Deeds:

A Tool for Digital Design and Embedded System Training

Giuliano Donzellini & Domenico Ponta

donzie@unige.it , ponta@unige.it

DIBE – Department of Biophysical and Electronic Engineering

University of Genoa, Italy

http://www.esng.dibe.unige.it/deeds/

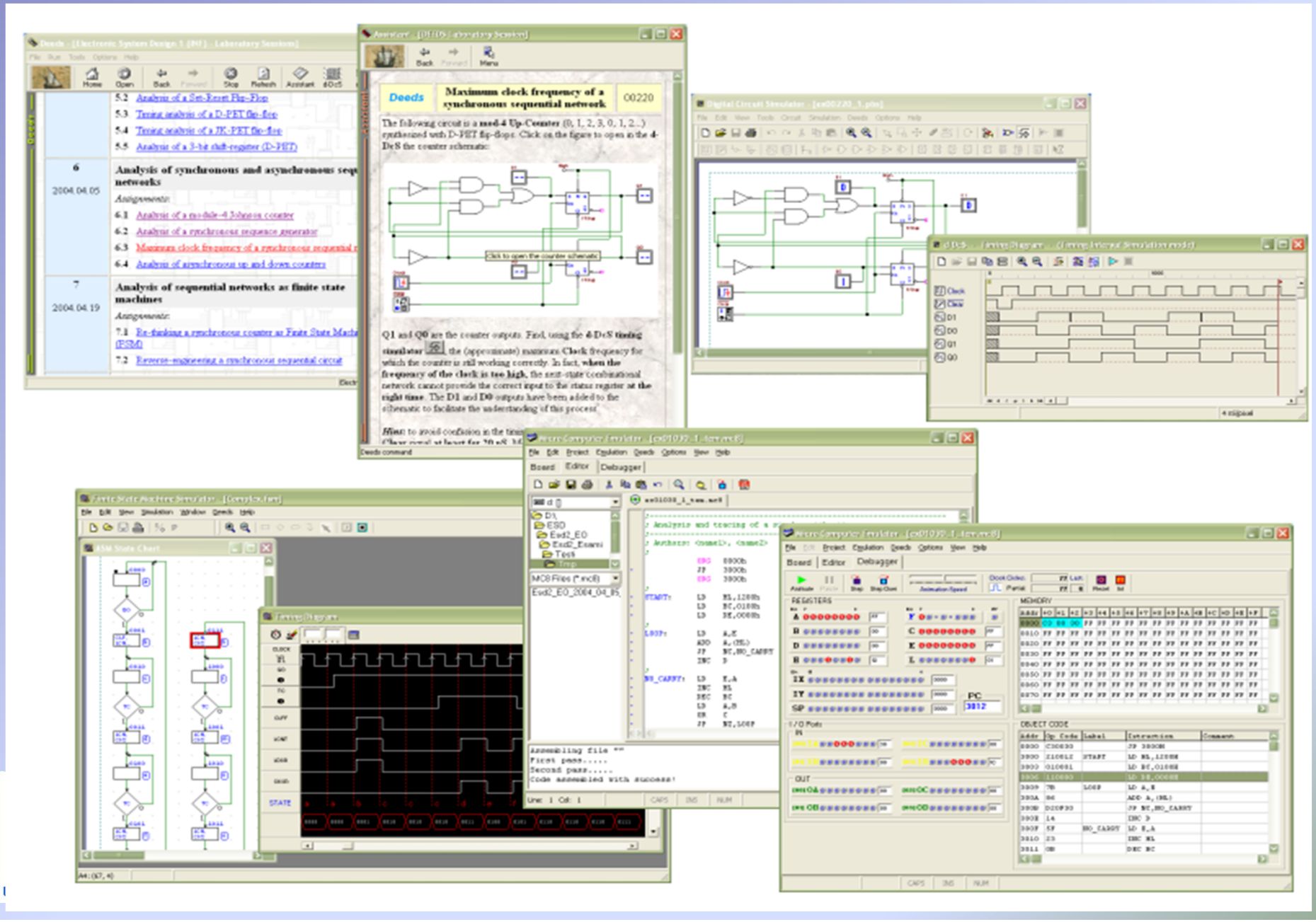
Symposium on Embedded Systems and Applications

University of Genoa

[esng] electronic systems and networking group
dibe - university of genoa - italy

GÖMSIS 2008

Deeds learning environment: a panoramic view



Learning digital design today

- The enormous and growing complexity of today's digital systems is putting new demands on education.
- EDA techniques are the core of a digital system development process, while traditional design and prototyping have lost their roles.
- EDA techniques are finding their way in engineering education.
- Professional EDA tools are available at very favourable conditions for educational institutions, and successfully used in engineering courses.

Starting from zero: how?

- A growing number of teachers introduce Hardware Description Languages and EDA tools in basic, introductory courses.
- Beginner students do not possess the skills and the frame of mind of the professional designers, whom the EDA tools are made for.
- We suspect that the use of professional tools and Hardware Description Languages in introductory courses may hide from learners important basic issues.
- Nevertheless, to achieve some familiarity with EDA techniques is a necessary target of a digital design course, even an introductory one.

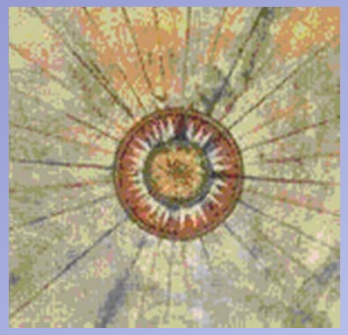
Learning the basics: "from zero to one"

- The design and simulation suite that we developed, ***Deeds***, represents our answer to the question: how?
- ***Deeds*** embodies our pedagogical approach to teaching and learning digital design ***"from zero to one"***
- Our target is to prepare students to the use of current and future EDA tools by building a solid understanding of the principle of digital design.
- This means to guide a learner, with no previous knowledge, typically in a one year long course, to achieve the foundation for designing embedded systems.

Deeds: Digital Electronics Education and Design Suite

- ***Deeds*** is developed at DIBE, University of Genoa
- The suite is composed by three simulators and a wide collection of associated ***learning material*** to learn-by-doing and practice with:
  - ***Combinational and sequential logic networks***
  - ***Finite state machine design***
  - ***Embedded microcomputer interfacing and programming***

Deeds: *the simulation tools*

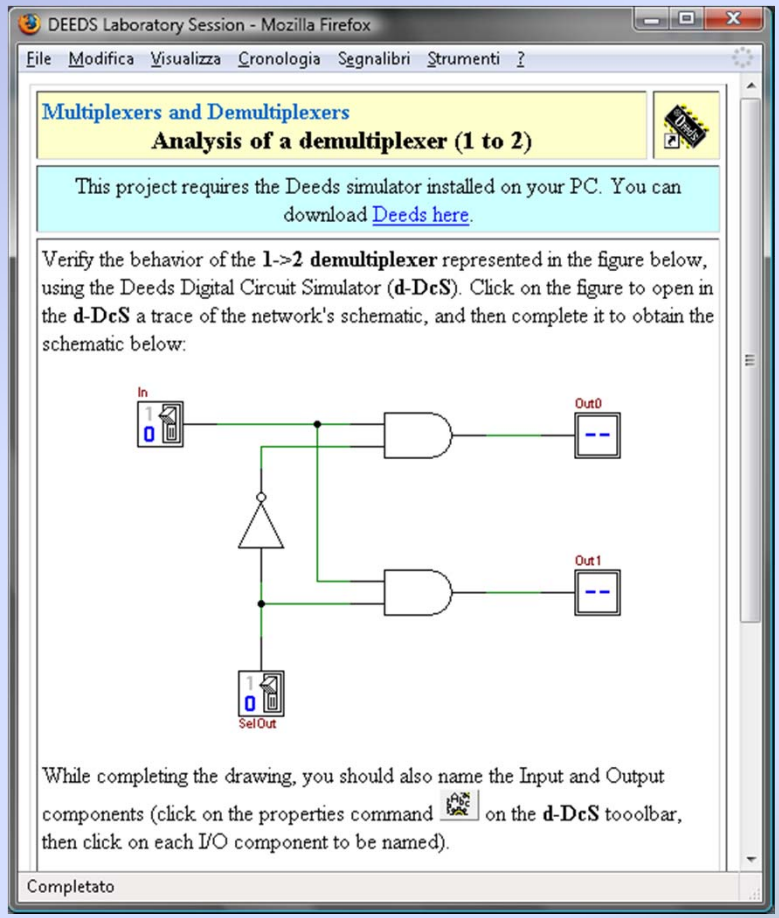


- d-DcS Digital Circuit Simulator
- d-FsM Finite State Machine Simulator
- d-McE Microcomputer Board Emulator

- The three simulators are *fully integrated*
- It is possible to design and simulate *digital systems* composed by *standard logic, finite state machines* and *microcomputers*
- It is therefore possible to understand the interaction among the hardware and software components of *embedded systems*

Deeds: the learning materials

- A ***laboratory session*** based on ***Deeds*** appears as web pages with text and figures
- Many of the schematics and visual objects are connected to the editing and simulation tools of ***Deeds***
- The web pages guide the learner in executing the assignment



Deeds: projects' list by topic

Deeds Home Page - Mozilla Firefox

File Modifica Visualizza Cronologia Segnalibri Strumenti ?

http://www.esng.dibe.unige.it/deeds/

Digital
Electronics
Education &
Design
Suite

- Last News
- What Is Deeds?
- The Environment
- Screen Shots
- **Learning Materials**
- Authors
- Version Notes
- Downloads
- Documents

[esng] electronic systems and networking group
dibe - university of genoa - italy

Chapter	Topics (*click on a topic title to open it in the Deeds Assistant*)	
1	**Introduction to digital electronics**	*Download*
	Introduction to the Deeds Digital Circuit Simulator	00010...
	Analysis of simple logic gates	00020...
2	**Multiplexers and Demultiplexers**	*Download*
	Analysis of a multiplexer (2 to 1)	00030...
	Analysis of a demultiplexer (1 to 2)	00040...
	Analysis of a simplified shared-line communication channel	00050...
3	**Applications of Boolean Algebra**	*Download*
	Analysis of a multi-level logic network	00060...
	Design of a programmable logic gate	00070...
	Synthesis of a boolean function	00080...
	Functional analysis of a two-level combinational network	00090...
	Analysis and design of multiplexer-based combinational networks	00100...
4	**Arithmetic circuits**	*Download*
	Design of a sign converter	00110...

Completato

Deeds: project assignment

Deeds - Exercises (by topic) - Mozilla Firefox

File Modifica Visualizza Cronologia Segnalibri Strumenti ?

http://www.esng.dibe.unige.it/deeds/Learnin

8	**Introduction to Finite State Machines** Re-thinking a synchronous counter as Finite State Mac Reverse-engineering a synchronous sequential circuit Design of a synchronous mod-5 up/down counter Design of a simple serial line receiver
9	**Design of finite state machines** Design of a timing sequence generator Design and synthesis of a 3-bit up-counter for signed r Design of a serial data processor
10	**Design of FSM-based digital systems** Design of a serial line receiver Design of a serial-programmable pulse generator Design of a programmable square-wave generator
11	**Micro-computer systems: introduction to** Introduction to the Deeds Micro Computer Emulator Analysis and tracing of a simple assembly program Analysis and tracing of a simple arithmetic program Analysis and tracing of a simple de-cryptographic prog Accessing tables in RAM memory

Completato

DEEDS Laboratory Session - Mozilla Firefox

File Modifica Visualizza Cronologia Segnalibri Strumenti ?

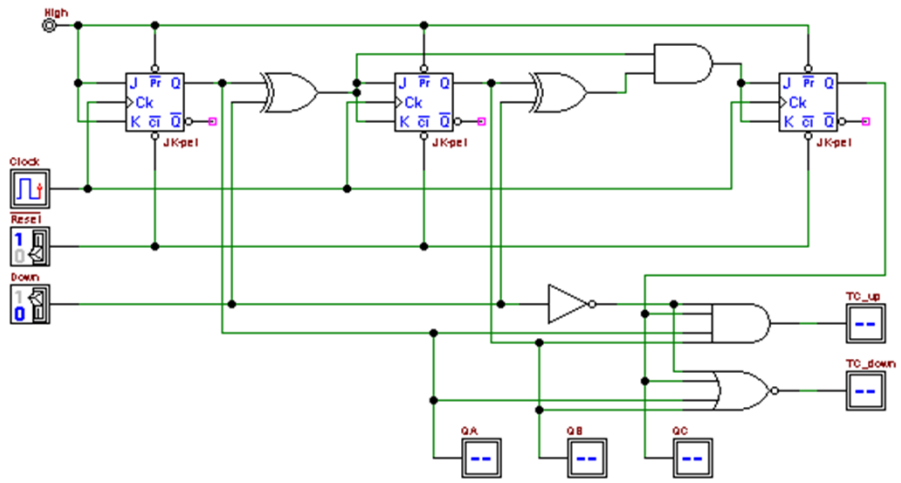
http://www.esng.dibe.unige.it/deec

Introduction to Finite State Machines

Re-thinking a synchronous counter as Finite State Machine (FSM)

This project requires the Deeds simulator installed on your PC. You can download Deeds here.

The following network is a **Synchronous Up/Down Binary Counter**. Click on the figure to open it in the **d-DcS**:



The input **Down** sets the direction of the count (*up* = '0', *down* = '1'). The output **TC_Up** (*Terminal Count, 'Up' direction*) is activated to **'1'** when the *up* count reaches the maximum number; instead, the output **TC_Down** (*Terminal Count, 'Down' direction*) is activated when the *down* count reaches zero. **Verify** such behaviour with the **timing simulator**.

