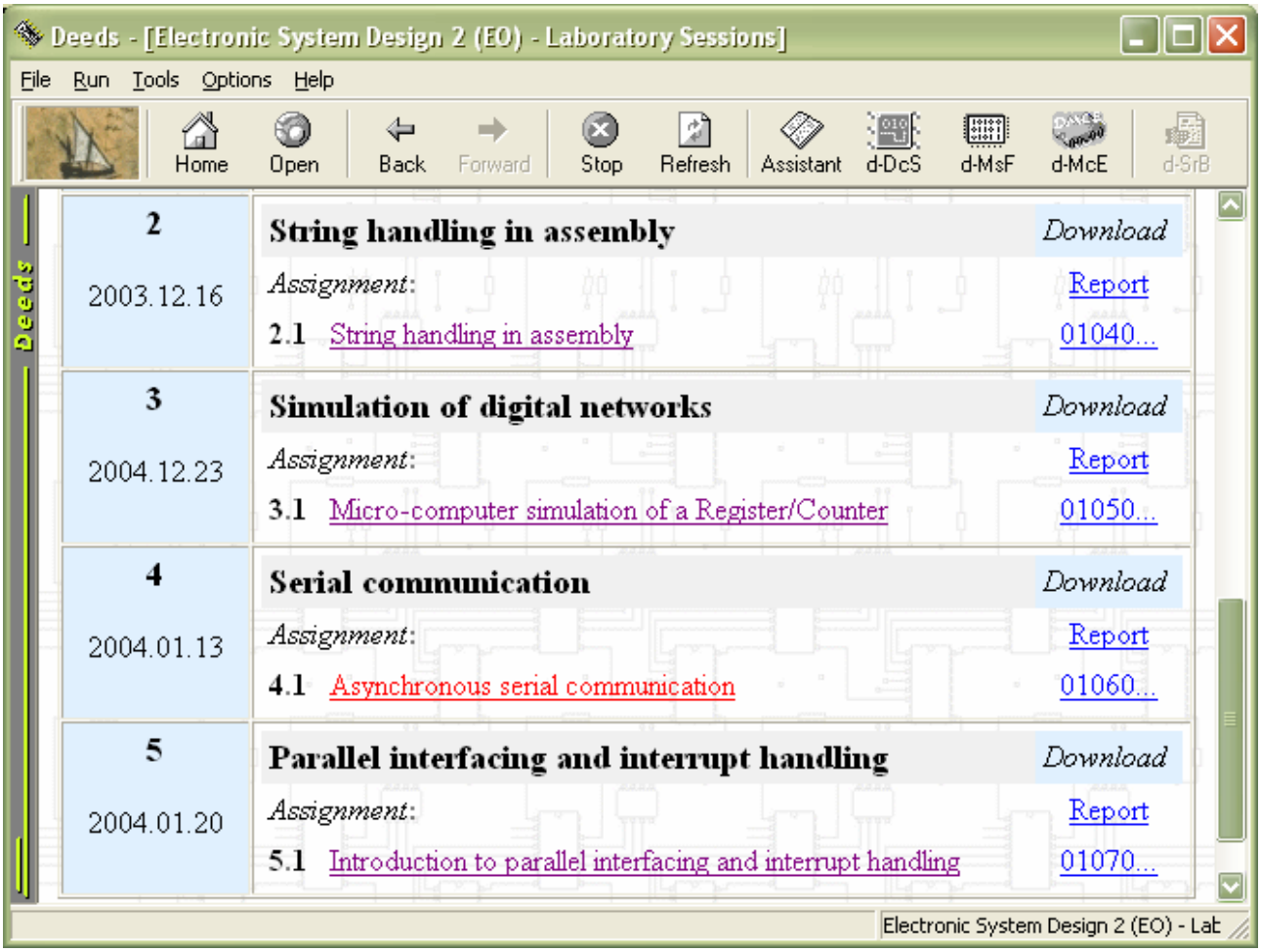


Fig. 22: A list of laboratory assignments, opened in the Deeds main browser.

With a click on the link, the assignment will open in the Assistant (see Fig. 23a and 23b).

