UNIVERSITY OF LIVERPOOL

Digital System Design ELEC373/473

Introduction to Logic Design with Verilog

Lecture 1 - Introduction

Prof J.S. Smith
Room A515;
E-mail: j.s.smith@liv.ac.uk

Some Ground Rules

- Please arrive on time for lectures (Semester 2) and laboratories.
- Switch mobile phones to silent in lectures and laboratories.
- No eating/drinking in lectures or laboratories.
- No talking, unless invited to, in lectures (Semester 2).
- Always register your attendance for lectures and laboratories.
- No plagiarism or collusion in course work.
- No cheating in the open book class tests.

Who's on this module?

- The ELEC373 module is core for:
  - HZ03 Electronics Pathway
- The ELEC373 module is optional for:
  - Most Year 3 and Year 4 programmes
- The ELEC473 module is core for:
  - HH76 Year 4 and HHR6 Year 5
  - EEMS – Microelectronic Systems
- The ELEC473 module is optional for:
  - Some Year 4 programmes

Contacting me

- Please contact me by e.mail.
- If you would like a meeting, send some suggested times to j.s.smith@liv.ac.uk
- If I get lots of common questions I may post the answers on CANVAS and refer you to it.
  - The slight problem is I have to duplicate everything for the ELEC373 and ELEC473 modules.
- My office is A515 (EEE 5th Floor).

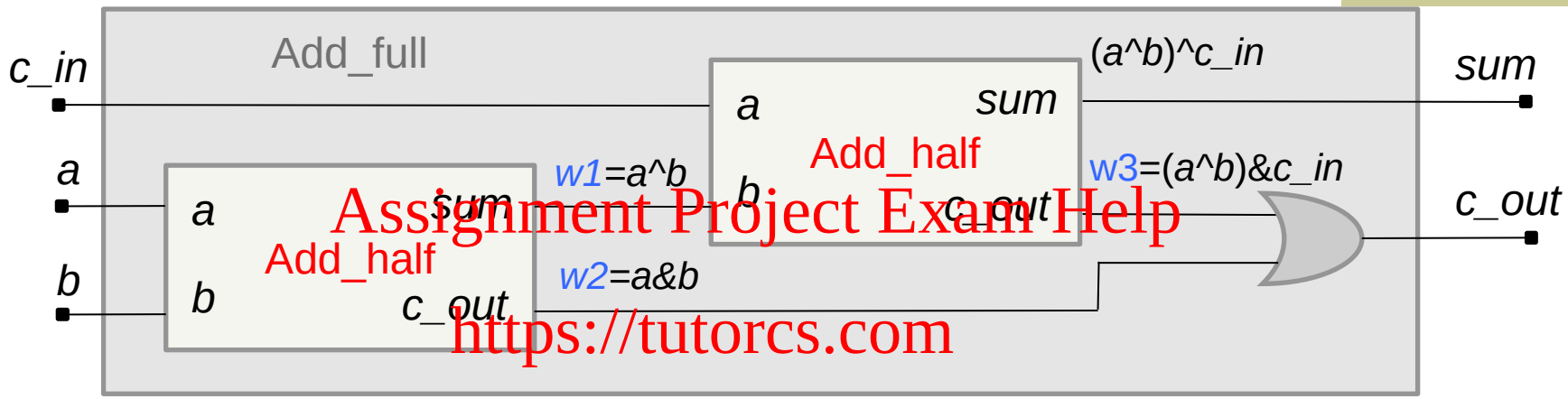
Module Aims

- Design and synthesise combinational and synchronous sequential digital systems with Verilog using:
  - Quartus package for synthesis and simulation (use V13.0 sp1)
  - Cyclone II FPGA on Altera DE2 development board
- Structural and Behavioural modelling and synthesis using Verilog
- Understand the problems of meta-stability
- Hardware test and design for testability
  - Use of logic analysers and (Altera's SignalTap logic analyser – when we get in the lab)
- Understand single-cycle, multicycle and pipelined processors architectures.
- Design and test a MIPS processor using ASMs and Verilog
  - This requires an understanding of assembly language programming
- Develop and test "System on a Programmable Chip" (SOPC) design using Altera NIOS II
  - This requires some C/C++ programming proficiency

