

## 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**

Fig. 1: The header of the Deeds Learning Materials page, in the Deeds web site (DIBE, University of Genoa, Italy). Fig. 2: A particular of the index of the Deeds Learning Materials, in the Deeds web site (DIBE, University of Genoa, Italy) Fig. 3: This page contains all the Deeds learning materials, organized as single exercises and classified by topics (DIBE, University of Genoa, Italy) Fig. 4: The page related to the course "Electronic System Design 1 (IN-TL-BM)" targeted to	4 5 6 7
University of Genoa, Italy). Fig. 2: A particular of the index of the Deeds Learning Materials, in the Deeds web site (DIBE, University of Genoa, Italy) Fig. 3: This page contains all the Deeds learning materials, organized as single exercises and classified by topics (DIBE, University of Genoa, Italy) Fig. 4: The page related to the course "Electronic System Design 1 (IN-TL-BM)" targeted to	5 6 7
(DIBE, University of Genoa, Italy) Fig. 3: This page contains all the Deeds learning materials, organized as single exercises and classified by topics (DIBE, University of Genoa, Italy) Fig. 4: The page related to the course "Electronic System Design 1 (IN-TL-BM)" targeted to	6 7
Classified by topics (DIBE, University of Genoa, Italy) Fig. 4: The page related to the course "Electronic System Design 1 (IN-TL-BM)" targeted to	7
(DIBE, University of Genoa, Italy) Fig. 4: The page related to the course "Electronic System Design 1 (IN-TL-BM)" targeted to	7
students of the first year of information and electronics engineering	
(DIBE, University of Genoa, Italy)	
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	11
has been automatically downloaded from the courseware site.	• •
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	17
and state variables .	
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	20
compilation and delivering.	
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	24
the trace of the solution, automatically downloaded from the courseware site.	
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	25
instructions.	
Fig. 27: Another example of the 'on line' instruction set documentation: the Shift and Rotate	26
instructions.	
Fig. 28: The program under test in the interactive debugger of the d-McE: a Warning has be sent	27
to the user.	
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	28
switches.	
Fig. 31: The student can download the report template to speed up its compilation and	28
delivering.	
Fig. 32: The simple template provided on the web page, that the student can download. Fig. 33: A partial view of a 'final' student report.	29 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: <u>http://esng.dibe.unige.it/Netpro/Deeds</u>, and it is necessary to click on the "Learning Materials" link, on the left of the page.

Run Iools Options Help	
Home Open Back Forward Sto	op Refresh Assistant d-DcS d-MsF d-McE d-SrB
<b>MECO</b> Digital I	Electronics: available exercises
Deeds Exercises	Collection of interactive exercises, developed for the courses of digital design (first and second year, information and electronics engineering) Department of Biophysical and Electronic Engineering
<u>Interactive</u>	e Laboratory Sessions (current year)
Interactive • Electronic System Design 1 (IN-TL-BM)	e Laboratory Sessions (current year) Introductory courses of Digital Design (first year information and
Interactive • Electronic System Design 1 (IN-TL-BM) • Electronic System Design 1 (EO)	e Laboratory Sessions (current year) Introductory courses of Digital Design (first year information and electronics engineering) Department of Biophysical and Electronic Engineering University of Genoa
Interactive • Electronic System Design 1 (IN-TL-BM) • Electronic System Design 1 (EO) • Electronic System Design 2 (EO)	e Laboratory Sessions (current year) Introductory courses of Digital Design (first year information and electronics engineering) Department of Biophysical and Electronic Engineering University of Genoa Introductory course to Microprocessor Systems (second year information and electronics engineering) Department of Biophysical and Electronic Engineering University of Genoa

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.