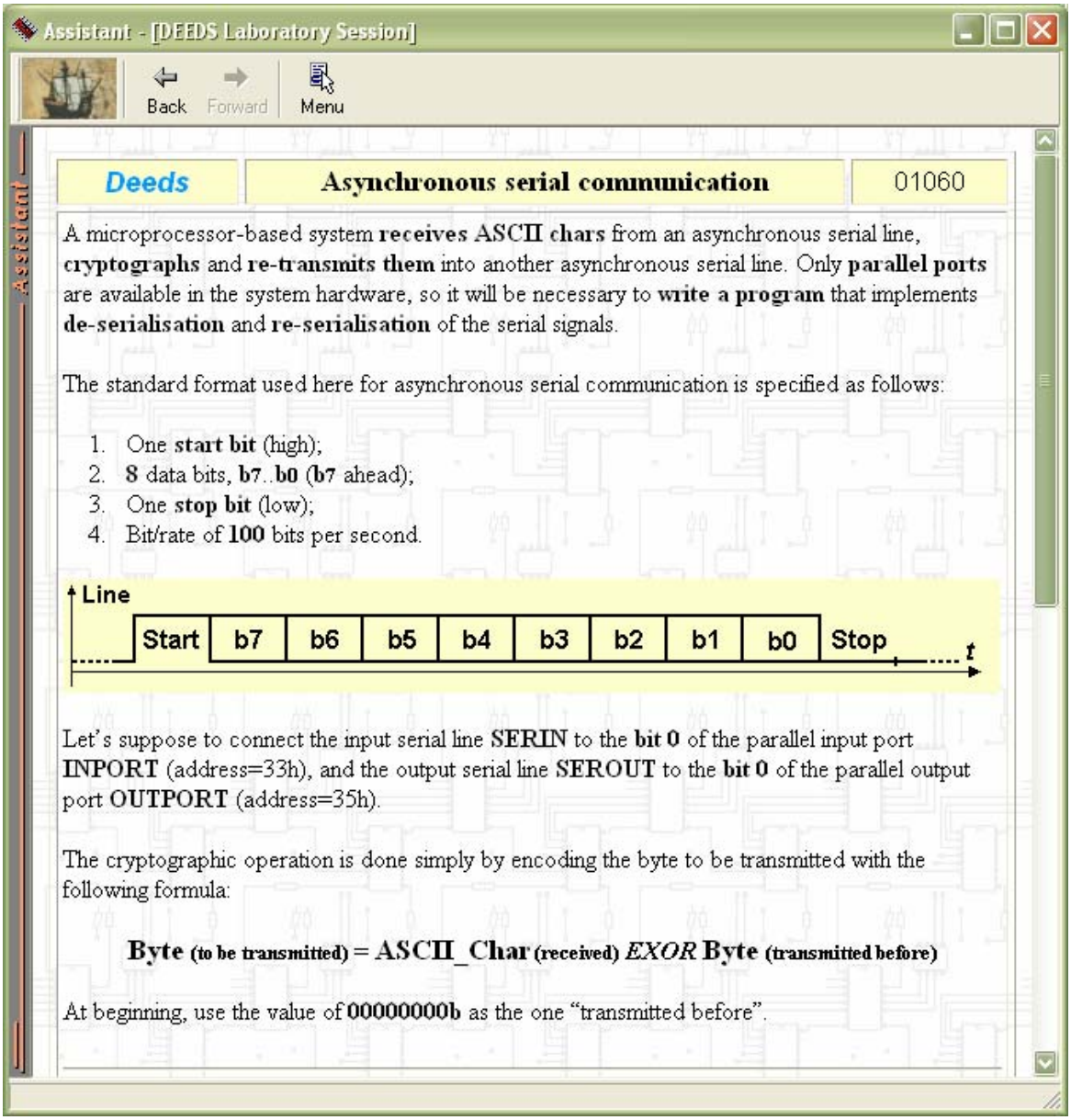


Fig. 23a: The specific laboratory assignment, opened in the Assistant browser (first part).

In this assignment (Fig. 23a), we require to the student to write a program to receive and retransmit serial asynchronous information, using the parallel ports available in the d-McE. The program should take in charge the operation of de-serializing and serializing data. Also a simple cryptographic method is applied to data before retransmitting it.

In the assignment is described the format of the serial data packet (standard 8 bit asynchronous serial communication, without parity control). That protocol previews one start bit at '1', eight data bits b7..b0 (b7 ahead), one stop bit at '0'. It is defined a low bit rate (100 bits per second), with the aim to let the user concentrate on the basic tasks, without bothering too attention to timing problems.

The text continues suggesting to connect the input and output serial lines to specific bits of the available input and output ports (INPORT and OUTPORT).

The simple cryptographic operation requires that the program remember the previous transmitted byte and combine it in a byte-wise EXOR operation with the currently received one.

Fig. 23b: The specific laboratory assignment, opened in the Assistant browser (second part).

The theme continue with the guidelines for a possible solution, as the student, at the moment of this laboratory session, faces this kind of problems for the first time (Fig. 23b).

The Deeds let to get a trace of the solution, with a simple click on the specific link. It will be automatically downloaded and opened in the source code editor of the d-McE (Fig. 24). As usual, this approach let the user simplify the operations necessary to start with the 'true' work.