Completato

University of Genoa

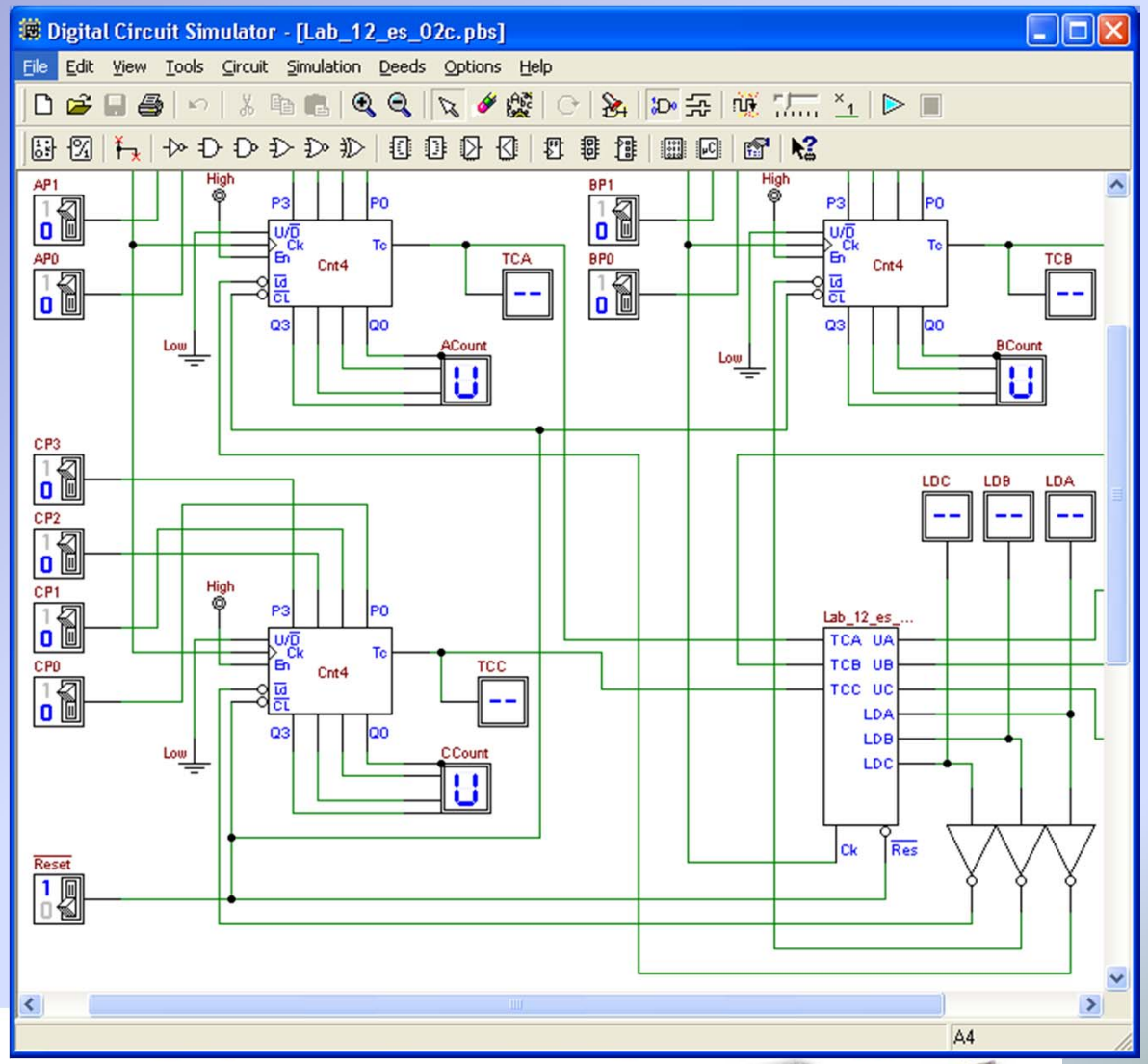
esng

electronic systems

dibe - university of geno

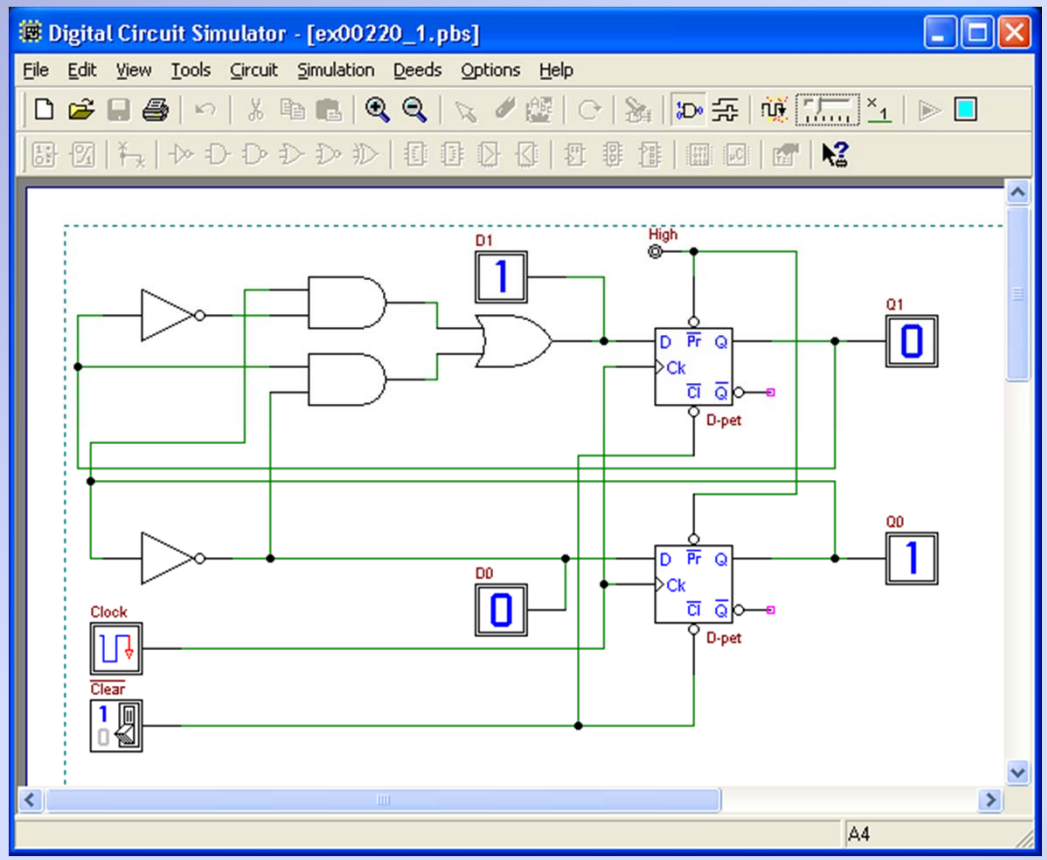
Deeds: the d-DcS Digital Circuit Simulator

- The d-DcS is a digital circuit simulator specifically developed with educational needs in mind
- The d-DcS has been designed to be easy to use, while maintaining quasi-professional features
- The user interface is intuitive
- The choice of digital components is based on their logical function, not on commercial lines
- Two simulation mode are available:
  a) *Interactive Animation*
  b) *Timing diagram*

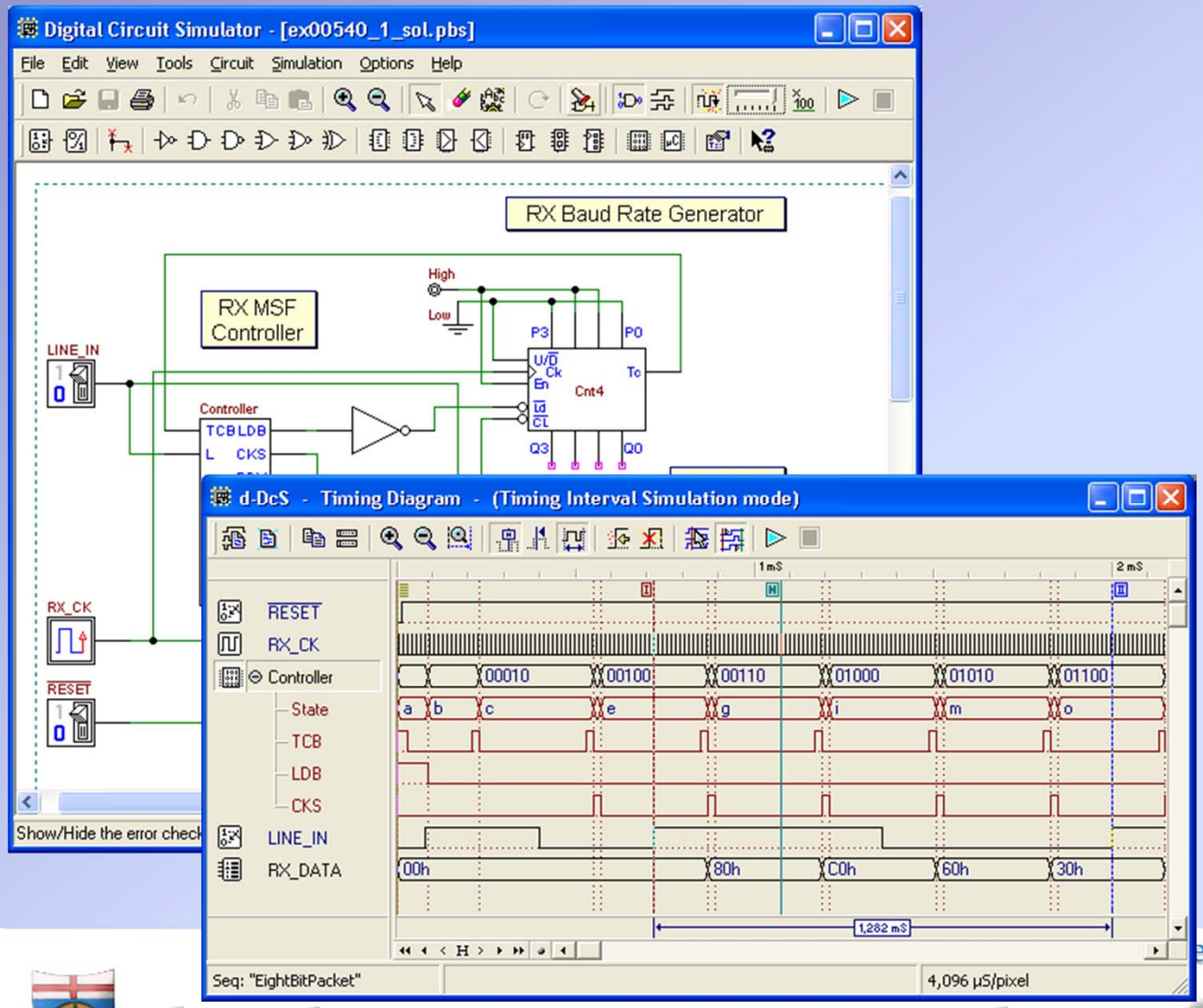


Deeds: the d-DcS Interactive Animation

- In the interactive animation mode the simulator processes input commands and displays output values
- The input components, on the left of the figure, are a clock and a logical level generator
- The level generator is represented as a switch, toggled by a mouse click
- The clock can be activated edge by edge or continously
- Output components display the values of the selected nodes (four in the figure)



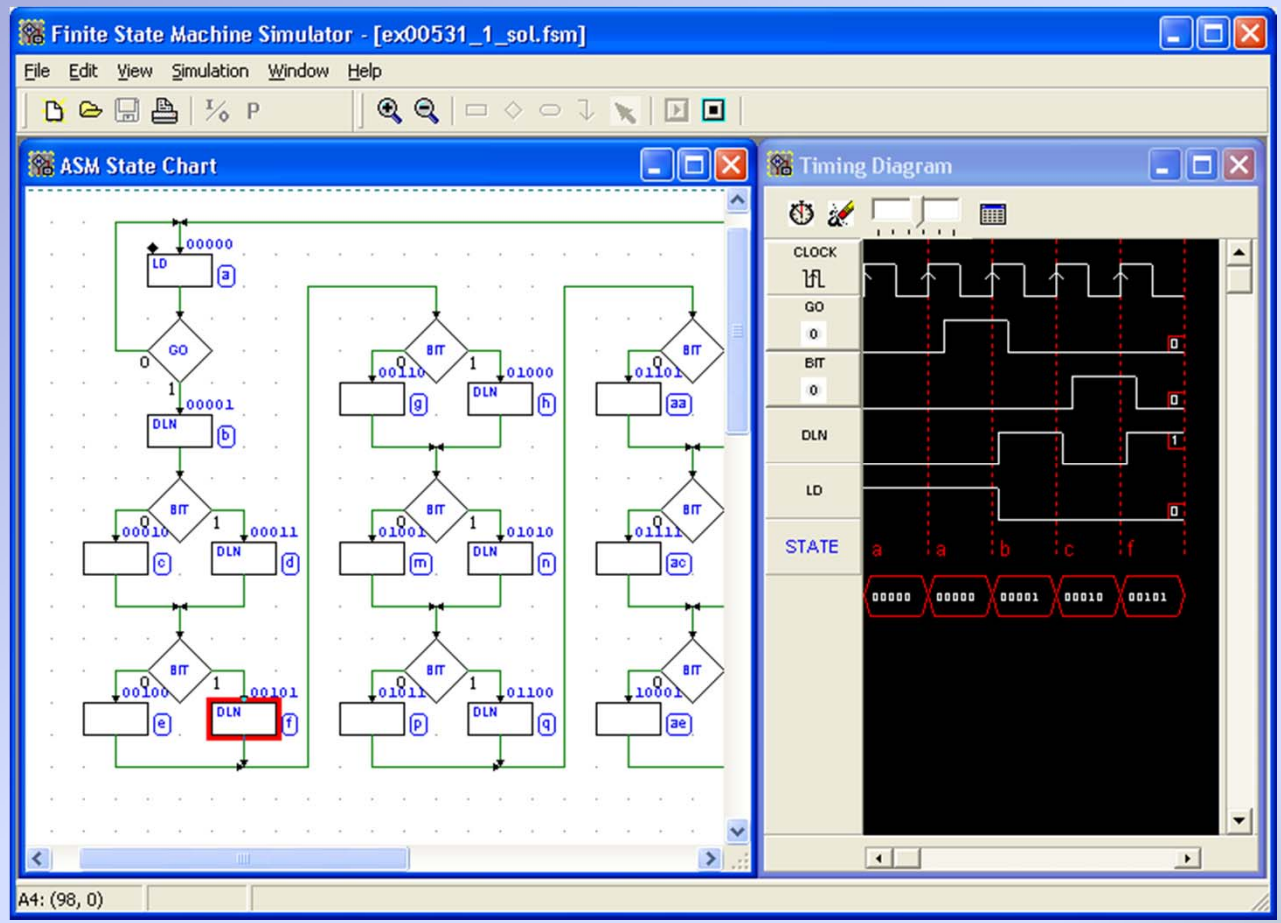
Deeds: the d-DcS Timing Diagram



- Timing diagram simulation is the most common tool to check circuits' functionality
- d-DcS Simulation features are similar to those of professional tools
- Clock and input signals can be easily edited
- Several input sequences can be saved with the circuit file.
- Saved sequences are useful for testing purposes and, in general, for educational applications

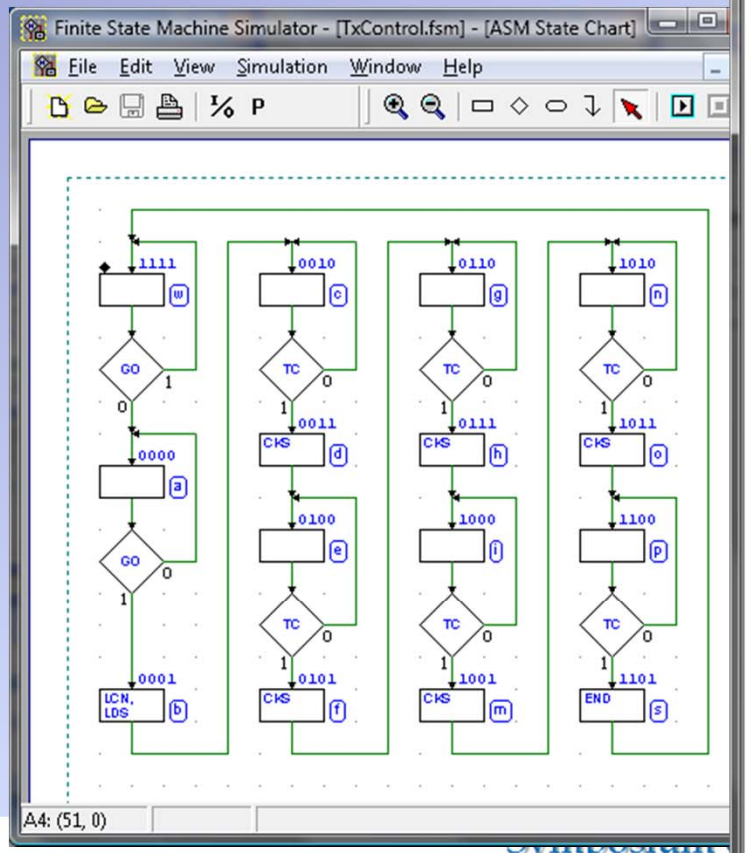
Deeds: the d-FsM Simulator

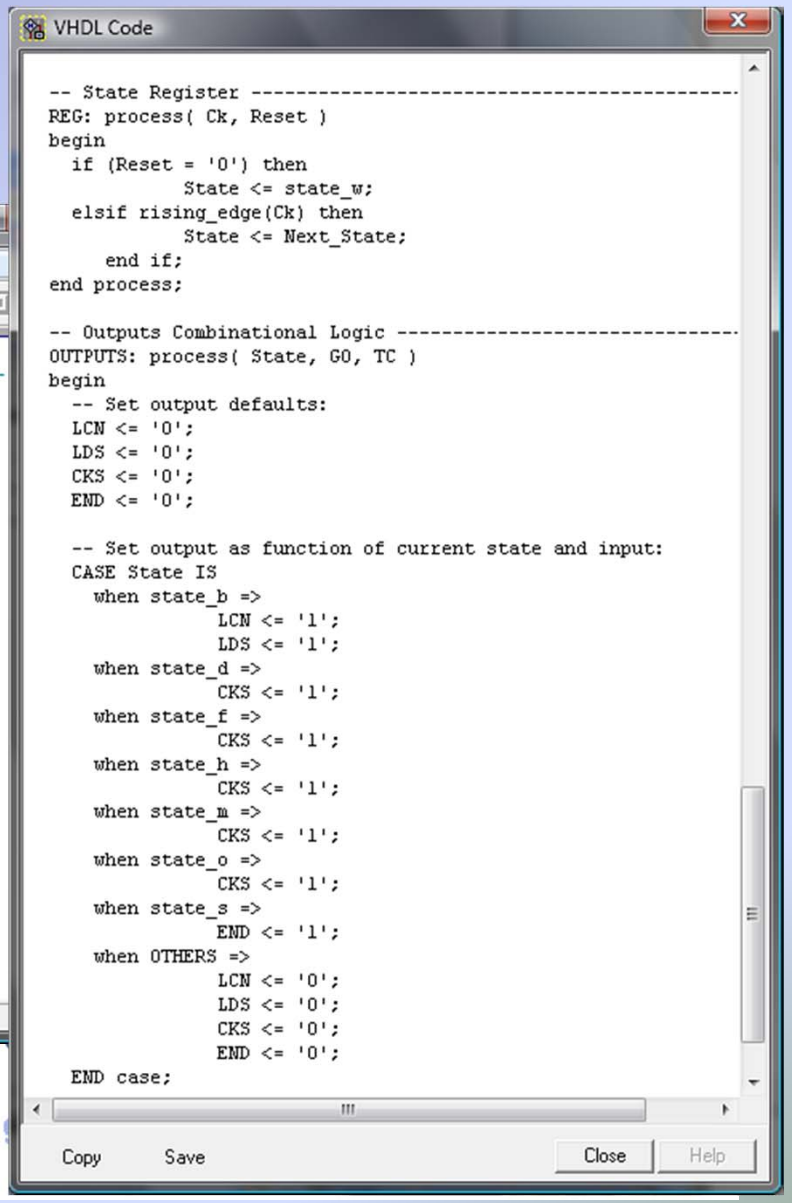
- Finite state machines (FSM) are designed with ASM charts
- Algorithms can be functionally tested (with a timing diagram), without circuit syntesis
- A FSM produced with the d-FsM tool can be used as a component by the d-DcS
- A FSM can also be exported in VHDL language, to allow reusing it in professional design tools