🦫 J	Deeds - [Deer Run Tools	ds - Exercises (by topic)] Options Help	
No. No.		ne Open Back Forward Stop Refresh Assistant d-DcS d-Ms	F d-McE d-SrB
1	8	Introduction to Finite State Machines	Download
eds		Re-thinking a synchronous counter as Finite State Machine (FSM)	00240
99		Reverse-engineering a synchronous sequential circuit	00250
		Design of a synchronous mod-5 up/down counter	<u>00270</u>
		Design of a simple serial line receiver	00280
	9	Design of finite state machines	Download
		Design of a timing sequence generator	00290
		Design and synthesis of a 3-bit up-counter for signed numbers	00260
		00300	
	10	Download	
		<u>00340</u>	
		Design of a serial-programmable pulse generator	<u>00310</u>
		00330	
	11	Micro-computer systems: introduction to assembly programming	Download
		Introduction to the Deeds Micro Computer Emulator	<u>01010</u>
		<u>01020</u>	
		Analysis and tracing of a simple arithmetic program	<u>01030</u>
	12	Micro-computer systems: assembly programming techniques	Download
		String handling in assembly	01040
	1	Deeds - Exer	cises (by topic)

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.

٠	🗞 Deeds - [Electronic System Design 1 (INF) - Laboratory Sessions]						
Eile	File Run Tools Options Help						
1	A	Home	Image: Constraint of the sector of the s	d-McE d-SrB			
1	e 0		Electronic System Design 1 (IN-TL-BM)				
18							
)ee(			Laboratory Sessions (2003-2004)				
1		A 11 1e c	sons and evercises available here have been decigned to be opened in the T	ands browsers			
	*	as <i>int</i>	<i>eractive sessions</i> . You can open it also in another browser as well, but you tive features. You can download the last <b>version of <i>Deeds</i> here</b>	will lose the			
	1.1			the II a late	E		
	Labora	tory	Tonics				
	Sessi	on	(click on a topic title to open it in the Deeds Assistant)				
	1 Simulation of simple combinational networks Download						
	2004.03	3.01	Assignments:	Report			
			1.1 Introduction to the Deeds Digital Circuit Simulator	00010. <u>.</u>			
			1.2 Analysis of simple logic gates	00020			
-	2		Simulation of combinational networks	Download			
	2004.03	3.08	Assignments:	Report			
			2.1 Analysis of a multiplexer (2 to 1)	00030. <u>n</u>			
			2.2 Analysis of a demultiplexer (1 to 2)	<u>00040</u>			
			2.3 Analysis of a simplified shared-line communication channel	<u>00050</u>			
	3		Analysis, synthesis and simulation of combinational				
	networks (a)						
	2004.03	5.15	Assignments:	Report			
			3.1 Analysis of a multi-level logic network	00060			
			3.2 Design of a programmable logic gate	00070			
			Electronic Syst	tem Design 1 (INF) - Lab	× //		

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".

Deeds - [Electron Run Tools Optio	ic System Design 1 (INF) - Laboratory Sessions]	
Home Rail	Image: Second state         Image: Second state	Assistant - [DEDS Laboratory Session]
2004.03.29	to sequential circuits         Assignments:         5.1       Analysis and elimination of static hazards         5.2       Analysis of a Set-Reset Flip-Flop         5.3       Timing analysis of a D-PET flip-flop         5.4       Timing analysis of a JK-PET flip-flop         5.5       Analysis of a 3-bit shift-register (D-PET)	Back Forward         Menu           Deeds         Analysis of a module-4 Johnson counter         O0200           The following synchronous sequential network is a module-4 Johnson counter. It is based on a two-stages Shift Register, with the last bit returned, inverted, to the first stage input. Click on the figure to open in the d-DcS the counter schematic:
<b>6</b> 2004.04.05	Analysis of synchronous and asynchron sequential networks         Assignments:         6.1       Analysis of a module-4 Johnson counter         6.2       Analysis of a synchronous sequence generato         6.3       Maximum clock frequency of a synchronous sequence sequence sequence         network       6.4	Clear Clock Cl
7 2004.04.19	Analysis of sequential networks as finite machines         Assignments:         7.1 Re-thinking a synchronous counter as Finite S (FSM)         7.2 Reverse-engineering a synchronous sequentia	Then, verify the network behaviour with the timing simulator 📅 Study, first, the count sequence observed on the outputs C_phase (MSB) and B_phase (LSB). What kind of binary code sequence is it? Next, consider all the four outputs (A_phase, B_phase, C_phase and
8	Design of simple finite state machines	<b>D_phase</b> ), and observe the time relation among the four waveform.
2004.04.26	Assignments:	

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).