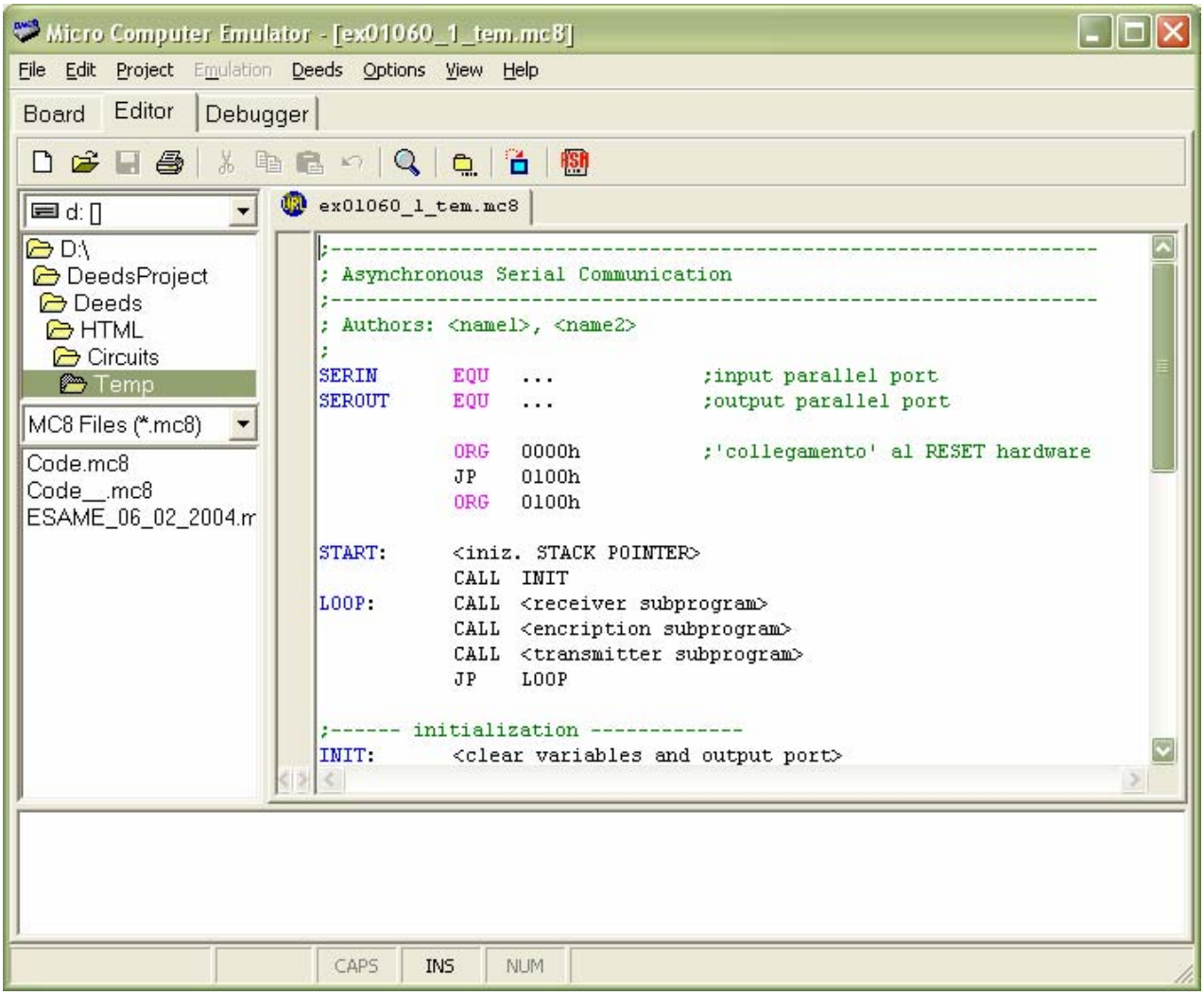


Fig. 24: The Micro Computer Emulator, opened by a click on the web page. The editor shows the trace of the solution, automatically downloaded from the courseware site.

Note the icon visible on top of the editor page: . In this case the symbol indicates that the file has been downloaded from the web. When the user will save it on the local disk, this little icon will change in .

The user can take advantage from the help system, that documents the architectural aspects of the DMC8 microprocessor, and its instruction set (see Fig. 25, 26, 27).