Key Points

- Verilog is not a “programming” language
- Verilog is a “hardware” description language (HDL)
- Other hardware description languages include:
  - VHDL “Very High Speed Integrated Circuit Hardware Description Language”
  - AHDL “Altera Hardware Description Language”
- There are programs that simulate the behaviour of hardware descriptions and also programmes that synthesise digital circuits to implement their behaviour
  - We will use Quartus 13.0 sp1 for synthesis
  - And ModelSim for simulation

Verilog Example - Full Adder

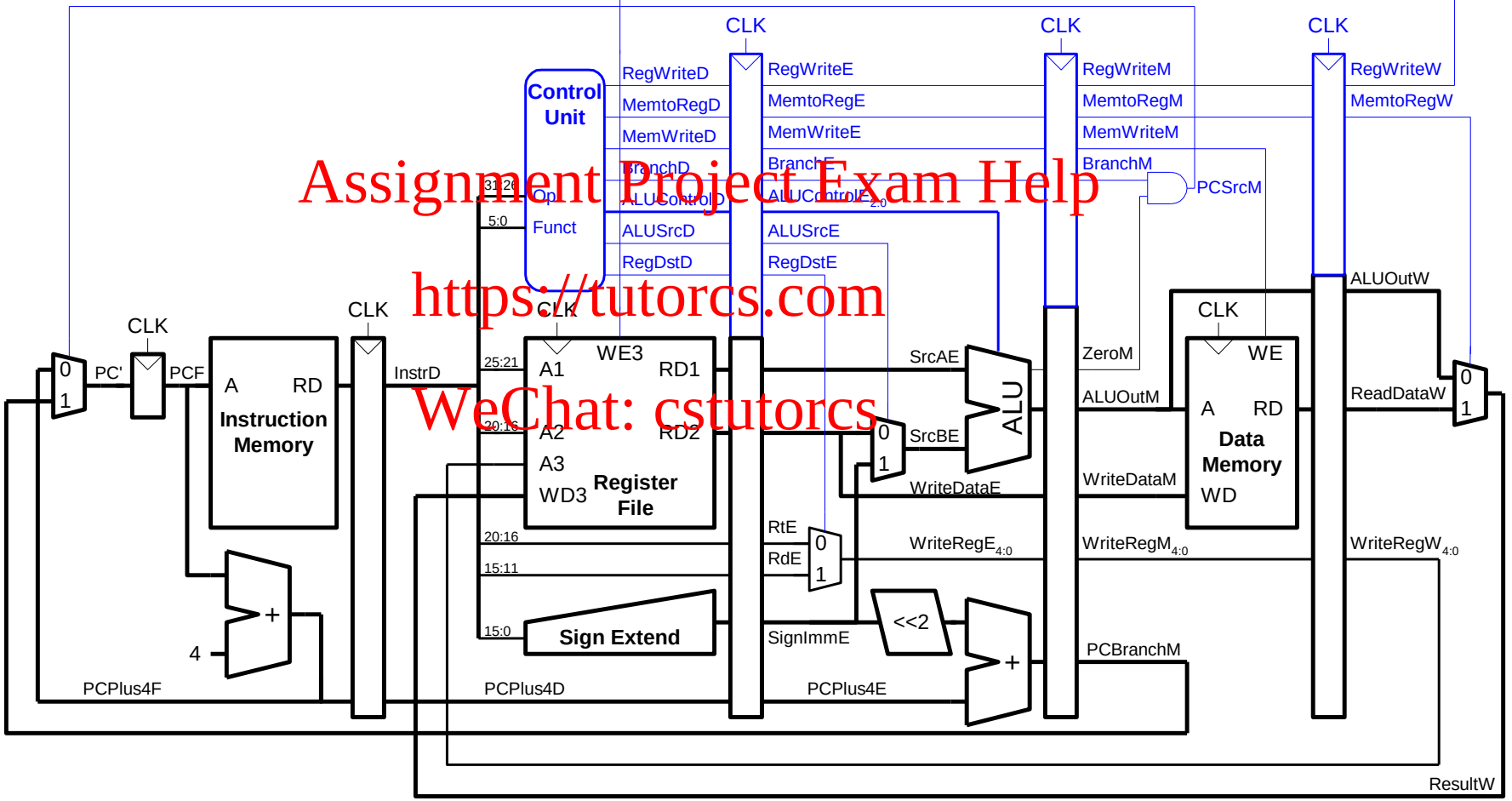



```
module Add_full (sum, c_out, a, b, c_in);
    input       a, b, c_in;
    output      sum, c_out;
    wire        w1, w2, w3;
    Add_half    M1 (w1, w2, a ,b);
    Add_half    M2 (sum, w3, c_in , w1);
    or          (c_out, w2, w3);
endmodule
```