Assistant - [DEEDS L	aboratory Session]	
Back For	★ B <sub>3</sub> ward Menu	
Deeds	Analysis of a demultiplexer (1 to 2)	00040
Verify the behavio using the Deeds D d-DcS a trace of t schematic below:	or of the 1->2 demultiplexer represented in the figure Digital Circuit Simulator (d-DcS). Click on the figure the network's schematic, and then complete it to ob	ure below, e to open in the otain the
To complete the d (click here $\stackrel{\text{Per}}{\longrightarrow}$ of UO component to Once completed t $\stackrel{\text{Once completed t}}{\longrightarrow}$ of the network waveforms should	Arawing, you should also name the Input and Output r on the same button on the <b>d-DcS</b> tooolbar, then be named). the schematic, you'll be ready to start the <b>functions</b> rk, and then the <b>timing simulation</b> in this cas d be defined in order to distinguish easily the select	t components click on each al simulation ase, the input ed output from

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).

-	🌤 Deeds - [Electronic System Design 1 (INF) - Laboratory Sessions]						
Eile	<u>File Run I</u> ools <u>O</u> ptions <u>H</u> elp						
The second	Home	Image: Open     Imag	-SrB				
		1.2 Analysis of simple logic gates	<u>00020</u>				
a B	2	Simulation of combinational networks	Download				
ĕ	2004.03.08	Assignments:	Report )				
Ĩ		2.1 Analysis of a multiplexer (2 to 1)	00030				
		2.2 Analysis of a demultiplexer (1 to 2)	00040				
		2.3 Analysis of a simplified shared-line communication channel	<u>00050</u>				
3 Analysis, synthesis and simulation of combinational networks Downlo							
	2004.03.15	Assignments:	Report				
		3.1 Analysis of a multi-level logic network	00060				
	3.2 Design of a programmable logic gate 000070						
		3.3 Synthesis of a boolean function	<u>00080</u>				
IJ	4	Analysis, synthesis and simulation of combinational networks	Download 💟				
	Electronic System Design 1 (INF) - Laborat 🥢						

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).

🕙 lab02_template. doc - Microsoft Word
Eile Modifica Visualizza Inserisci Formato Strumenti Iabella Finestra ? ×
🗄 🗋 📸 🛃 💪 🎒 🛃 🕵   🌮 🎎   🖇 🔤 🕰 🕩 -   🧶 📰   100% 🕞 🎯   🖽 Lettura 🍟
Titolo + Casella : ▼ Arial   ▼ 14 ▼   G C S   三 三 三   三   ⊡ ▼ ▲ ▼ 🙄
A · · · 1 · · · 2 · · · 3 · · · 4 · · · 5 · · · 6 · · · 7 · · · 8 · · · 9 · · · 10 · · · 11 · · · 12 · · · 13 · · · 14 · ▷      I     I
3) Timing Diagram paste the timing diagram here
Assignment 2.2: Analysis of a de-multiplexer (1 to 2)
1) Schematic paste here your schematic
2) Truth Table fill the truth table
In SelOut Outl Outl
3) Timing Diagram
paste the timing diagram here
Assignment 2.3: Analysis of a simplified shared-line communication channel
1) Schematic paste here your schematic
2) Timing Diagram
necessary to copy and paste the timing results in a few separate images

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.