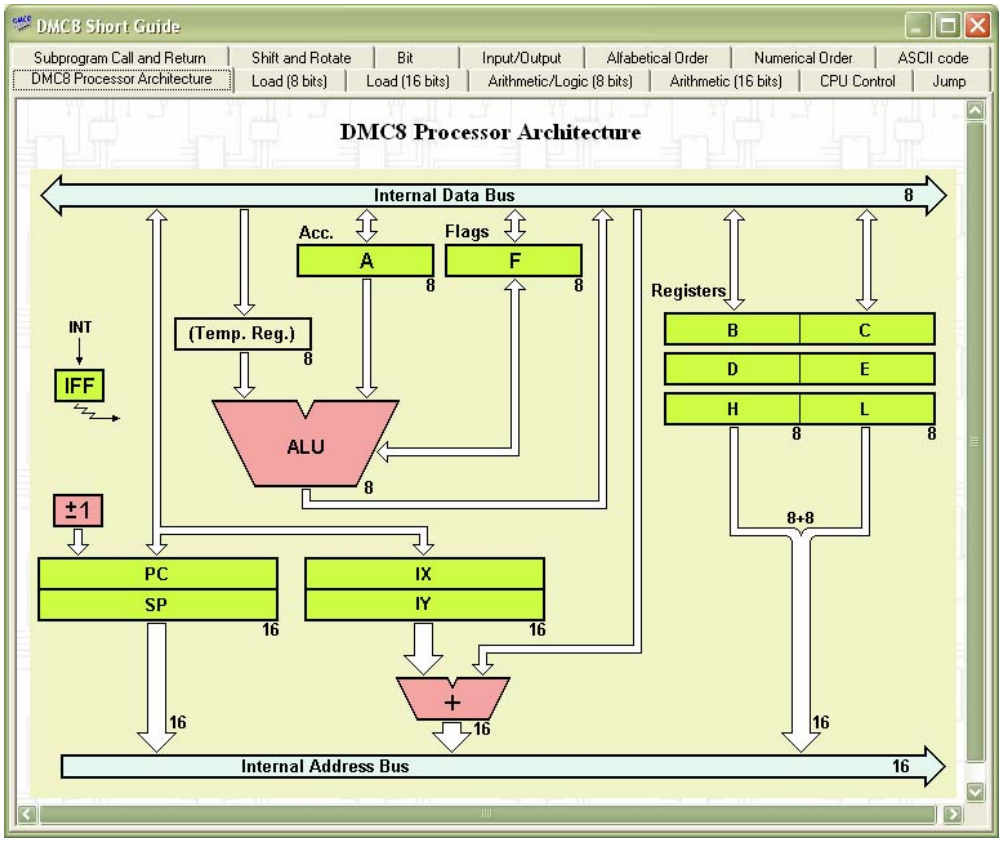


Fig. 25: The DMC8 "architecture", as shown by the help-system.

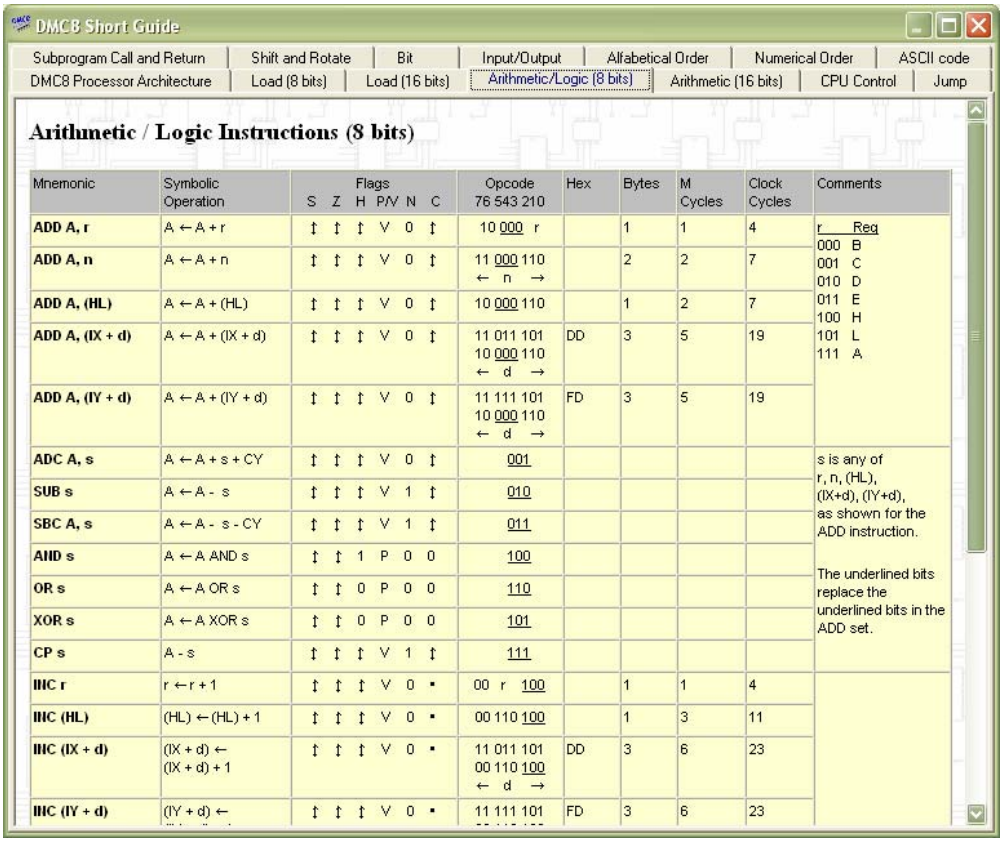


Fig. 26: An example of the 'on line' instruction set documentation: the Arithmetic and Logic instructions.