Module instance name

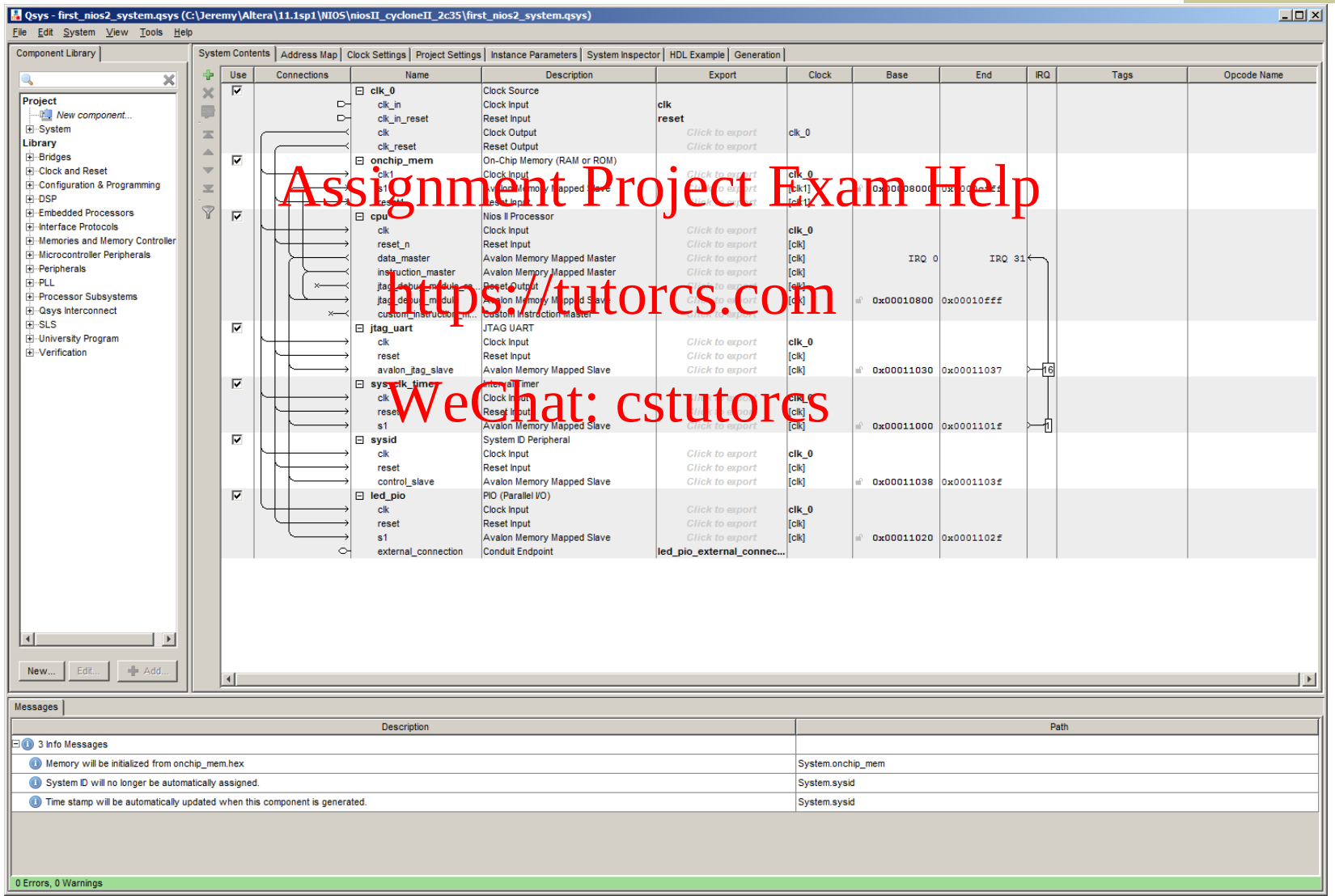
```
module Add_half ( sum, c_out, a ,b);
    input       a, b;
    output      c_out, sum;
    xor         (sum, a, b);
    and         (c_out, a, b);
endmodule
```