Deeds: the d-FsM to VHDL encoder

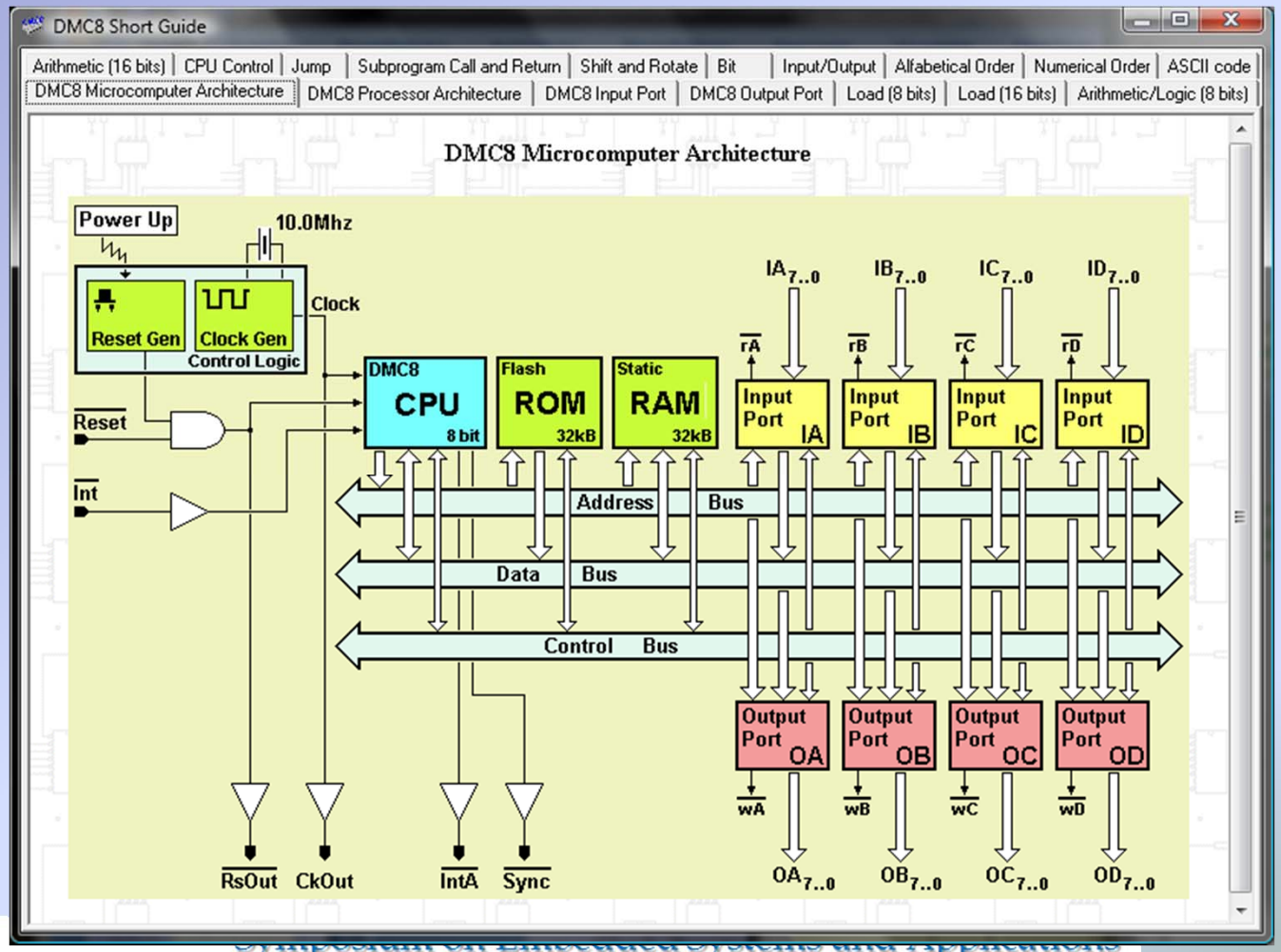
- A FSM can also be exported in VHDL language, to allow reusing it in professional design tools





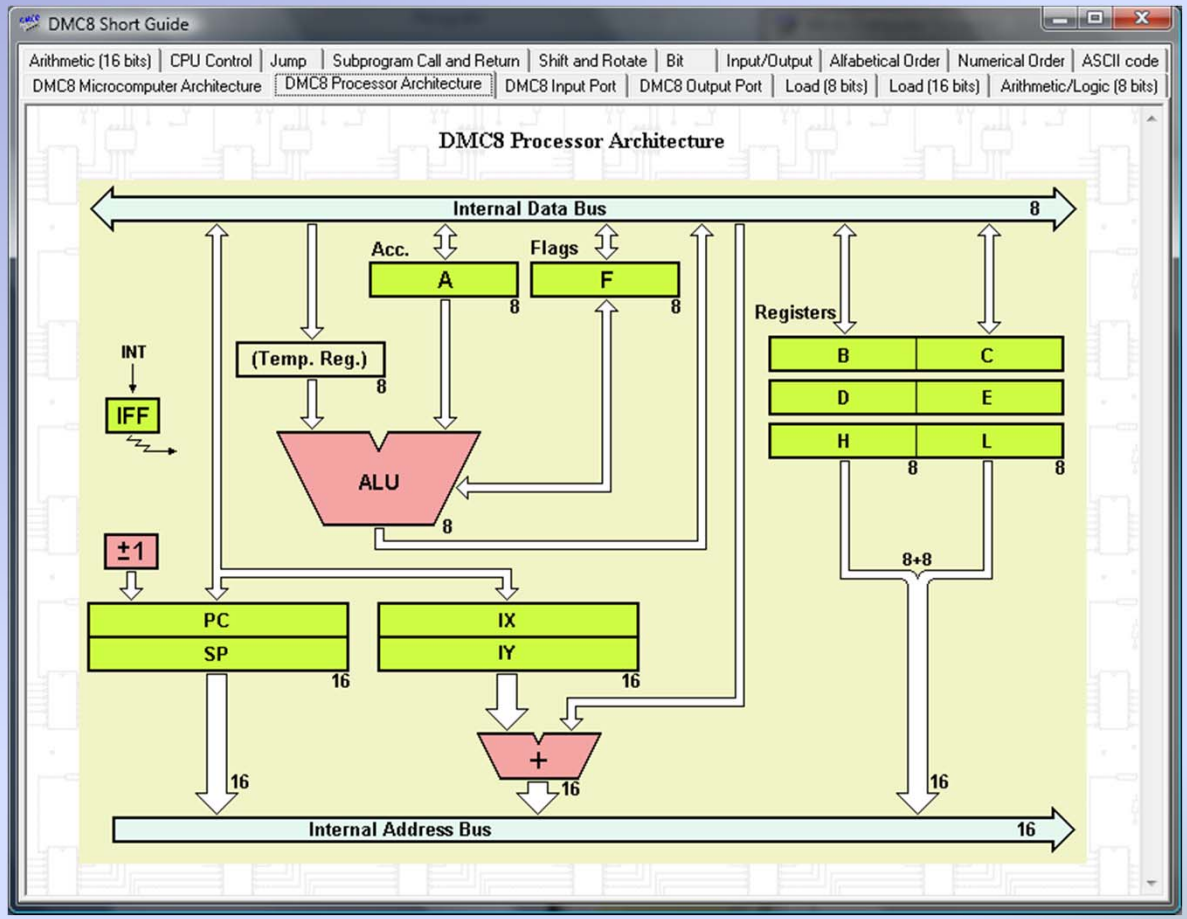
Deeds: the d-McE μC emulator

- The ***d-McE*** micro-computer emulator is based on an 8-bits microprocessor, RAM and ROM and a simple parallel input/output port system
- It interfaces the external world through four input and four output parallel ports
- Address and data busses are not available outside, but port control signals are available to extend the ports
- Clock, Reset and Interrupt signal are available



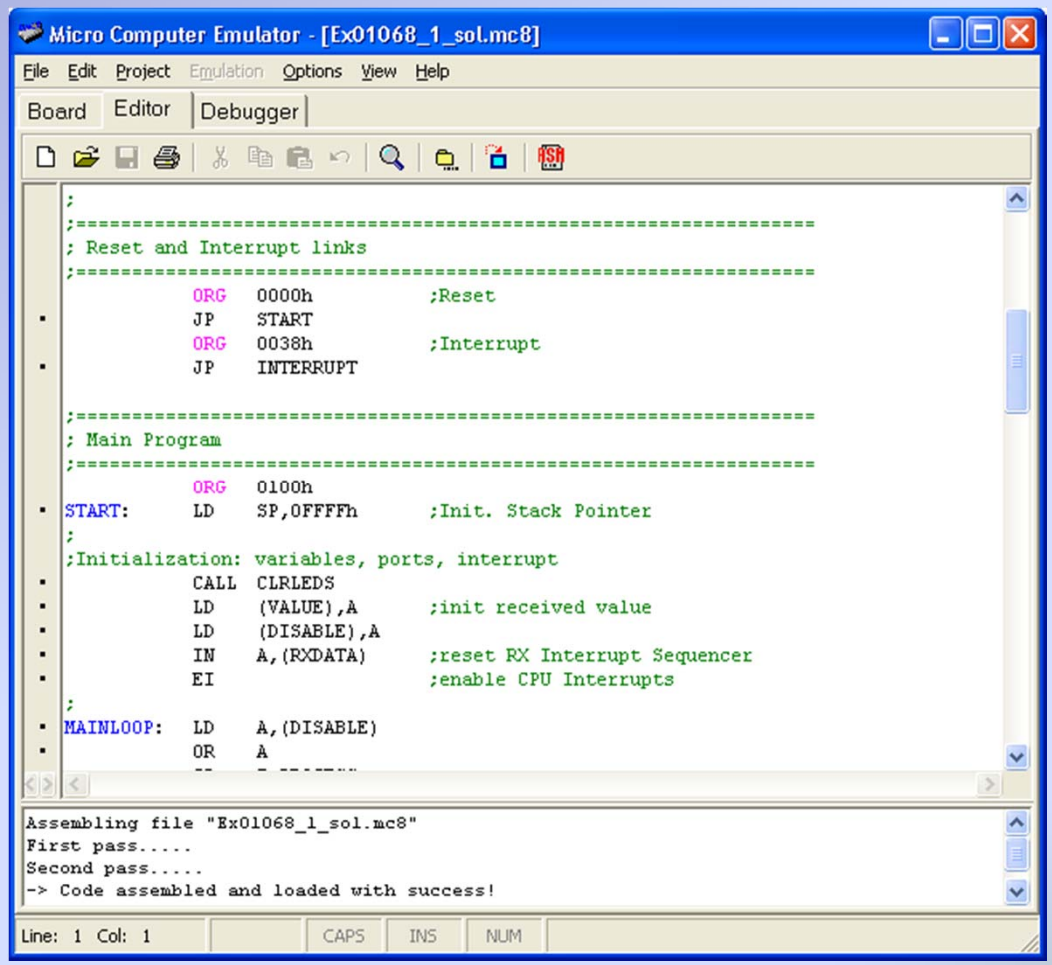
Deeds: the CPU architecture

- The 8-bit micro-processor is the **DMC8**
- **DMC8** emulates a simplified version of the well-known Z80-CPU
- The use of a "state of the art" CPU is not compatible with our pedagogical targets



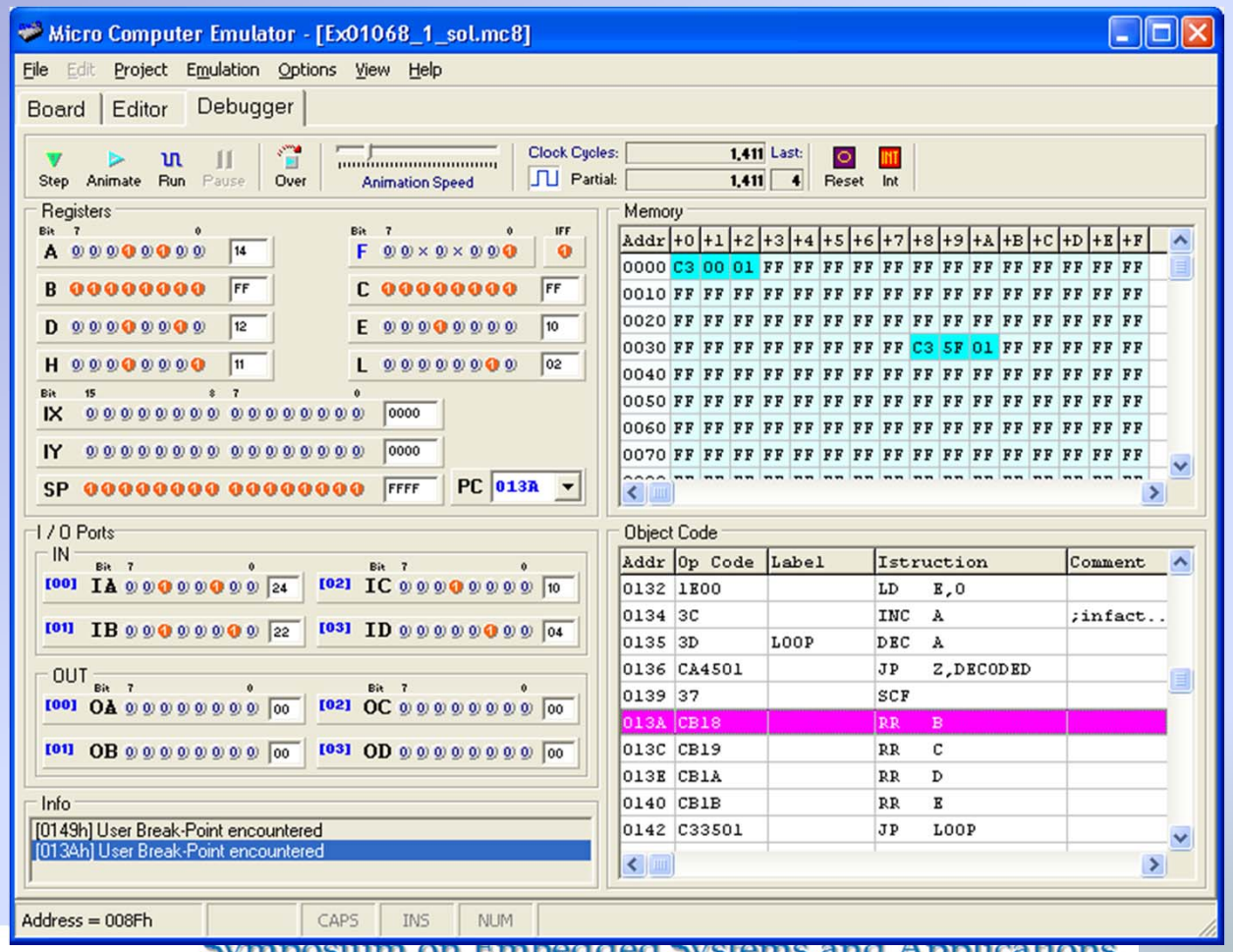
Deeds: the d-McE code editor

- The micro-computer emulator allows to edit assembly code with syntax highlighting
- Assembling, linking and loading operation are transparent to the user