🕸 Deeds - [Electronic System Design 1 (INF) - Laboratory Sessions]							
Eile Run Iools Options Help							
1	Home	Image: Open     Ima	McE d-SrB				
   		6.3 Maximum clock frequency of a synchronous sequential network       00220         6.4 Analysis of asynchronous up and down counters       00230					
Dee	7	Analysis of sequential networks as finite state machines	Download				
	2004.04.19	Assignments:	Report				
		7.1 <u>Re-thinking a synchronous counter as Finite State Machine (FSM</u>	<u>) 00240</u>				
		7.2 <u>Reverse-engineering a synchronous sequential circuit</u>	<u>00250</u>				
	8	Design of simple finite state machines	Download				
	2004.04.26	Assignments:	Report				
		8.1 Design of a synchronous mod-5 up/down counter	<u>00270</u>				
		8.2 Design of a simple serial line receiver	<u>00280</u>				
	9	Design of finite state machines	Download				
	2004.05.03	Assignments:	Report				
		9.1 Design of a timing sequence generator	<u>00290</u>				
		9.2 Design of a serial data processor	<u>00300</u>				
	10	Design of a serial-programmable pulse generator	Download				
	2004.05.10	Assignments:	Report 🛛				
file:\\	\D:\DeedsProject\We	ebSite\NetPro\Deeds\LearningMaterials\LM\00240_Re_Thinking_Sync_Col Electronic System	n Design 1 (INF) - Labc //				

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.

1	Back Forward Menu	
	You can use the <u>ASM diagram template</u> provided, where you'll find the state variables X,Y and Z already defined, as well as the <b>outputs</b> QC, QB and QA, and the <b>inputs</b> DIR and EN. In the template, the codes of five states have been also defined. At the <b>reset</b> , the counter should start from zero. For this reason, the 'a' state is the 'Reset' state (i.e. the 'starting' state of the FSM, at the activation of the asynchronous !Reset). In the template, as it is convenient in the present case, the state codes have been assigned equal to the outputs values ( $X = QC$ , $Y = QB$ and $Z = QA$ ).	
N. LAND & SALLARD	Verify, using the <b>timing simulation D</b> , the correct <b>sequence</b> of the output values and the state codes. Once you have finished the FSM design, you can <b>import</b> it in the <b>d</b> -DeS as a component. You can use the <b>d</b> DeS advected template	
ALC: NO	provided, and complete it with the FSM component. Repeat the simulation of the counter with the d-DcS timing simulator	-

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.

髂 Finite State Machine Sin	nulator - [ex00270_	1_tem.fsm] - [ASM Stat	e Chart]	
🎆 Eile Edit View Simulation	<u>W</u> indow <u>D</u> eeds <u>H</u> elp			_ 8 ×
📙 🔁 🗁 🔛 🏝 🛛 🛠 P	] @, @,   ⊏	1 < 0 J 🔪 🖪 🛛	3	
. 000	001	010	011	100 💻
l ( )	Ф	QB 0	QB,QA	<sup>oc</sup> e
A4: (111, 22)				11

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 <u>16</u>.



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  $\stackrel{P}{\longrightarrow}$ ). 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).