Pipelined MIPS Processor



Is this a “von Neumann” or “Harvard” architecture ?

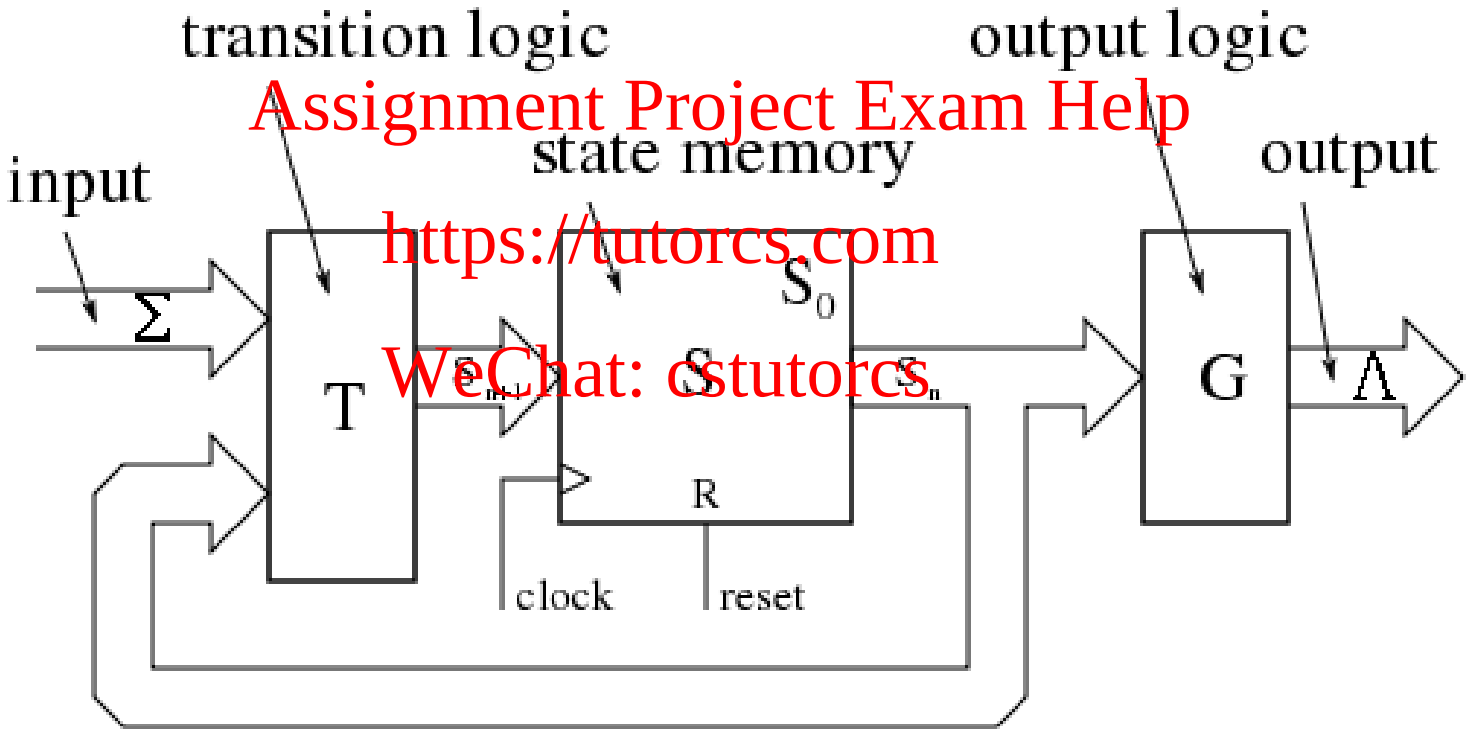
NIOS II - Qsys – SOPC Example



Expected background knowledge

- Number representations
- Understanding of basic Boolean logic.
- Understanding of basic combinational components
  - Gates, Decoders, Encoders, Multiplexers, Adders
- Karnaugh Map minimisation
- Understanding of basic sequential components
  - D Type, T Type, J-K flip flops and shift registers
- Sequential circuit design
  - Moore and Mealy state machines
- Algorithmic State Machine (ASM) Design
  - I'll give a brief recap on this in another Webinar.

Is this a Mealy or Moore State Machine (SM)?



Module Material

- Lecture notes on Canvas
- Recorded lectures on Canvas
- Some recorded Webinars on Canvas
- Texts
  - M. D. Cilleti, "Advanced Digital Design with the Verilog HDL", Prentice Hall
  - Samir Palnitkar, "Verilog HDL", 2003, ISBN 0-13-044911-3
  - David. A. Patterson and John L. Hennessy, "Computer Organization and Design: The Hardware/Software Interface" ,
- Standards
  - IEEE Std.1364-2001, IEEE Standard Verilog Hardware Description Language, IEEE, Inc., 2001.
  - IEEE Std 1364.1-2002, IEEE Standard for Verilog Register Transfer Level Synthesis, IEEE, Inc., 2002