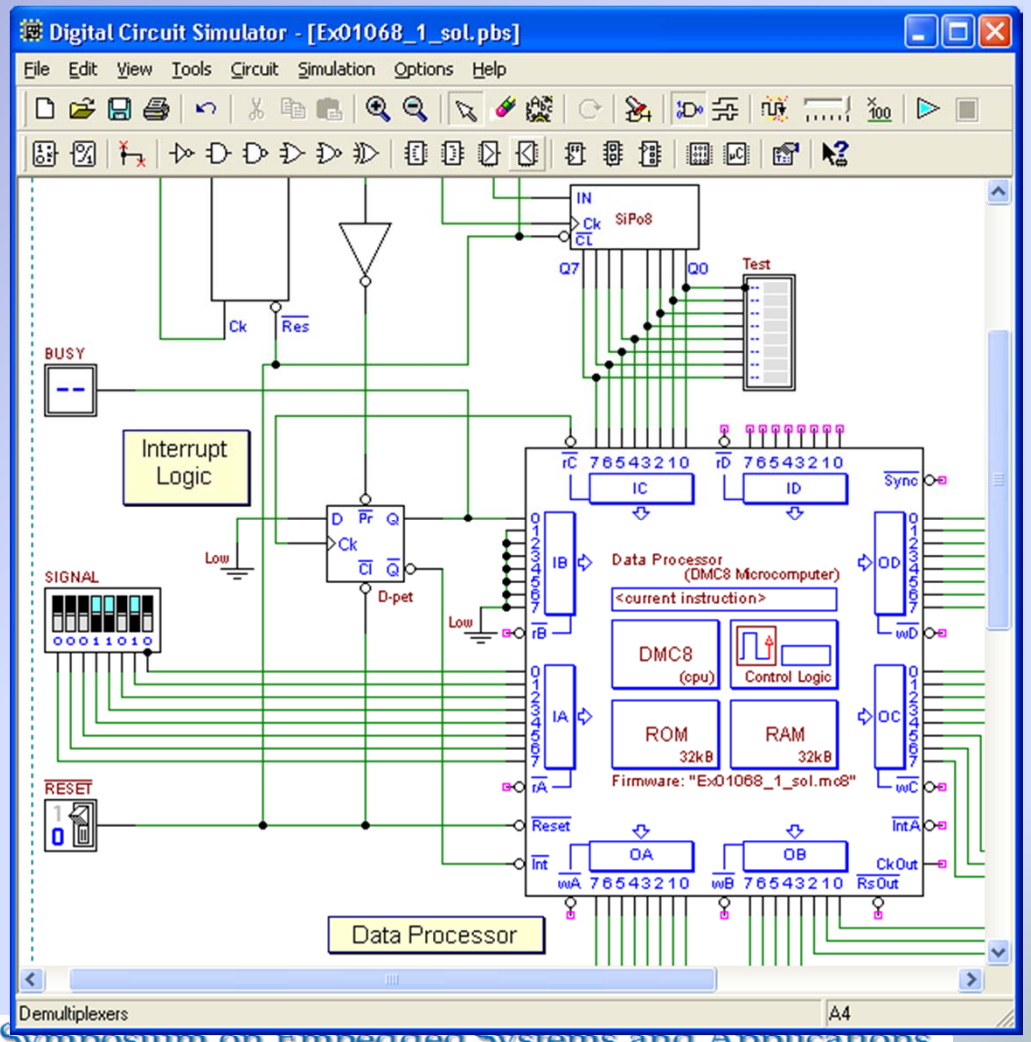


Deeds: the d-McE debugger

- The micro-computer emulator allows to debug the program step-by-step or in animation mode
- The interactive visual debugger shows memory, registers and ports contents
- The tool allows a full control of the microcomputer, including I/O operations



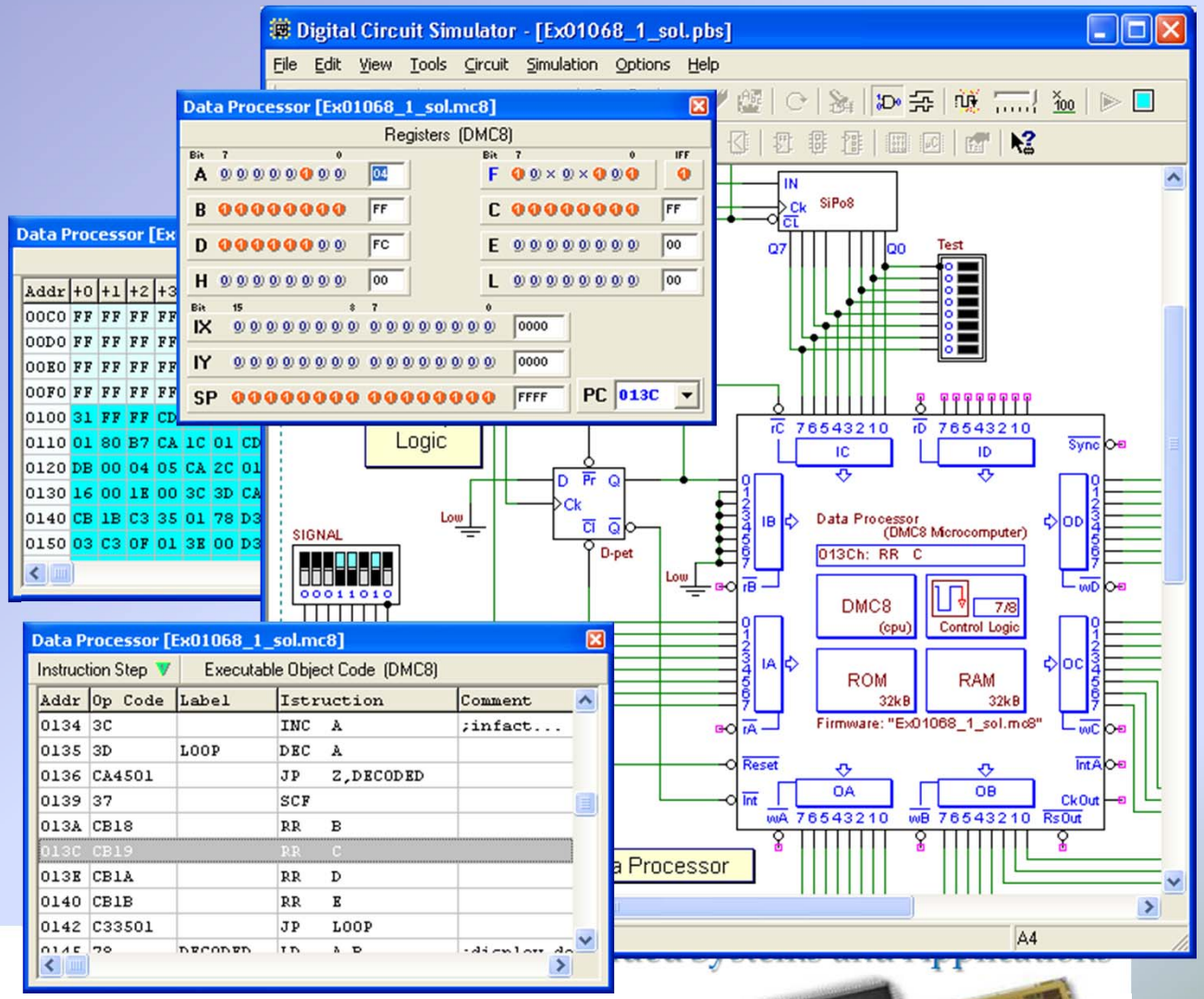
The micro-computer as d-DcS component



- The micro-computer is a component of the library of the d-DcS
- A digital circuit, embedding one or more micro-computers, can be simulated by the d-DcS
- The embedded micro-computer can be programmed with the d-McE tool.

The micro-computer in the d-DcS

- The functionality of the d-McE debugger is available in the d-DcS
- Memory, CPU registers, port status can be monitored during simulation of a digital system that embeds one or more micro-computer
- Through the debugger windows we can analyse the program logic while testing the hardware behaviour

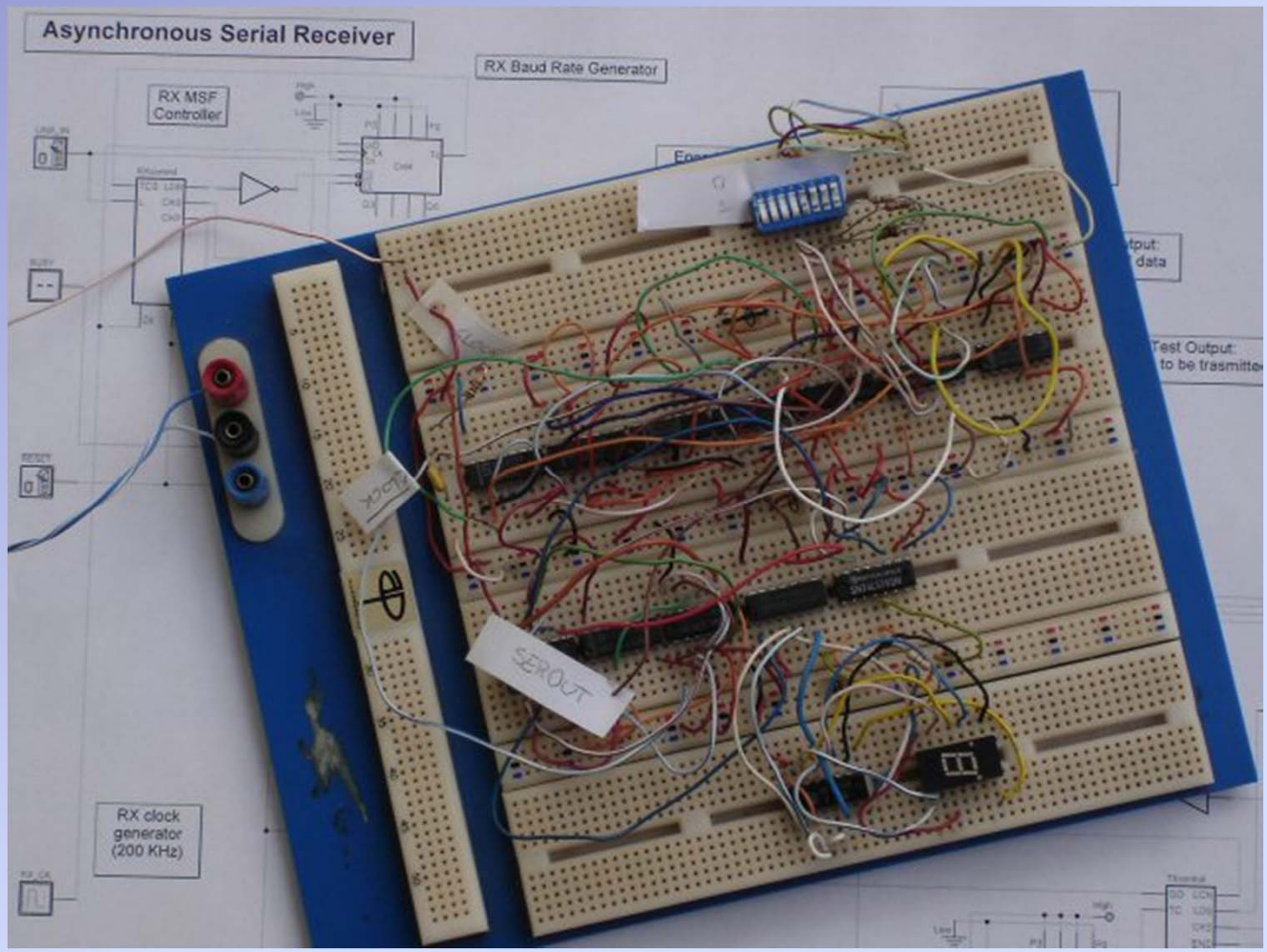


Deeds in practice at University of Genoa




- ***Deeds***, as support to traditional teaching, has been extensively used in our institution by thousands of students of the first and second year of the ***information engineering curricula***. This practice has been very successful, as demonstrated over the years by the evaluation procedures.
- As tool for Project Based Learning - only courses, ***Deeds*** has been used in conjunction with the NetPro (Network Based Project Learning), a European project of the Leonardo DaVinci programme, for both local and international courses.
- Students' and teachers' feedback has been very encouraging. Several colleagues from European universities have adopted Deeds in their teaching.

Not long time ago… a breadboard of a serial transmitter!

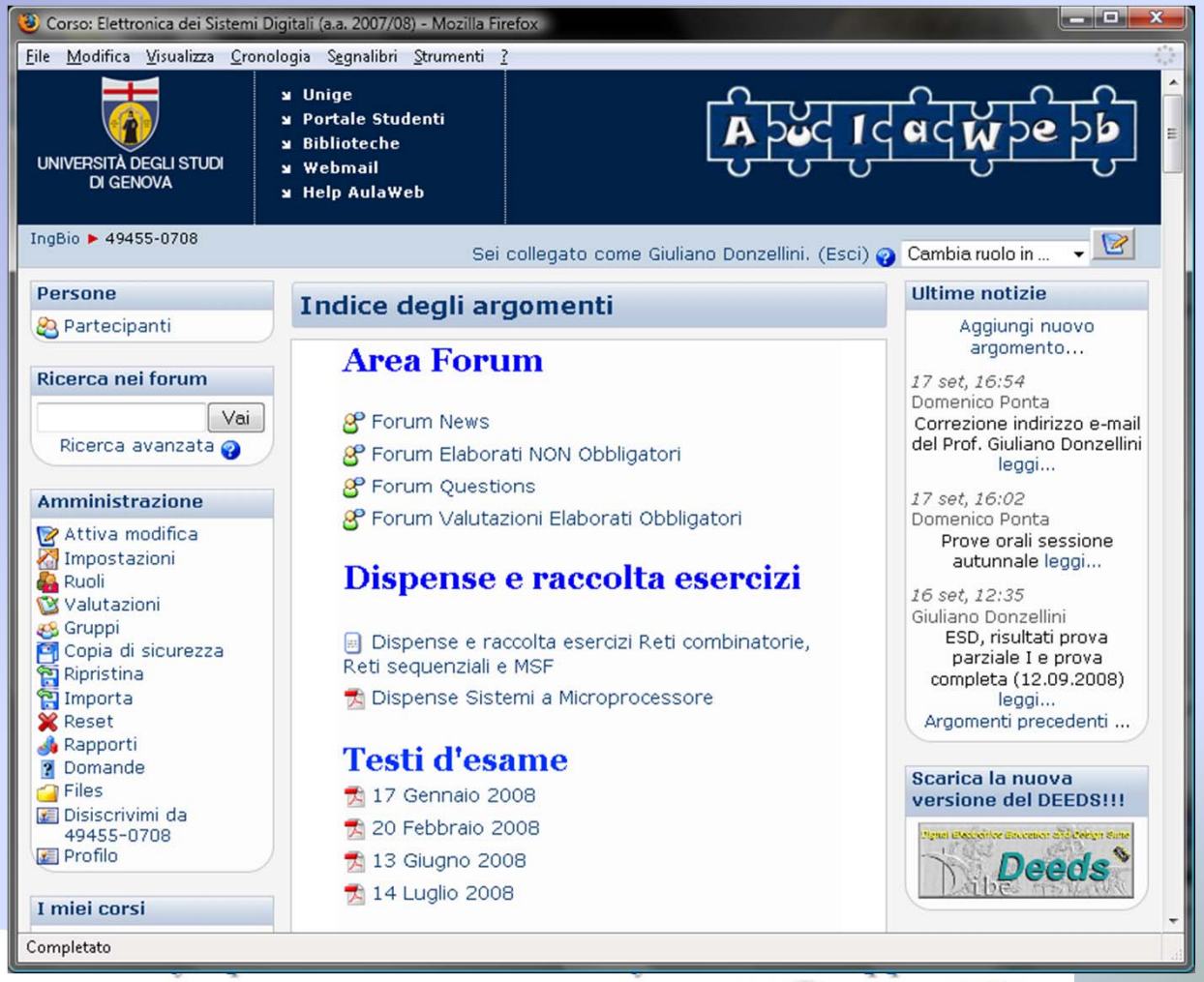


Blended learning with Deeds

- Our largest experience has taken place using Deeds in a blended learning environment, where traditional lectures coexist with a problem-based laboratory.
- The laboratory is delivered, at the same time, in a PC classroom, with tutorial assistance, or in distant mode, through Internet.
- Students can access the laboratory from home.
- In both cases the delivery is supported by a Learning Management System (NetPro at the beginning, then Moodle)

Moodle Learning Management System (LMS)

- The integration of ***Deeds*** material with a ***Learning Management System (LMS)*** provides added value for teachers and students alike.
- Teachers can keep track of students' activity, provide news and guidance, have access to the project deliverables and, generally, take advantage of the ***LMS*** features to manage the course.