*	🗞 Deeds - [Electronic System Design 1 (INF) - Laboratory Sessions]						
Eile	<u>R</u> un <u>T</u> ools <u>O</u> ptio	ns <u>H</u> elp					
and the second s	Home	Open         Image: Constraint of the sector of the s	d-McE d-SrB				
- spe	7	Analysis of sequential networks as finite state machines	Download				
ě	2004.04.15	Assignments:	Report				
Ī		7.1 <u>Re-thinking a synchronous counter as Finite State Machine</u> (FSM)	<u>00240</u>				
		7.2 Reverse-engineering a synchronous sequentize incuit	00250				
	8	Design of simple finite state machines	Download				
	2004.04.26	Assignments:	Report				
	2004.04.26	Assignments: 8.1 Design of a synchronous mod-5 up/down counter	<u>Report</u> 00270				
	2004.04.26	Assignments: 8.1 Design of a synchronous mod-5 up/down counter 8.2 Design of a simple serial line receiver	Report 00270 00280				
	2004.04.26 <b>9</b>	Assignments: 8.1 Design of a synchronous mod-5 up/down counter 8.2 Design of a simple serial line receiver Design of finite state machines	Report 00270 00280 Download				
	2004.04.26 9 2004.05.03	Assignments: 8.1 Design of a synchronous mod-5 up/down counter 8.2 Design of a simple serial line receiver Design of finite state machines Assignments:	Report 00270 00280 Download Report				
	2004.04.26 9 2004.05.03	Assignments:         8.1 Design of a synchronous mod-5 up/down counter         8.2 Design of a simple serial line receiver         Design of finite state machines         Assignments:         9.1 Design of a timing sequence generator	Report 00270 00280 Download Report 00290				
	2004.04.26 9 2004.05.03	Assignments:         8.1 Design of a synchronous mod-5 up/down counter         8.2 Design of a simple serial line receiver         Design of finite state machines         Assignments:         9.1 Design of a timing sequence generator         9.2 Design of a serial data processor	Report           00270           00280           Download           Report           00290           00300				

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.

	×
<u><sup>1</sup> Eile Modifica Visualizza Inserisci Formato Strumenti I</u> abella Fi <u>n</u> estra <u>2</u>	×
🗄 🗋 🚰 🛃 💪 🎒 🛃 🔍 🖤 🎎 ⊨ 🛍 🗠 – 🗠 – 😫 🎫 🖣 🗍 🖽 Lettura –	⊧: ₹
j Titolo + Casella : ▼ Arial ▼ 14 ▼   <b>G</b> <i>C</i> <u>S</u>   <b>≡ ≡ ≡</b>   <b>∷</b>   <b>∏</b> ▼	•• ₹
۲۵۰ · · · · · · · · · · · · · · · · · · ·	~
Deliverable n. 8	
(ESDTINE, 2003/2004 – Laboratory Session #8)	
Design of Simple Finite State Machines	_
	-
Group # nn - <name 1="" and="" surname=""> - <name 2="" and="" surname=""></name></name>	
Assignment 8.1: Design of a synchronous mod-5 up/down counter	
1) d-FsM ASM Diagram	Γ
paste here the ASM Diagram	
2) d-FsM Timing diagram	
2) d-FSM Timing diagram paste here the timing diagram	
2) d-FsM Timing diagram paste here the timing diagram 3) d-DcS Schematic paste schematic	
2) d-FSM Timing diagram paste here the timing diagram 3) d-DcS Schematic paste schematic	×
<ul> <li>2) d-FsM Timing diagram paste here the timing diagram</li> <li>3) d-DcS Schematic paste schematic</li> <li>4) d-DcS Timing diagram paste here the time paste here there the time past</li></ul>	×
<ul> <li>2) d-FsM Timing diagram paste here the timing diagram</li> <li>3) d-DcS Schematic paste schematic</li> <li>4) d-DcS Timing diagram paste here the timing diagram</li> </ul>	± 0 7
2) d-FsM Timing diagram paste here the timing diagram 3) d-DcS Schematic paste schematic 4) d-DcS Timing diagram paste here the timing diagram = 0 0 3 0 <	± 0 ∓

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).

De	shae		Asv	nchro	nous	erial o	omm	nicati	on		11060
re avai re avai e-seri he stat 1. ( 2. 8	oprocess graphs a ilable in t ialisation ndard for One star data bit One stop	nd re-t he syste n and re-t mat use t bit (hi s, b7b bit (lo <sup>*</sup>	ed system ransmit em hardw e-seriali ed here f gh); gh); 0 (b7 ah w);	n receiv is them ware, so is ation for asyn ead);	into and it will b of the se	other asy be neces erial sign as serial	rs from a mchrono sary to v als. commun	an async us seria vrite a j	nronou 1 line. O progran s specifi	s serial in nly paral n that imj ied as fol	ie, <b>llel por</b> blement lows:
4. I	Bit/rate of	E <b>100</b> b:	its per se	econd.							
4. I	Bit/rate of	f 100 b b7	its per se <b>b6</b>	b5	b4	b3	b2	b1	b0	Stop	

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.