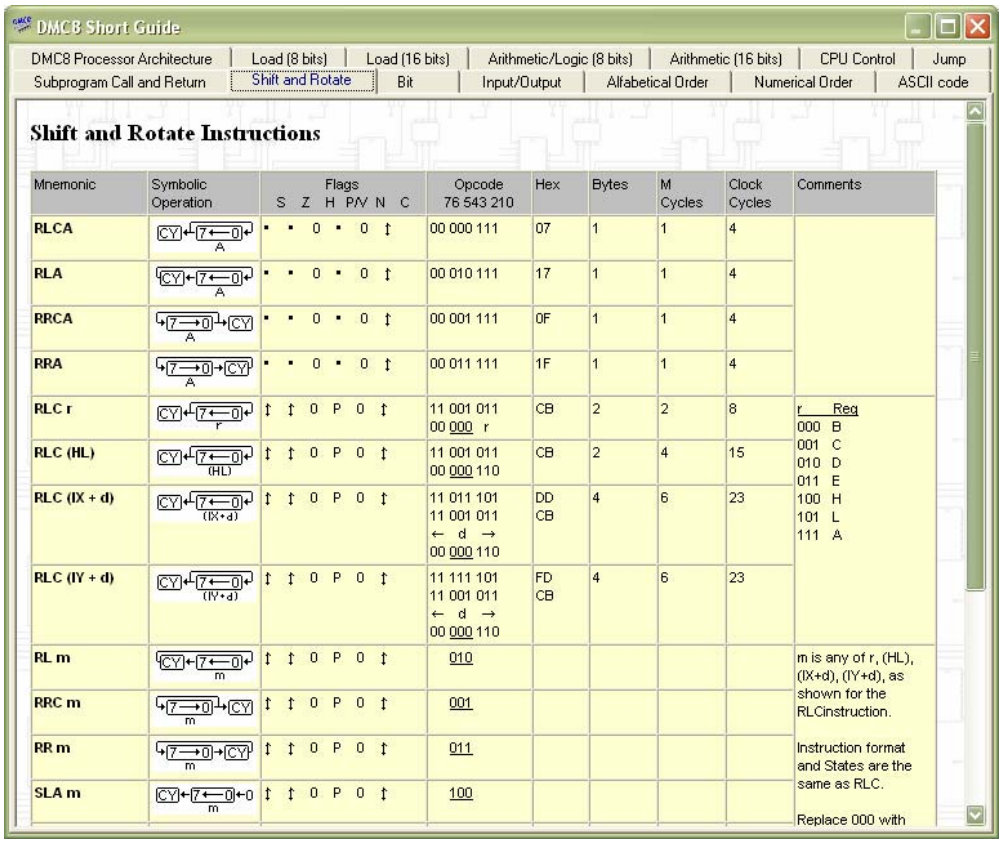


Fig. 27: Another example of the 'on line' instruction set documentation: the Shift and Rotate instructions.

Once completed the assembly coding of the program, the student will compile it. If no syntax error has been found, the verification of the program functionality can start (Fig. 28).