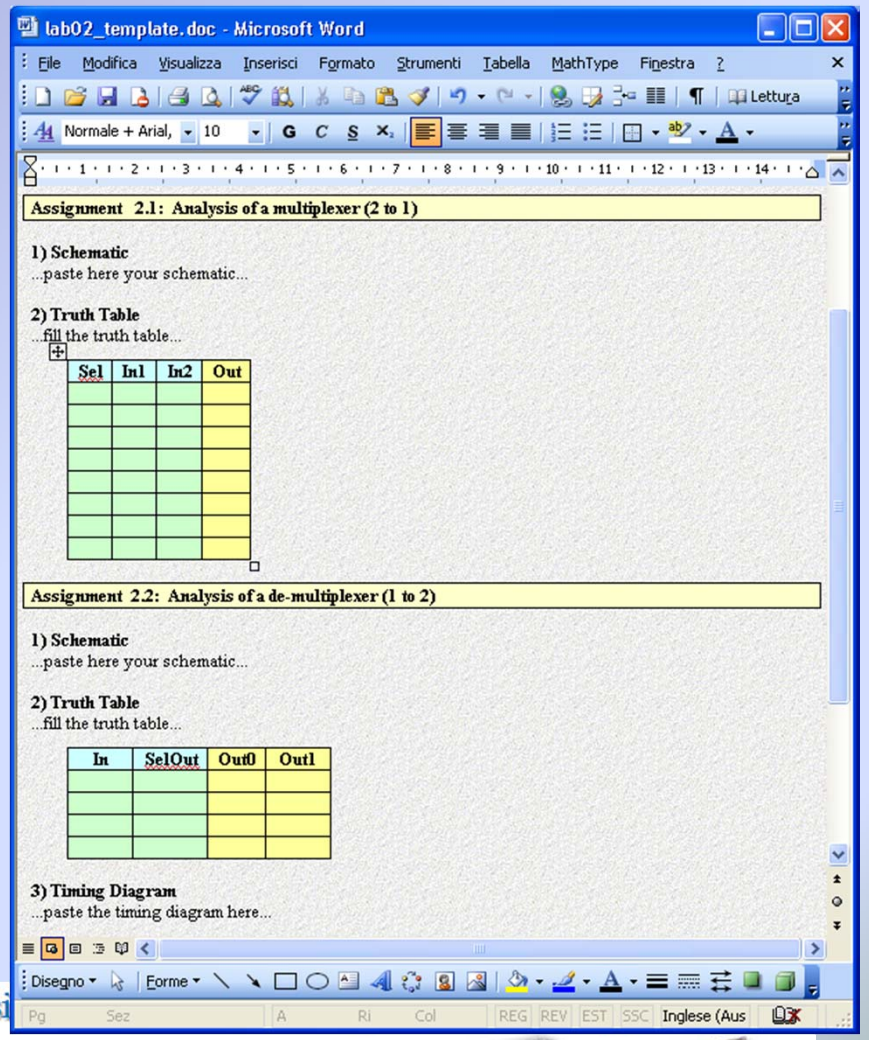


Moodle Learning Management System (LMS)

- ***Deeds Learning Material*** is delivered through the ***LMS*** that supports laboratory activity;
- Students get a large amount of flexibility in the execution of the projects, that may be local or remote, exchange information with their peers and get help by the teachers through the discussion forums.

Students' Project Report

- *"Templates"* that facilitate the preparation of the *laboratory report* are provided with each laboratory session
- Students find on the *report template*: truth tables & maps ready to be filled, space to paste circuit schematics, ASM diagrams, timing diagrams, comments and so on
- Reports are uploaded on the *Learning Management System*



Deeds is a resource for teachers

- Deeds is not a commercial product, it is free for academic institution
- Extensive learning material available in English
- Teachers can developed new ***project assignments*** with any web page editor

Deeds is a ***shared*** resource for teachers

- Deeds ***project assignments*** have been shared among European schools (within the European Union LeonardoDaVinci NetPro project)
- Sharing projects among teachers is very cost-effective, promotes cooperation and homogeneisation of courses and programs, encourages students exchange (Erasmus)
- ***Project assignments*** have been translated in other languages, last of which is ***Turkish***

Deeds on Turkish web pages…

Çizgi Söğüt Gölgesi : e-Akademi - Mozilla Firefox

File Modifica Visualizza Cronologia Segnalibri Strumenti ?

http://portal.cizgi.com.tr/library/category.aspx?id=15

Çizgi TAGEM • Çizgi Elektronik • Kullanıcı İşlemleri

Online Internet KÜTÜPHANESİ ...
... ve Uzaktan Eğitim Merkezi !

Forum • Haberler • e-Akademi • Güncel Yazılımlar • Köşe Yazıları • Diğer...

e-Akademi

e-akademi de ara...

Ana Sayfa > e-Akademi > Bilgisayar Teknojileri Eğitimi - Bilgisayar Mimarisi

Tübider Okul Bilişim Editör'ün Mesajı
İçerik: 1 Dosya, 12 Resim
Gönderen: Niyazi SARAL
Ekleme: 09 Ocak 2008 Çarşamba
Okuma: 751 (Günlük Ortalama 2,6)

Bilgisayar Mimarisi Eğitimi
Bilgisayar tasarımı, merkezi işlem birimi, aritmetik devre tasarımı, hafıza ve kontrol birimleri ile bir bilgisayar sisteminin katmanlı modeli
İçerik: 8 Sayfa
Gönderen: admin
Ekleme: 27 Mayıs 2008 Salı
Okuma: 2.310 (Günlük Ortalama 15,1)
Editör'ün Tercihi

Merkezi İşlem Birimi Emülatörü (PIPPIN)
Bilgisayar mimarisi temel çalışma biçimini gösteren sanal bir Merkezi İşlem Birimi
İçerik: 1 Sayfa
Gönderen: admin
Ekleme: 27 Mayıs 2008 Salı
Okuma: 466 (Günlük Ortalama 3,0)

CPU Simülatörü
Mikroişlemci Simülatörü
Gönderen: Mehmet Çelik
Ekleme: 10 Eylül 2007 Pazartesi
Okuma: 1.942 (Günlük Ortalama 4,7)

Completato

Deeds on Turkish web pages…

Çizgi Söğüt Gölgesi : e-Akademi - Mozilla Firefox

File Modifica Visualizza Cronologia Segnalibri Strumenti ?

http://portal.cizgi.com.tr/library/category.aspx?id=15

MIPS R2000 CPU Simülatörü
MIPS R2000 CPU Simülatörü
İçerik: 1 Sayfa, 1 Dosya, 5 Resim
Gönderen: Mehmet Çelik
Ekleme: 12 Eylül 2007 Çarşamba
Okuma: 1.166 (Günlük Ortalama 2,8)

SimHYMN Simülatörü
SİmHYMN simülatörü ile CPU'nun bir program parçasını nasıl simüle ettiğini görebilirsiniz.
İçerik: 1 Sayfa, 2 Dosya, 6 Resim
Gönderen: Mehmet ÇELİK
Ekleme: 06 Aralık 2007 Perşembe
Okuma: 403 (Günlük Ortalama 1,2)

Logisim
Sayısal Devreler Simülatörü
İçerik: 41 Sayfa, 3 Dosya, 12 Resim
Gönderen: Mehmet Çelik
Ekleme: 14 Eylül 2007 Cuma
Okuma: 4.677 (Günlük Ortalama 11,4)
Editör'ün Tercihi

Logisim ile Bilgisayar Mimarisi Dersi
Jean REBIFFE (İngilizce)
İçerik: 12 Dosya, 12 Resim
Gönderen: Mehmet ÇELİK
Ekleme: 27 Kasım 2007 Salı
Okuma: 784 (Günlük Ortalama 2,3)
Editör'ün Tercihi

Deeds Simülatörü
Deeds, dijital elektronik sistemleri için araç ve öğrenim materyali sunan bir simulatör setidir.
İçerik: 52 Sayfa
Gönderen: admin
Ekleme: 28 Mayıs 2008 Çarşamba
Okuma: 3.318 (Günlük Ortalama 21,8)

Hades Simülatörü
Hades hem temel sayısal sistem tasarımını öğretmek hem de sistem simülasyonu ve donanım/yazılım benzetimi üzerine çalışmalar için bir araç olarak kullanılır.
İçerik: 25 Sayfa
Gönderen: admin
Ekleme: 29 Mayıs 2008 Perşembe
Okuma: 2.082 (Günlük Ortalama 13,8)

Completato

Deeds on Turkish web pages…

Deeds Learning Material in Turkish: partial view

Çizgi Söğüt Gölgesi : e-Akademi - Mozilla Firefox

File Modifica Visualizza Cronologia Segnalibri Strumenti ?

http://portal.cizgi.com.tr/library/topic.aspx?id=1131

Digital Electronics Education and Design Suite

Deeds (Digital Electronics Education and Design Suite) Sayısal elektronik için eğitim araçları setidir, aktif eğitim yöntemi (uygulayarak öğrenme) olarak tanımlanır. Deeds içindeki simülatör dijital elektronikteki şu alanları kapsar: bileşimli mantık ağlar(from simple gates to decoders, encoders, multiplexers and demultiplexers); sıralı mantık ağlar(from simple flip-flops to registers and counters) sonlu durum makine dizaynı; mikro-bilgisayar programlama (assembly düzeyi) ve arayüz oluşturma. Deeds araçları karmaşık ağların ve gömülü sistemlerin tasarımını ve benzetimini sağlar. Deeds ESNG, DIBE, University of Genoa, Italy tarafından geliştirilmektedir.

Bu içeriğin telif hakları korunmaktadır. © 2002-2008 by Giuliano Donzellini, Domenico Ponta, Davide Anguita

1. Sayfa

Sonraki: Deeds Dijital Devre Simulasyonuna Giriş

İçindekiler

Completato

Deeds Learning Material : an example

Çizgi Söğüt Gölgesi : e-Akademi - Mozilla Firefox

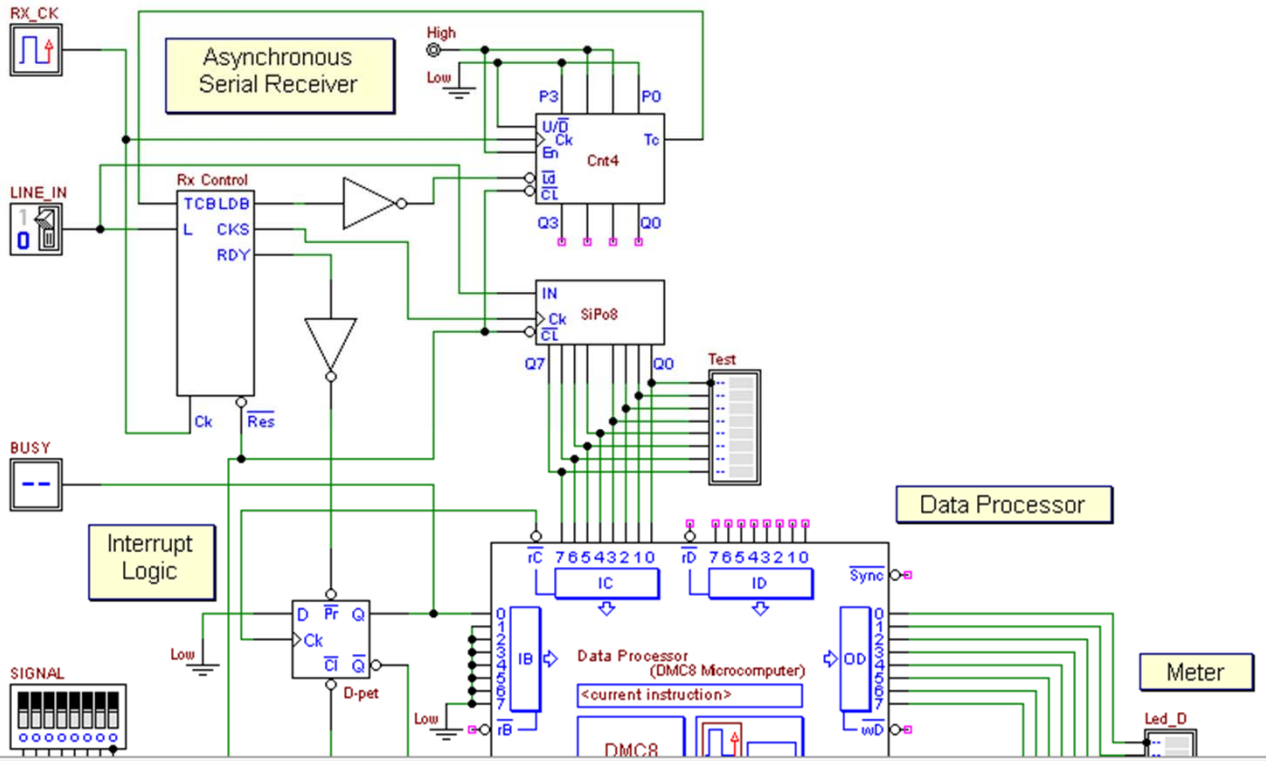
File Modifica Visualizza Cronologia Segnalibri Strumenti ?

http://portal.cizgi.com.tr/library/topic.aspx?id=1131&page=48

Digital Sayaç

Bu proje için Deeds simulatörünün bilgisayarınızda yüklü olması gerekmektedir. Deeds'i indirmek için buraya tıklayınız.

Bu alıştırmada **Dijital Sayaç** üzerinde çalışacaksınız. Aşağıdaki şekilde şematik yapıyı görebilirsiniz (şemayı editörde açmak için şeklin üstüne tıklayınız).