Y	↓     ↓       Back     Forward       Menu
	Solution guidelines
Γο cap μS).	ure the start bit, the receiver samples SERIN with a period of $1/16$ of the bit time (625
When t sample assume looking	he line goes <b>high</b> , the receiver continues to sample <b>SERIN</b> at the same rate. If the line is <b>d high for S times continuously</b> , the receiver declares " <b>recognised</b> " the <b>start bit</b> and s to be at the <b>middle</b> of the bit time. On the contrary, the receiver re-starts operations, again for the next valid start bit.
lf the ro time (i bits (b	ceiver recognises the start bit, it will continue sampling the line, but <b>only once every bit</b> 0 mS), starting <b>from the middle</b> of the recognised start bit, acquiring the following <b>8 data</b> ( <b>b0</b> ), and the <b>stop bit</b> .
lf the s start b	<b>op bit</b> is <b>wrong</b> , the data bits are <b>ignored</b> and the system re-starts, waiting again for a <b>new</b> t.
If the s be seri	op bit is correct, the cryptographic operation takes place, and the resulting byte will ally transmitted, according to the defined standard, onto the SEROUT line.
The protect of a po	gram should start on activation of the system hardware <b>RESET</b> . Click here to load a <u>trace</u> ssible solution.
Sugge For the As sug	<i>tions</i> sake of simplicity, suppose that it is <b>not possible</b> to <b>receive data</b> when <b>transmitting</b> . sested in the solution trace, divide the assembly code in reusable subprograms, and suppose a two delay subroutines named <b>BITTIME</b> (10 mS) and <b>TIME16</b> (625 uS)

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.

Micro Computer Emulator - [ex01060_1_tem.mc8]						
ile <u>E</u> dit <u>P</u> roject E <u>m</u> ulation <u>D</u> eeds <u>O</u> ptions <u>V</u> iew <u>H</u> elp						
Board Editor Debugger						
D 🛩 🖬 🚭   X 🖻 🖻 🛩   🔍   💁   🛗   👹						
📾 d: ] 🔹 🔹 ex01060_1_tem.mc8						
➢ D:\ ➢ DeedsProject ➢ Deeds ➢ HTML ➢ Circuits SERIN EOU ;input parallel port						
MC8 Files (*.mc8)						
Code_mc8 JP 0100h Codemc8 ORG 0100h ESAME_06_02_2004.m						
START: <iniz. pointer="" stack=""> CALL INIT</iniz.>						
LOOP: CALL <receiver subprogram=""> CALL <encription subprogram=""> CALL <transmitter subprogram=""> JP LOOP</transmitter></encription></receiver>						
; initialization INIT: <clear and="" output="" port="" variables=""></clear>						
CAPS INS NUM						

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.