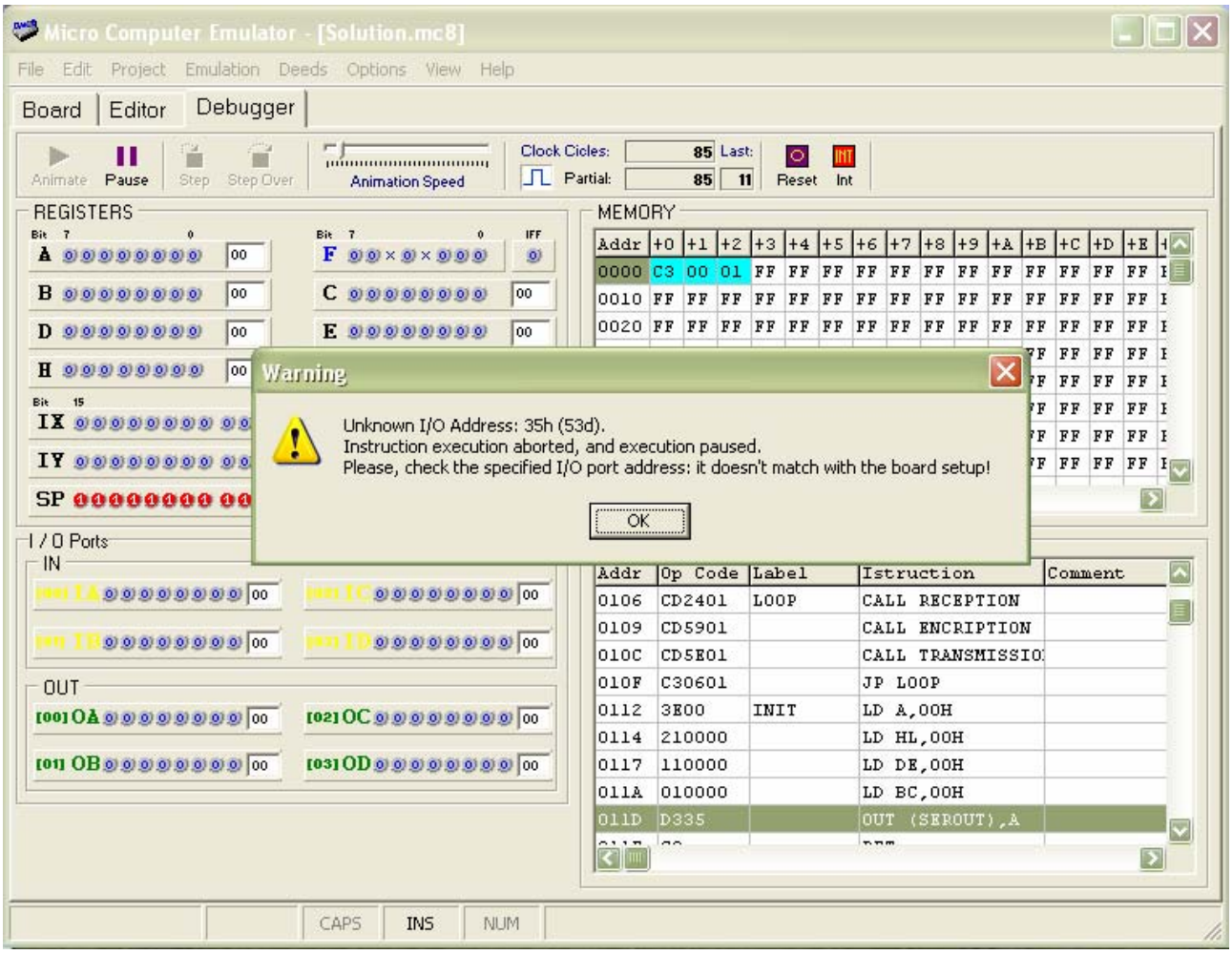


Fig. 28: The program under test in the interactive debugger of the d-McE: a Warning has be sent to the user.

In Fig. 28 the program is ‘Animated’ by the student, i.e. it is automatically executed step by step, at a ‘human readable’ speed. The speed is controlled by the cursor visible on the tool bar (“Animation Speed”).

In this example, a typical warning message is generated by the debugger. In a real case, if a port hardware address is not correctly instanced in the program code, unpredictable events could result. By the learner point of view, it could be very difficult realize what really happens in the system.

The d-McE debugger, instead, has been designed to track many common mistakes, reporting them to the student before then unwanted results could complicate the understanding of the wrong behaviour of the program.

In the present case (Fig. 28), the processor should execute the OUT instruction at address 011Dh. But the address instanced by the instruction is 35h, while no port has been set to respond to this address. So, the student has two possibilities: to return to the editor and change the source code, adapting it to the board setup, or to change the board setup.

To change the board setup, for instance, it is possible to activate (with a right-click on the port pane) the "I/O Ports Address Decoding" dialog window (Fig. 29).

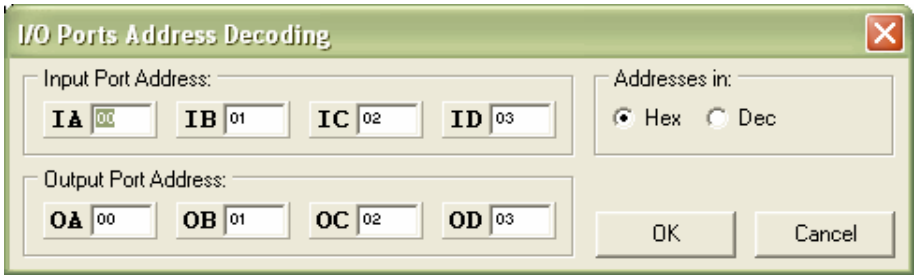


Fig. 29: Port addresses can be modified in the "I/O Ports Address Decoding" dialog window.

Another possibility, that resembles the real case, is to switch the current d-McE "page" and visualize the physical board, as seen in Fig. 74. Now it is possible to toggle, with a mouse click, the address 'dip-switches' that define the hardware address decoding (Fig. 30).
IA, IB, IC and ID are the addresses of the four parallel input ports available on board; OA, OB, OC and OD are those of the four output ports.