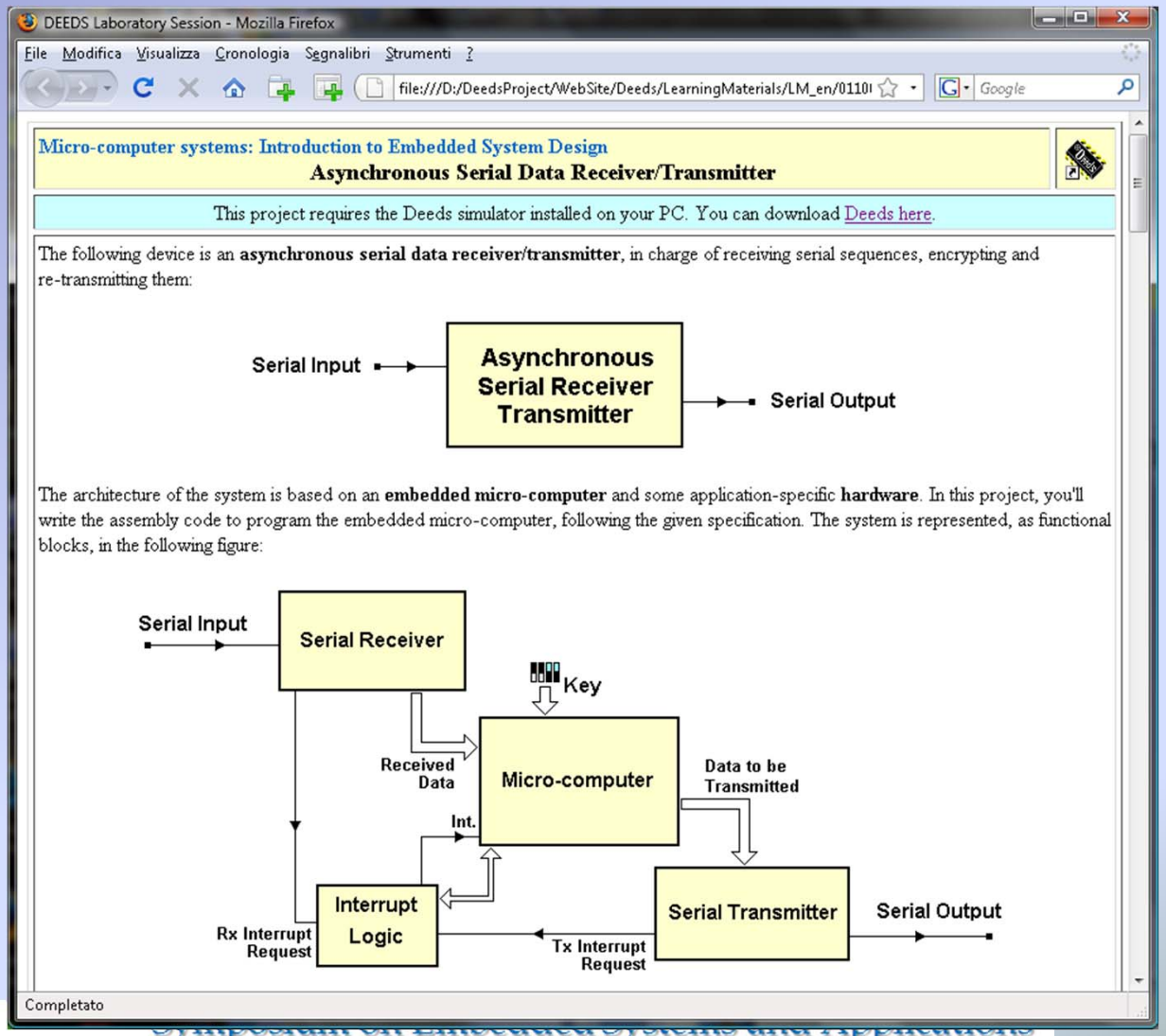
Completato

Deeds to introduce Embedded System Design

- Let's assume that students have familiarised themelves with analysis and design of systems based on ***standard logic circuits***, ***finite state machines*** and a ***microcomputer.***
- The knowledge and skills gained in the previous phases are necessary to understand and design embedded systems in a ***bottom-up approach***.
- We believe that the task of ***introducing*** embedded systems can be made easier by starting with the analysis of a pedagogically-oriented implementation.
- On the following slides, we present an exercise where students are facing a mixed work of ***analysis*** of the hardware structure and ***design*** of the microcomputer controlling software.

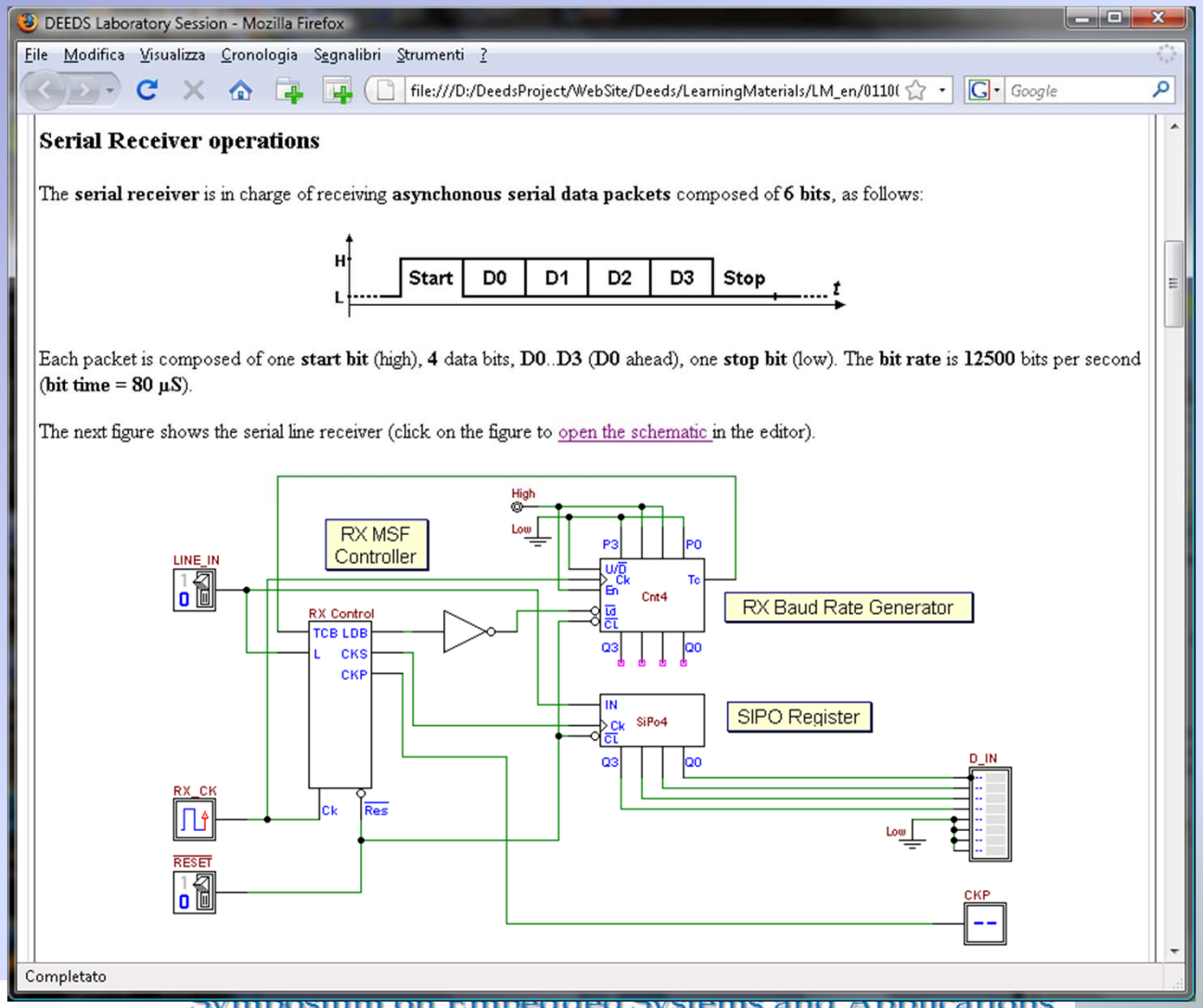
Embedded System: a project assignment

- The system is a Asynchronous Serial Data Receiver – Transmitter.
- The receiver and the transmitter are implemented by hardware.
- The micro-computer buffers and encrypts data.
- The system hardware is given; the software must be written.
- The target is to understand dynamics and interactions among modules of an ***interrupt-driven system.***



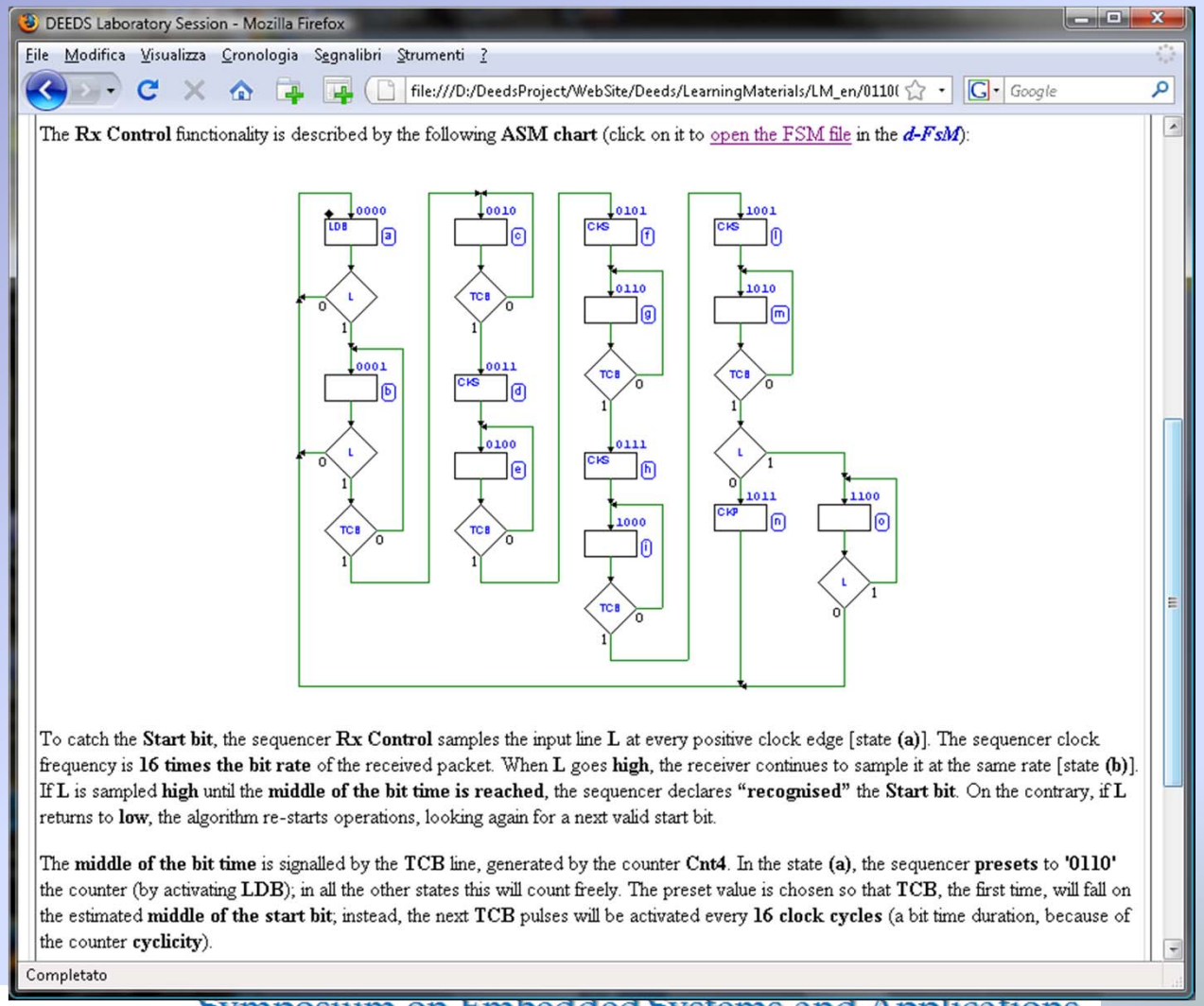
The asynchronous serial receiver

- The assignment explains the operation of the serial receiver
- The receiver system is built around a sequencer controlling a counter and a shift register
- The schematic must be opened with the ***d-DcS simulator***
- Students should understand the behaviour of the system



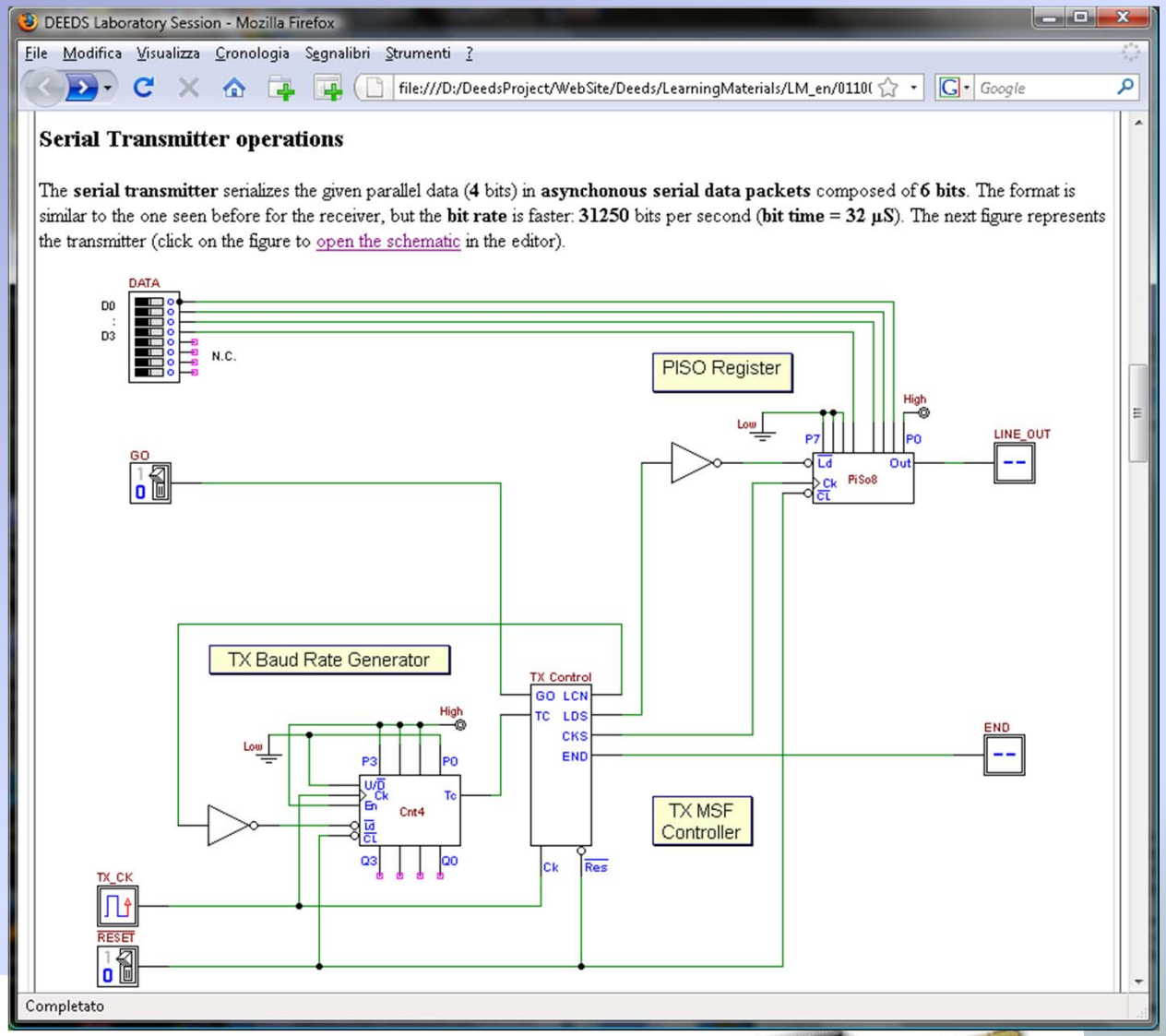
The serial receiver controller

- The assignment provides the ***ASM chart*** of the ***state machine*** controlling the receiver
- The algorithm can be opened in the ***d-FsM simulator***
- The analysis of the algorithm is essential to understand the system timing
- The ***d-DcS simulator*** is used to complete the analysis of the receiver module



The asynchronous serial transmitter

- The text details the operation of the transmitter module
- The transmitter is sequenced by a state machine controller
- Students will use ***Deeds*** to study the system as in the case of the receiver



The serial transmitter controller

- Also for the transmitter, the text provides the ***ASM chart*** of the controlling ***state machine***
- The analysis of the transmitter module is carried out with the ***d-FsM simulator*** and ***d-DcS simulator***

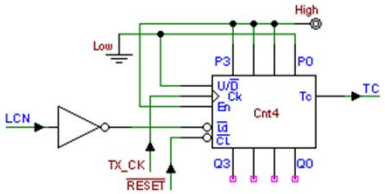
DEEDS Laboratory Session - Mozilla Firefox

File Modifica Visualizza Cronologia Segnalibri Strumenti ?

file:///D:/DeedsProject/WebSite/Deeds/LearningMaterials/LM_

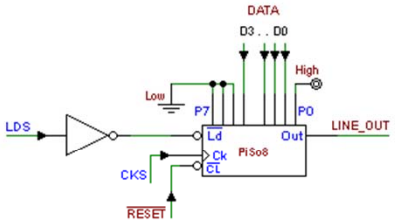
Asynchronous Serial Data Transmitter functionality

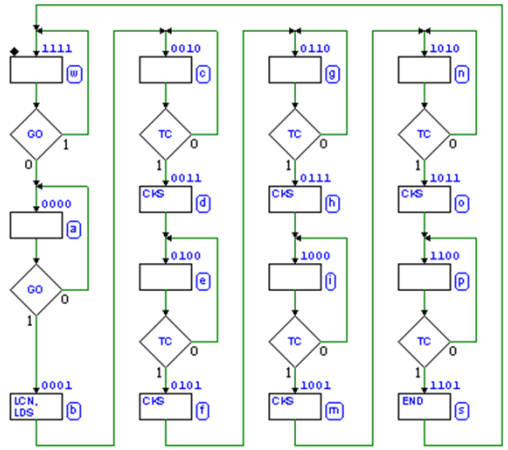
The transmitter is composed by the **Cnt4** counter, the **SiPo8** shift register and the **Tx Control** sequencer.