Subprogram Call ar DMC8 Processor Ar	nd Return   Shift rchitecture   Load	and Rotate (8 bits)	Bit	Input/Outpu Arithmetic/	t   Logic (8	Alfabetica bits)	l Order	Numeric (16 bits)	al Order ASCII o
Arithmetic /	Logic Instru	ctions (8	bits)		머				
Mnemonic	Symbolic Operation	FI SZH	lags P/VNC	Opcode 76 543 210	Hex	Bytes	M Cycles	Clock Cycles	Comments
ADD A, r	A ← A + r	1 1 1	1 0 V	10 <u>000</u> r		1	1	4	r <u>Req</u>
ADD A, n	A ← A + n	1 1 1	A 0 ţ	11 <u>000</u> 110 ← n →		2	2	7	001 C 010 D
ADD A, (HL)	$A \leftarrow A + (HL)$	1 1 1	1 0 V	10 <u>000</u> 110		1	2	7	011 E
ADD A, (IX + d)	A ← A + (IX + d)	1 1 1	V 0 1	11 011 101 10 <u>000</u> 110 ← d →	DD	3	5	19	101 L 111 A
ADD A, (IY + d)	A ← A + (IY + d)	1 1 1	V 0 1	11 111 101 10 <u>000</u> 110 ← d →	FD	3	5	19	
ADC A, s	A ← A + s + CY	1 1 1	1 0 V	001					s is any of
SUB s	A ← A - s	1 1 1	V 1 1	<u>010</u>					- r, n, (HL), (IX+d), (IY+d),
SBC A, s	A ← A - s - CY	1 1 1	V 1 ‡	<u>011</u>	1				as shown for the ADD instruction.
AND s	A ← A AND s	1 1 1	P 0 0	<u>100</u>	1				
OR s	A ← A OR s	0 1 1	P 0 0	<u>110</u>					replace the
XOR s	A ← A XOR s	0 1 1	P 0 0	<u>101</u>					underlined bits in the ADD set.
CP s	A - s	1 1 1	V 1 t	<u>111</u>					
INC r	r ← r + 1	1 1 1	V 0 •	00 r <u>100</u>		1	1	4	
INC (HL)	(HL) ← (HL) + 1	1 1 1	V 0 •	00 110 <u>100</u>	1	1	3	11	
INC (IX + d)	(IX + d) ← (IX + d) + 1	1 1 1	V 0 •	11 011 101 00 110 <u>100</u> ← d →	DD	3	6	23	
INC (IY + d)	(IY + d) ←	1 1 1	V 0 •	11 111 101	FD	3	6	23	

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

DMC8 Processor	r Architecture	oad (8 bits)	Load (16	bits) Arithm	netic/Log	jic (8 bits) Alfabe	Arithme	tic (16 bits)	CPU Control J	lump
Shift and ]	Rotate Instru	ictions								
Mnemonic	Symbolic Operation	Fla SZH	ags P/VNC	Opcode 76 543 210	Hex	Bytes	M Cycles	Clock Cycles	Comments	
RLCA		••••	t 0	00 000 111	07	1	1	4		
RLA		••••	0 ţ	00 010 111	17	1	1	4	-	
RRCA		••••	1 0	00 001 111	OF	1	1	4	-	
RRA		••••	1 0	00 011 111	1F	1	1	4		
RLC r		1 1 0 F	1 0	11 001 011 00 <u>000</u> r	СВ	2	2	8	r <u>Reg</u> 000 B	
RLC (HL)		1 1 0 F	1 0	11 001 011 00 <u>000</u> 110	СВ	2	4	15	001 C 010 D 011 E	
RLC (IX + d)		1 1 0 F	0 1	11 011 101 11 001 011 ← d → 00 <u>000</u> 110	DD CB	4	6	23	100 H 101 L 111 A	
RLC (IY + d)	CX+ <u>(14+0</u> +	1 1 0 F	0 ţ	11 111 101 11 001 011 ← d → 00 <u>000</u> 110	FD CB	4	6	23		
RL m		t t o F	1 0	010					m is any of r, (HL), (IX+d), (IY+d), as	
RRC m		1 1 0 F	0 1	001					shown for the RLCinstruction.	
RR m		1 1 0 F	0 1	<u>011</u>					Instruction format and States are the	
SLA m		1 1 0 F	1 0	100					same as RLC.	

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).