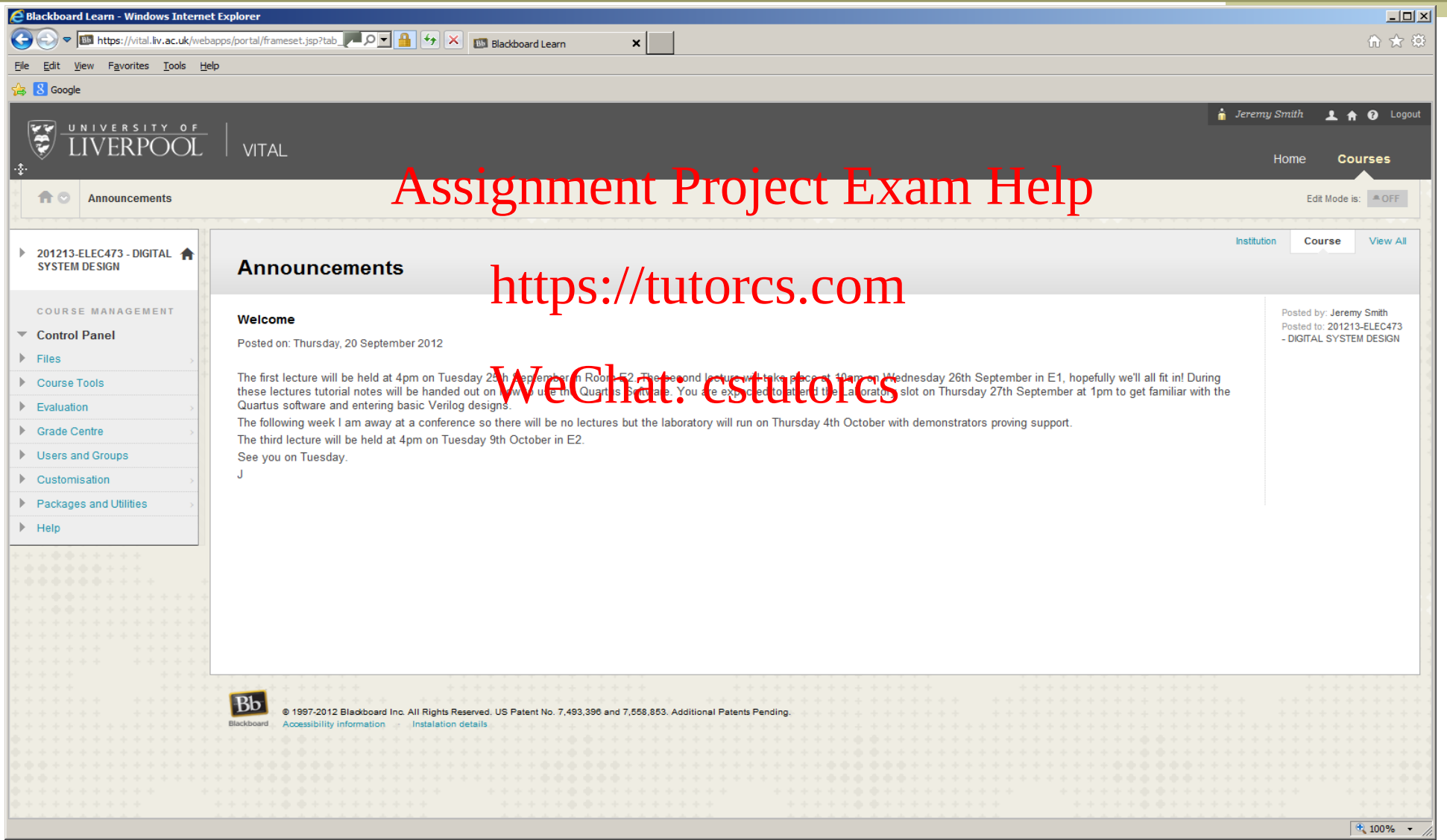
Assessment

- For both cohorts (ELEC373 & ELEC473) the module is 100% Continuous Assessment.
  - There will be 4 assignments - two in each Semester (80%).
  - Two **open book** class tests (one in each semester) (20%).
- Remember this is a 15 credit course which should total 150 Study hours
  - 24 hours lectures (on average one per week but front loaded i.e. more in the first semester than 2nd Semester)
  - Up to 12 hours tutorials (when needed)
  - 114 hours labs / private study / assignments
  - i.e. 4.75 hours of lab/private study per week

Likely Assignments (TBC)

- Note the assignments are front loaded, Assignments 3 and 4 require less time than Assignments 1 and 2:
  - Assignment 1 (15%) – Probably a UART design.
  - Assignment 2 (25%) – The full UART design.
  - Assignment 3 (20%) – Simulation and programming of a MIPS processor.
  - Assignment 4 (20%) – Designing using the NIOS-II processor.
- Note that some aspects of the Assignments will be different for the students on ELEC473 compare with ELEC373 as ELEC 473 is assessed as a Level 7 Module

Details will be available on CANVAS



Software

- This module will make heavy use of the Altera Quartus II package (V13.0 SP1)
  - Don’t install a later version as this is the last version that supports the Cyclone II FPGA on the DE2 board.
  - This software provides synthesis, timing analysis, and programming of the designs.
  - Designs can be entered by HDL languages (Verilog, VHDL, AHDL), state graph, and schematic entry.
  - The package provides a “wrapper” which uses ModelSim as a simulation engine
- I will not teach you how to use the Quartus II package
  - You must do this yourself using the tutorials and the help menu of the package.
- Note that Intel purchased Altera in 2015 for $16.7Billion so software should be downloaded from the Intel site
  - You may need to register before you can download. If you do please use you University of Liverpool email address.