The binary counter **Cnt4** is synchronized by the **TX_CK** clock at **500 KHz**. The counter is cabled to count down cyclically. Every time the outputs **Q3..Q0** reach the number **'0000'**, the output **TC** (**terminal count**) is activated, resulting in a pulse on **TC** every **16** clock cycles. The cyclic activation of **TC** is used by the **TX Control** sequencer to synchronize the transmission of data bits at the given bit/rate (31250 b/s = 500 KHz / 16). When the **LCN** signal is activated by the sequencer, the counter is preset to the value **'1110'** (from on the inputs **P3..P0**) to begin the transmission sequence (see the description of the **TX Control** sequencer).

The **PiSo8** register, when the signal **LDS** is activated, loads in parallel the four data bits **D3..D0**, together with a bit set to **'1'** (**P0**, the **Start** bit) and the others to **'0'** (including the **Stop** bit, **P5**). At the same time, the load operation starts the transmission, since it sets **LINE_OUT** to **'1'**, as **Start** bit of the serial sequence. Then, every time the register clock (**CKS**) is pulsed, the bits are shifted "right", so transmitting the next bit of the data packet on **LINE_OUT**.





The **TX Control** sequencer functionality is described by the ASM chart reported on the side (click on it to open the FSM file in the ***d-FsM***). The first two states, **(w)** and **(a)** are in charge of waiting for the positive edge of the **GO** command. In state **(b)**, **LCN** presets the counter **Cnt4** and **LDS** loads the data in the **PiSo8** shift register. The couple of states **(c)** and **(d)** wait for **TC** from the counter, i.e. waits **32 µS** before activating **CKS**, the clock of the shift register.

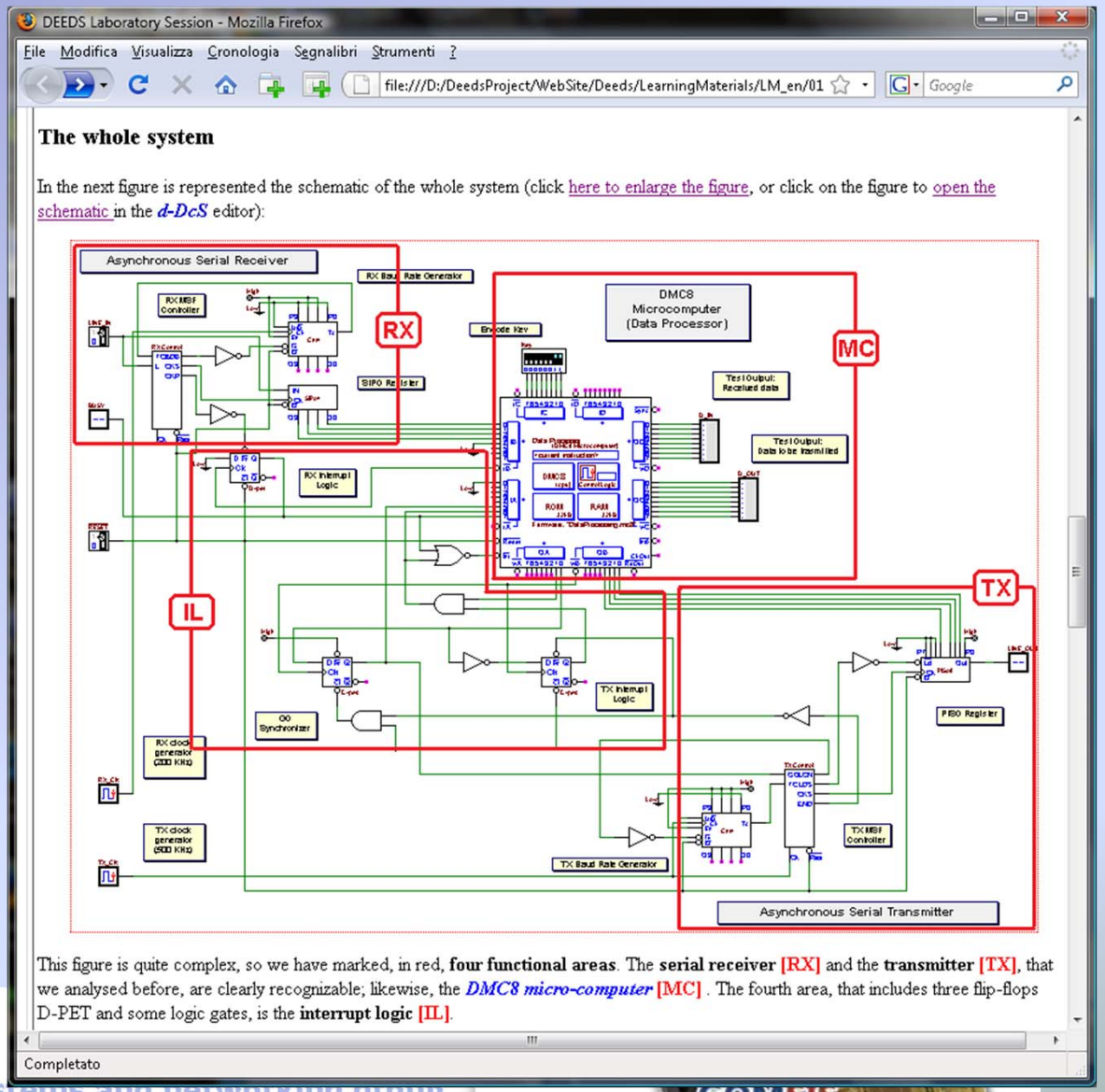
State sequence from **(e)** to **(o)** repeats four times the same task, and the shift register is progressively emptied (data is transmitted bit after bit, each one every **32 µS**).

After the transmission of the stop bit, the output **END** is generated **(s)** to signal the end of transmission.

Completato

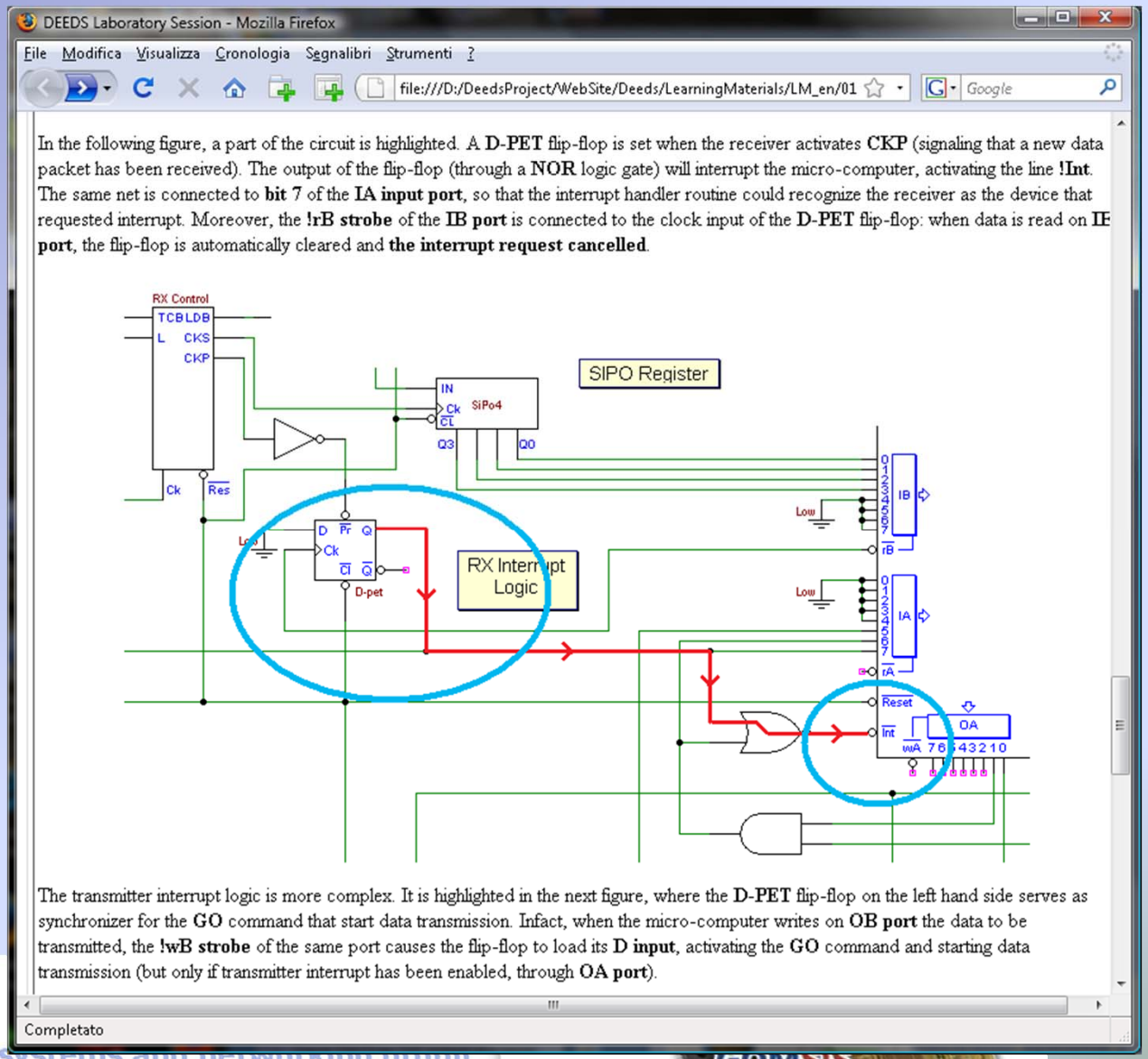
The complete system

- The whole system is presented as schematic, divided into logical blocks.
- The ***embedded micro-computer*** **(MC)** links the receiver and trasmitter modules **(RX and TX)** and controls data flow.
- The interrupt logic block **(IL)** is a **key element** of the project



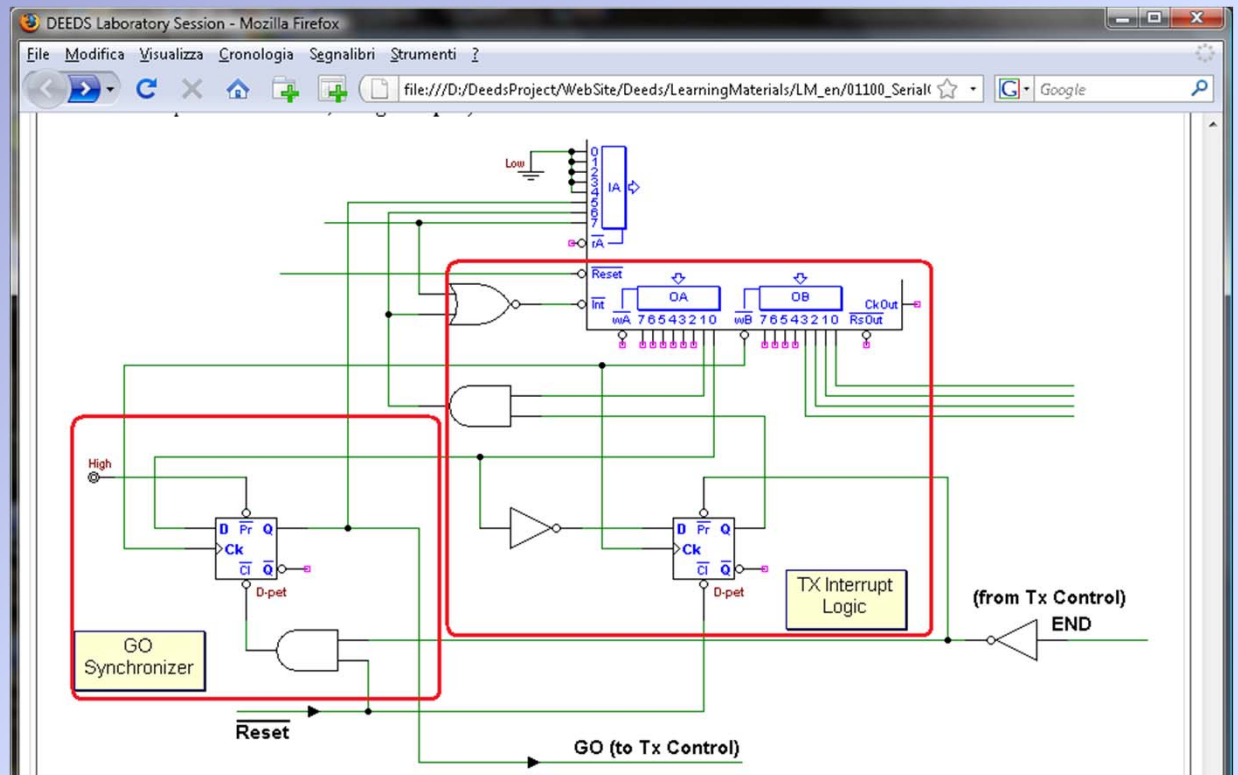
The receiver interrupt controller

- The text explains the purpose and operation of the receiver interrupt logic.
- The understanding of the interrupt controller logic behavior is essential for developing the micro-computer program



The transmitter interrupt controller

- The text continues with the explanation of the operation of the transmitter interrupt logic.
- This circuit is more complex than the previous one.
- The student learns how to deal with a programmable interrupt controller.
- A full understanding of the interrupt operation could be difficult at this stage.