Micro Computer Emulator - [Solution.mc8]				
File Edit Project Emulation Deeds Options View Help				
Board Editor Debugger				
Animate Pause Step Over Animation Speed	oles: 85 Las rtial: 85	st: <mark>O</mark> III Reset Int		
REGISTERS	MEMORY			
Bik 7 0 Bik 7 0 IFF <b>A</b> 0 0 0 0 0 0 0 0 0 0 <b>F</b> 0 0 × 0 × 0 0 0 0	Addr +0 +1 +2	+3 +4 +5 -	+6 +7 +8 +9 +A +E	8 +C +D +E 4
Paragener III Carronner III	0000 C3 00 01	FF FF FF I	FF FF FF FF FF FI	FFFFFFF
B 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0010 FF FF FF	FF FF FF I	FF FF FF FF FF FI	FFFFFFFF
D 99999999 00 E 99999999 00	0020 FF FF FF	FF FF FF I	FF FF FF FF FF FI	FFFFFFF
H 0000000 00 Warning				FFFFFFF
Bia 15				FF FF FF I
IX 000000000 00 🔥 Unknown I/O Address: 35h (53	id).		6	FFFFFFFF
Instruction execution aborted,	and execution pause	ed.		PR FF FF FF F
SP 0000000 00	ОК			
	Addr Op Code	Label	Istruction	Comment 🔼
	0106 CD2401	LOOP	CALL RECEPTION	
	0109 CD5901		CALL ENCRIPTION	
	010C CD5E01		CALL TRANSMISSIO	
OUT	010F C30601	1	JP LOOP	
[00]OA000000000 00 [02]OC00000000 00	0112 3800	INIT	LD A,OOH	
	0114 210000		LD HL,00H	
[01] OB000000000000000000000000000000000000	0117 110000		LD DE,OOH	
	011A 010000		LD BC,00H	
	011D D335		OUT (SEROUT),A	
		d /		
CAPS INS NUM				

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).

🏇 <u>I</u>	🏶 Deeds - [Electronic System Design 2 (EO) - Laboratory Sessions]						
Eile	<u>R</u> un <u>T</u> ools <u>O</u> ptio	ns <u>H</u> elp					
	Home	Image: style="text-align: center;">Image: style="text-align: c	SF d-McE d-SrB				
1	2	String handling in assembly	Download				
· Deeds	2003.12.16	Assignment: 2.1 String handling in assembly	<u>Report</u> 01040				
	3	Simulation of digital networks	Download				
	2004.12.23	Assignment:	Report				
		3.1 Micro-computer simulation of a Register/Counter	<u>01050</u>				
	4	Serial communication	Download				
	2004.01.13	Assignment:	Report				
		4.1 Asynchronous serial communication	01060				
	5	Parallel interfacing and interrupt handling	Download				
	2004.01.20	Assignment:	Report				
		5.1 Introduction to parallel interfacing and interrupt handling	01070				
		Electronic Sy	stem Design 2 (EO) - Lat //				

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).

🙆 http://ne	etpro.evtek.fi/pdc/pdc/centre787/group2 🔳 🗖 🗙
File Modific	a Visualizza Inserisci Formato Strumenti Tabell 🎽 🥂
	🔹 💿 - 💌 🗟 🏠 🔎 Cerca 🤺 Preferiti 🎽
Indirizzo 🙆	nttp://netpro.evtek.fi/pdc/pc 💟 ラ Vai 🛛 Collegamenti 🎽 🥎 🗸
Commenti fi	nali 🛛 🝷 Mostra 🕶   🤣 🤣 🤣 👻 🗸 🏷 🖉 💂
Assignment	4.1: Asynchronous serial communication
SERIN	EQU 33h
SEROUT	EQU 35h
	0RG 0000h
	JP 0100h
	ORC 0100h
START:	LD SP.0000h
	CALL INIT
1000	CALL DECEDITION
100P:	CALL ENCRIPTION
	CALL TRANSMISSION
11.1.25.10	JP LOOP
TNTT-	LD & 00b
	LD HL, OOh
Stell Steeld	LD DE, 00h
	LD BC, 00h
Shell Print	OUT (SEROUT), A
	F
BITTIME:	NOP ;Delay equal to a 'bit time'
TIMRIS	RET NOD Delay actual to a ('bit time' ( 16)
THISTO.	RET
1.15.25.20	
RECEPTION	IN A, (SERIN)
	AND 00000001
St. A. Standy	JP Z, RECEPTION 🔍
	LD B, 8
CTRLSTART:	CALL TIME 6
	AND 00000001
۲	Area sconosciuta

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