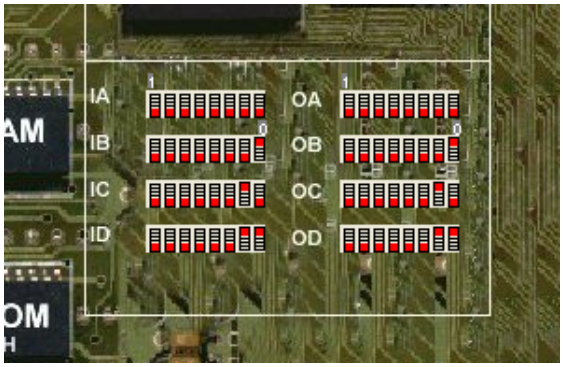


Fig. 30: Port addresses can be modified by a mouse click on the simulated 'on board' dip switches.

When finished, the student had to compile and deliver a report. A template file for the report is available in the assignment page (see Fig. 31).

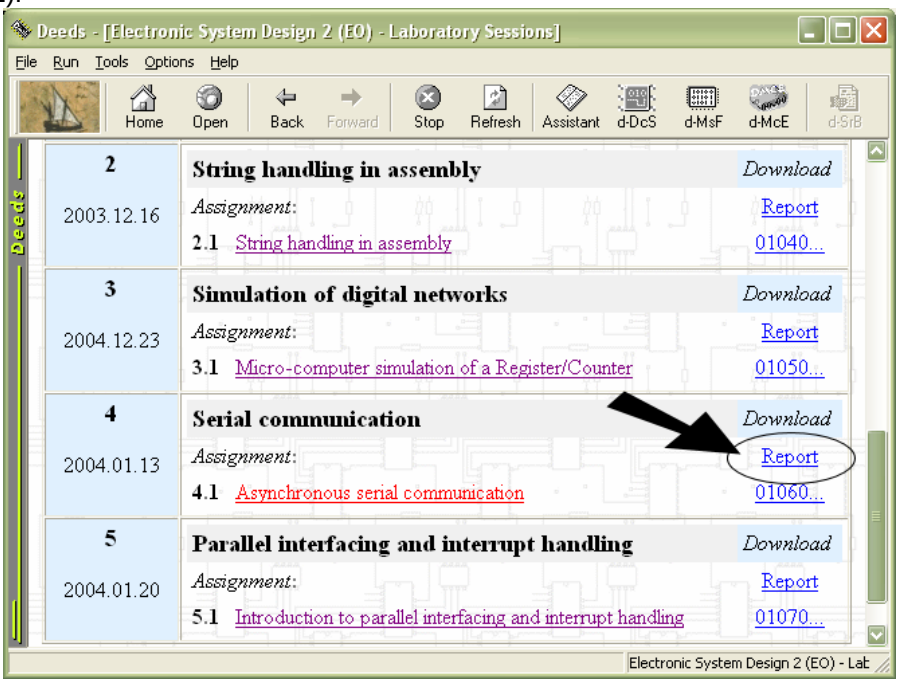


Fig. 31: The student can download the report template to speed up its compilation and delivering.

In this case, the template presents only a header that permit to uniform all the report styles, making easier the teacher task (Fig. 32).