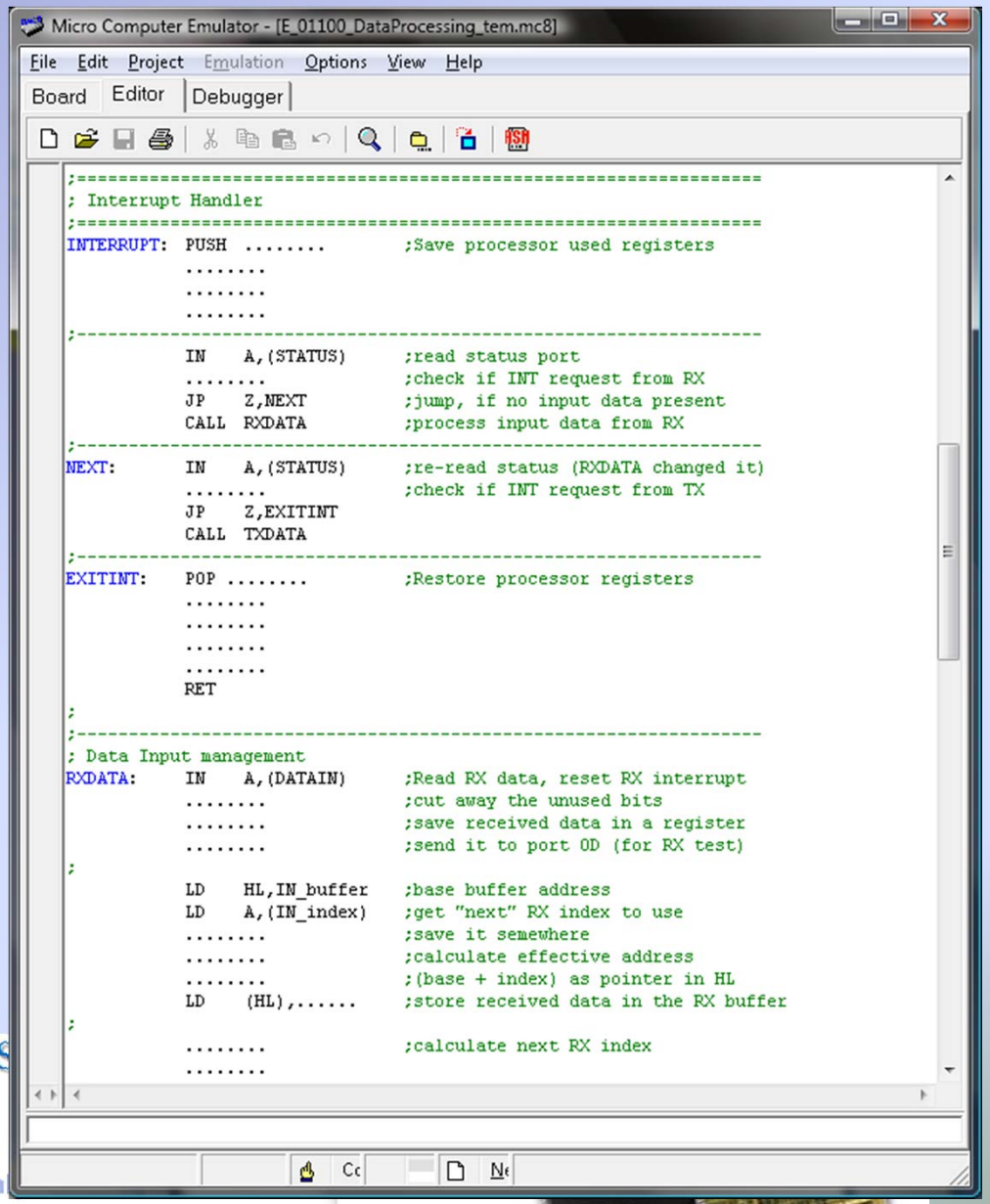
The other **D-PET** flop-flop is clocked in same way, when **OB port** is written. However, this operation serves normally to clear the flip-flop. Infact the transmitter, when the transmission process terminates, activates the **END** signal that, in turn, sets the flip-flop, requesting interrupt to the micro-computer. The same request is connected to **bit 6** of the **IA input port**, so that the interrupt handler routine could recognize the transmitter as author of the request.

When interrupted by the transmitter, the micro-computer will send another byte to the **OB port** and this operation, at the same time: **a)** will start the transmission (activating **GO**), **b)** will clear the interrupt request; **c)** prepare the new interrupt request that will be generated on the end of transmission. **OA port** permits also to set the interrupt logic in a particulare mode that enables the generation of an interrupt request the first time we need to transmit data.

Completato

Programming the embedded processor (1)

- The system analysis is over.
- Next stage is the writing of the embedded processor code.
- Assembly language programming allows a full understanding and control of the system.
- At this stage, starting from scratch could be very difficult for the majority of students.
- We support the learner by providing a commented trace of the program.
- A few “critical” sequences of instructions are provided in the trace.



Programming the embedded processor (2)

- The finished program, coded following the given trace, must be ***compiled and loaded*** into the micro-computer memory.
- The completed system is now ready for testing.

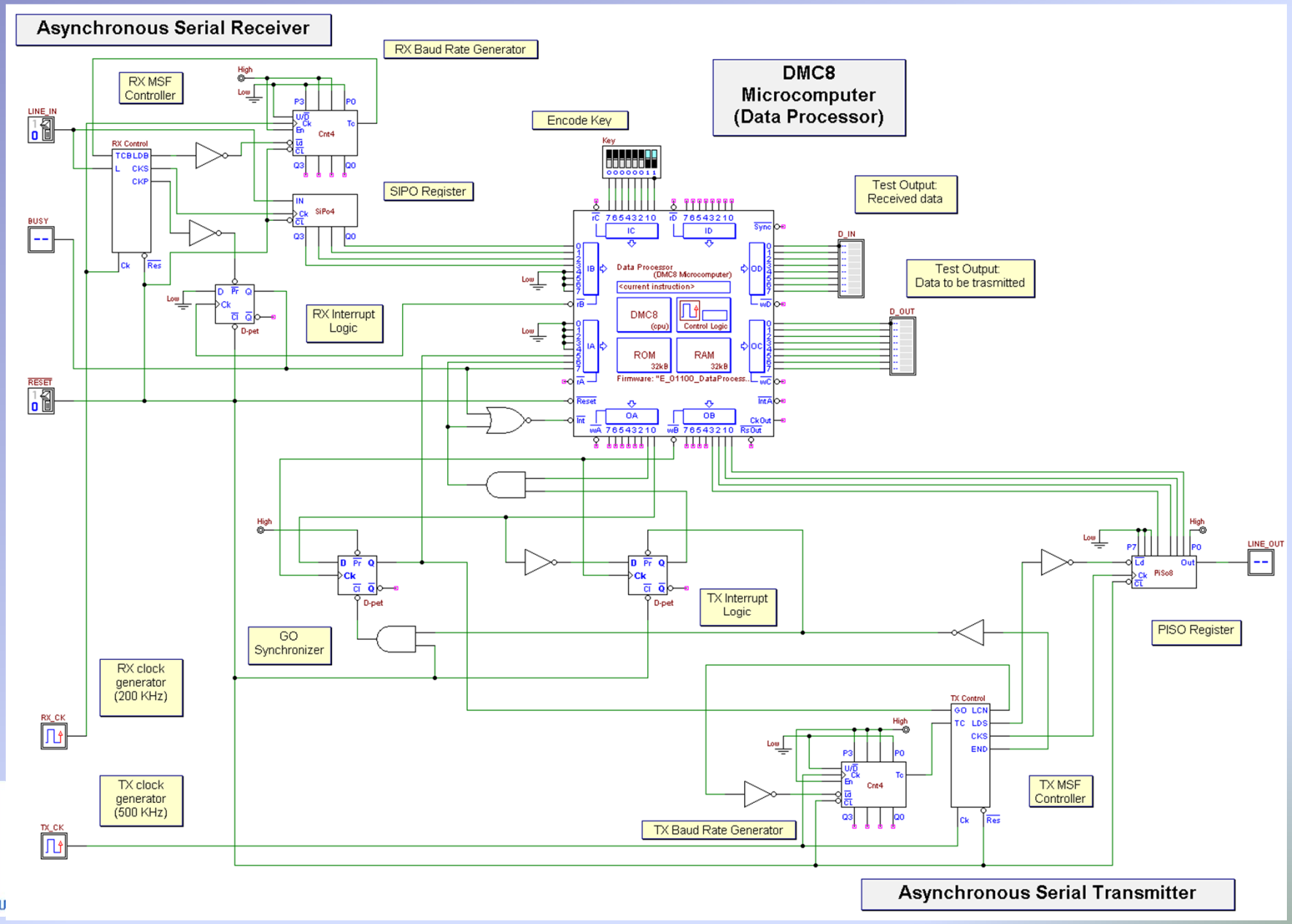
Micro Computer Emulator - [E_01100_DataProcessing_Sol.mc8]

File Edit Project Emulation Options View Help

Board Editor Debugger

```
;==================================================================
; Interrupt Handler
;==================================================================
INTERRUPT:  PUSH  AF            ;Save processor used registers
            PUSH  BC
            PUSH  DE
            PUSH  HL
;------------------------------------------------------------------
            IN    A,(STATUS)    ;read status port
            BIT   7,A           ;check if INT request from RX
            JP    Z,NEXT        ;jump, if no input data present
            CALL  RXDATA        ;process input data from RX
;------------------------------------------------------------------
NEXT:       IN    A,(STATUS)    ;re-read status (RXDATA changed it)
            BIT   6,A           ;check if INT request from TX
            JP    Z,EXITINT
            CALL  TXDATA
;------------------------------------------------------------------
EXITINT:    POP   HL            ;Restore processor registers
            POP   DE
            POP   BC
            POP   AF
            EI
            RET
;
;------------------------------------------------------------------
; Data Input management
RXDATA:     IN    A,(DATAIN)    ;Read RX data, reset RX interrupt
            AND   00001111b     ;cut away the unused bits
            LD    C,A           ;save received data in C register
            OUT   (RXTEST),A    ;send it to port 0D (for RX test)
;
            LD    HL,IN_buffer  ;base buffer address
            LD    A,(IN_index)  ;get "next" RX index to use
            LD    D,A           ;save it in D
            ADD   A,L           ;calculate effective address
            LD    L,A           ;(base + index) as pointer in HL
            LD    (HL),C        ;store data in the RX buffer
;
            LD    A,D           ;calculate next RX index
            INC   A
            CP    maxcount      ;check it against max count
            JP    NZ,EXITRX     ;lower than max, jump
;
```

The complete embedded system under test



Testing the system: timing diagram analysis

- The timing diagram analysis, in relation with CPU state and code execution

d-DcS - Timing Diagram - (Timing Interval Simulation mode)

RESET
Data Proce...
Clock
Int
IntA
IA Port
IB Port
rB (IB)
OA Port
OB Port
wB (OB)
RX_CK
LINE_IN
RX Control
State
TCB
LDB
CKS
CKP
BUSY
D_IN
Key
TX_CK
TX Control
State
GO
TC
LCN
LDS
CKS
END

Data Processor [E_01100_DataProcessing_Sol.mc8]

Registers (DMC8)

A	05	F	0
B	00	C	00
D	02	E	00
H	80	L	12
IX	0000		
IY	0000		
SP	FFF3	PC	0194

Data Processor [E_01100_DataProcessing_Sol.mc8]

Executable Object Code (DMC8)

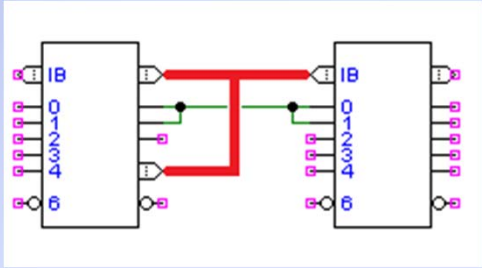
Addr	Op Code	Label	Istruction	Comment
018B	3A1480		LD A,(OUT_INDEX	;get "next" TX index to use
018E	57		LD D,A	;save it in D
018F	85		ADD A,L	;calculate effective address
0190	6F		LD L,A	;(base + index) as pointer in
0191	7E		LD A,(HL)	;get data to transmit
0192	D302		OUT (TXTEST),A	;send it to port 0C (for TX t
0194	D301		OUT (DATAOUT),A	;send it to the transmitter
0196	7A		LD A,D	;eval next TX index
0197	3C		INC A	
0198	FE04		CP MAXCOUNT	;check it against max count
019A	C2A101		JP NZ,EXITTX	;lower, jump
019D	3E00		LD A,00H	
019F	D300		OUT (TXCTRL),A	;disable TX engine
01A1	321480	EXITTX	LD (OUT_INDEX),	;update TX index in memory
01A4	C9		RET	;return to the caller

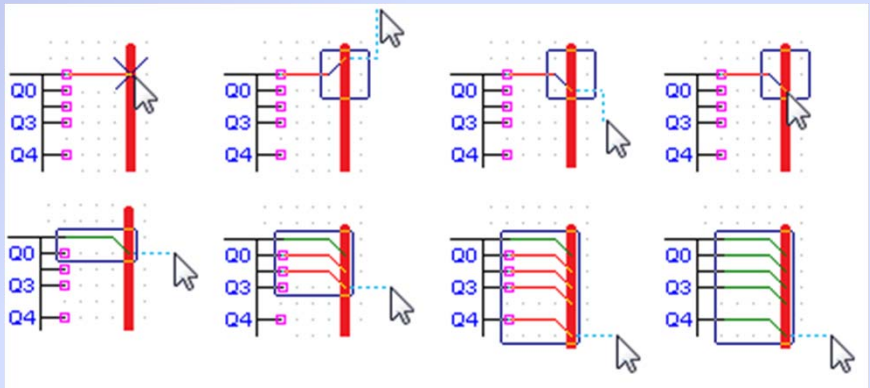
2	1,788 mS	Warning: [0132h] User Break-Point encountered (Data Processor [E_01100_DataProcessing_Sol.mc8])
3	2,388 mS	Warning: [0132h] User Break-Point encountered (Data Processor [E_01100_DataProcessing_Sol.mc8])
4	2,868 mS	Warning: [0194h] User Break-Point encountered (Data Processor [E_01100_DataProcessing_Sol.mc8])

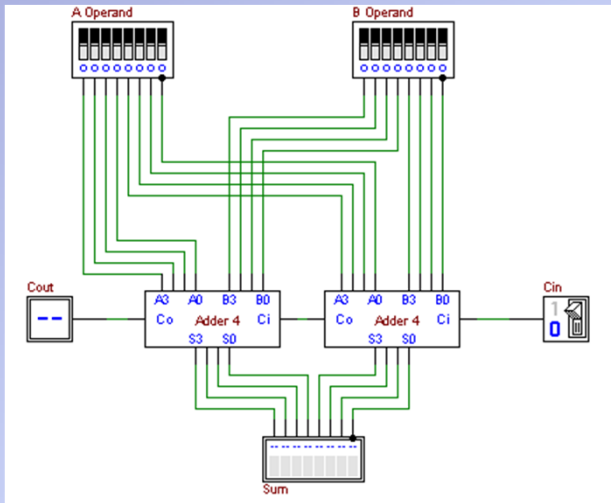
4,096 µS/pixel

Deeds: work in progress… (1)

- In the next major release:
- Support for BUSes
  - Components with “multiwire” pins
  - Classical wire-bus connections
- Data path components
  - Adders, comparators, barrel shifter
- RAM and ROM memories

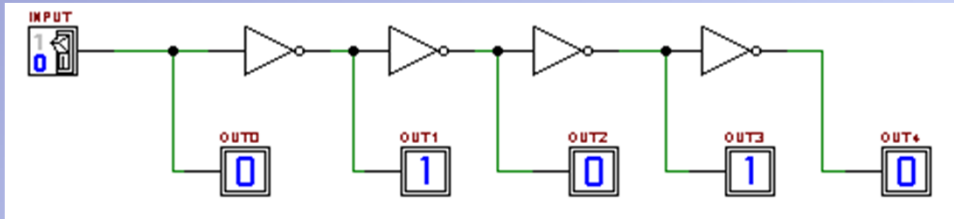


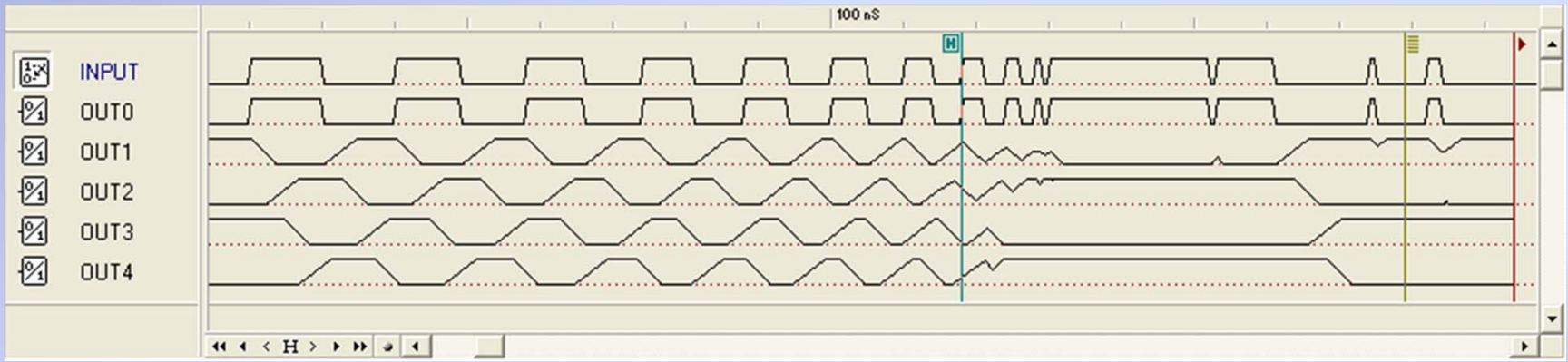




Deeds: work in progress… (2)

- In the next major release:
  - "Inertial" propagation time simulation





Thank you for your attention!

"The Deeds of Gallant Knights"
This image from a picture of G. David, XVI Century
Paris, Musèe de l'Armèe