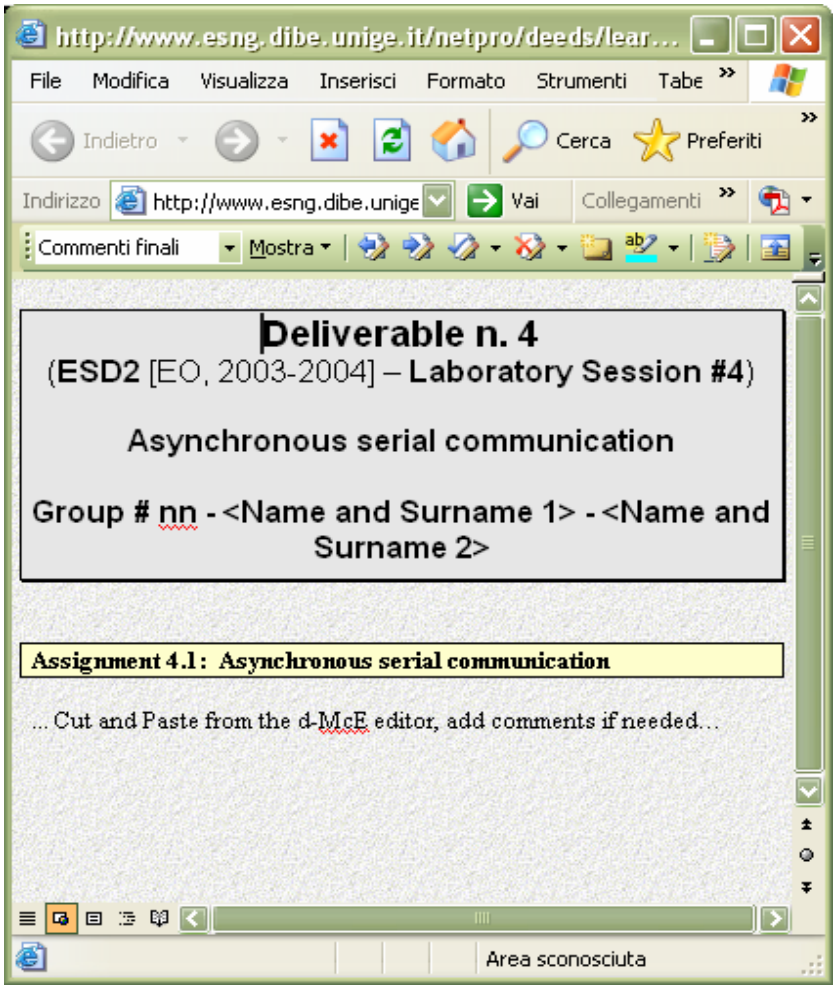


Fig. 32: The simple template provided on the web page, that the student can download.

In the next figure, an example of complete report is displayed (Fig. 33).

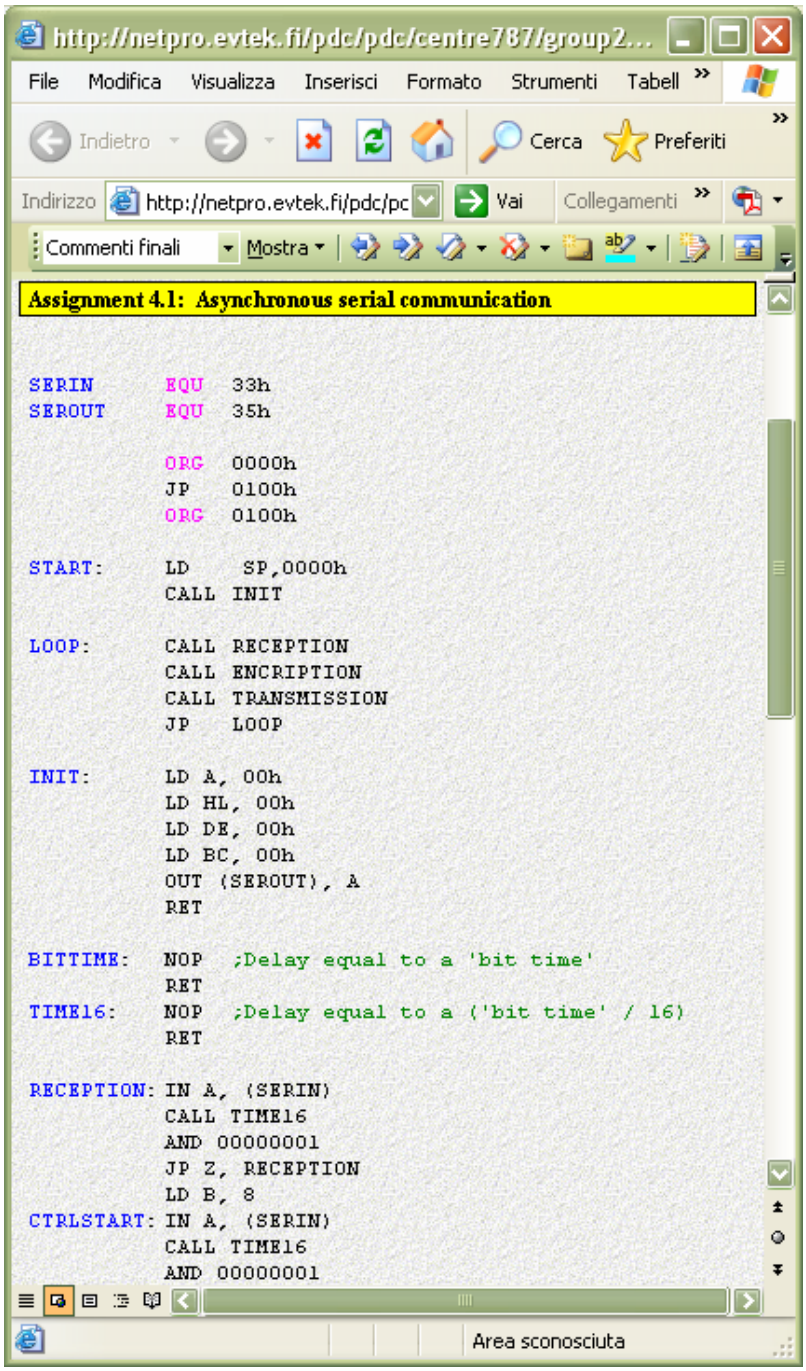


Fig. 33: A partial view of a ‘final’ student report.