Altera/Intel Quartus versions

Altera Quartus II Software v13.0 — Subscription Edition vs. Web Edition

Categories	Features	Web Edition Software	Subscription Edition Software
General Information	Getting started	Download (www.altera.com/download) and DVD (www.altera.com/dvdrequest)	
	Operating system support	Windows: XP (32/64 bit), 7 (32/64 bit), Windows Server 2008 R2 (64 bit) Linux: Red Hat Enterprise Linux 5 (32/64 bit), Red Hat Enterprise Linux 6(64 bit)	Windows: XP (32/64 bit), 7 (32/64 bit), Windows Server 2008 R2 (64 bit) Linux: Red Hat Enterprise Linux 5 (32/64 bit), Red Hat Enterprise Linux 6 (64 bit)
Device Support	CPLD	MAX® series devices: All	MAX series devices: All
	Low-cost FPGA	Cyclone® V FPGAs: All (Excluding 5CEA9, 5CGXC9, and 5CGTD9) Cyclone IV FPGAs: All Cyclone III FPGAs: All Cyclone II FPGAs: All Cyclone FPGAs: None	Cyclone V FPGAs: All Cyclone IV FPGAs: All Cyclone III FPGAs: All Cyclone II FPGAs: All Cyclone FPGAs: All
	Mid-range FPGA	Arria® V FPGAs: None Arria II FPGAs: EP2AGX45 Arria FPGAs: None	Arria V FPGAs: All Arria II FPGAs: All Arria FPGAs: All
	High-end FPGA	Stratix® series devices: None	Stratix series devices: All
	SoCs	Cyclone V SoCs: All	Cyclone V SoCs: All
	ASIC	HardCopy® series: None	HardCopy series: All
Intellectual Property (IP)	Altera and partner IP	Yes, including free OpenCore Plus evaluation feature	
	Full-license IP base suite	IP available for purchase	DSP: FIR, FIR II, FFT, and NCO Compilers Devices: Generic II Memory controllers: DDR, DDR2, and DDR3 for ALTMEMPHY, DDR2, DDR3, and LPDDR2 for UniPHY, RLDRAM II, QDR II, RLDRAM II for UniPHY and QDR II / II+ for UniPHY
Design Entry	Qsys	Yes	
	Schematic entry and language support	Schematic entry, Verilog, VHDL, and System Verilog	
Design Environment	Tcl scripting and command-line support	Yes	
Implementation and Optimization	Incremental compilation and team-based design	No	Yes
	LogicLock™ incremental design capability	No	Yes
	Multiprocessor support	Available with TalkBack enabled	Yes
	Rapid Recompile	No	Yes
	Physical synthesis optimizations	Yes	
	Chip Planner	Yes	
	Live I/O checking	Yes	
	TimeQuest timing analyzer and optimization advisor	Yes	
	Synopsys Design Constraint (SDC) format support	Yes	
	Early power estimator	Available for download on www.altera.com at no cost	
	PowerPlay power analysis and optimization	Yes	
Verification and Debug	SignalTap™ II logic analyzer	Available with TalkBack enabled	Yes
	SignalProbe feature	Available with TalkBack enabled	Yes
	Transceiver Toolkit	No	Yes
	ModelSim®-Altera® Starter Edition	Included	
	ModelSim-Altera Edition	This option is sold for $945	
	Embedded logic analyzer interface	Yes	
	RTL viewer and technology map viewer	Yes	
	Pin planner	Yes	

The PCs in the A301Lab and A304 have the full Subscription edition as do some computers in the Harold Cohen.

For use on your own “MS Windows” computer you can either install the Web edition or install the subscription edition and authenticate against the University’s licence servers using the University’s VPN software (you need to apply for VPN access).

For work submitted it must be configured to work with the full subscription edition.

For assignment 4 you will need the subscription edition to use the “IP”

Quartus Licence File

- Requires a licence file with these lines:

  SERVER lxc.liv.ac.uk 1730

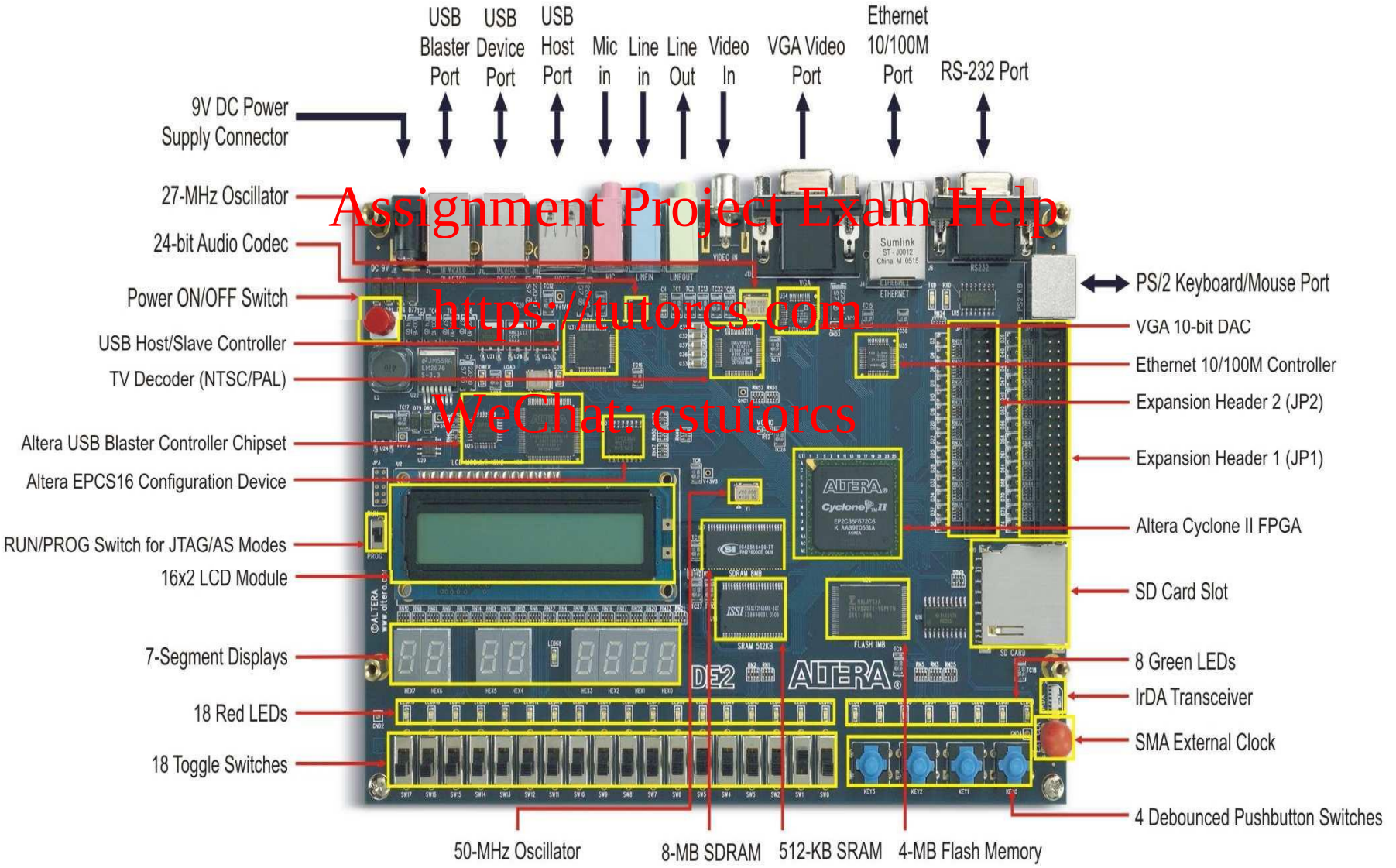
  USE_SERVER

- Or you could just directly enter:
  - 1730@lxc.liv.ac.uk
- You must be on a network that can access lxc.liv.ac.uk, i.e. either on the University network or connected via the VPN service
  - You need to apply to CSD to use the VPN service
  - I'm still checking to see if I can bulk register you for the VPN Service

Laboratories

- The Laboratories are scheduled for Friday 2-5pm in EEE A301+A304.
- The first Laboratory on Friday is a standard using Quartus Tutorial.
- This allows you to get familiar with the hardware (The DE2 Board).

Laboratory Equipment :
DE2 (Development and Education) FPGA board



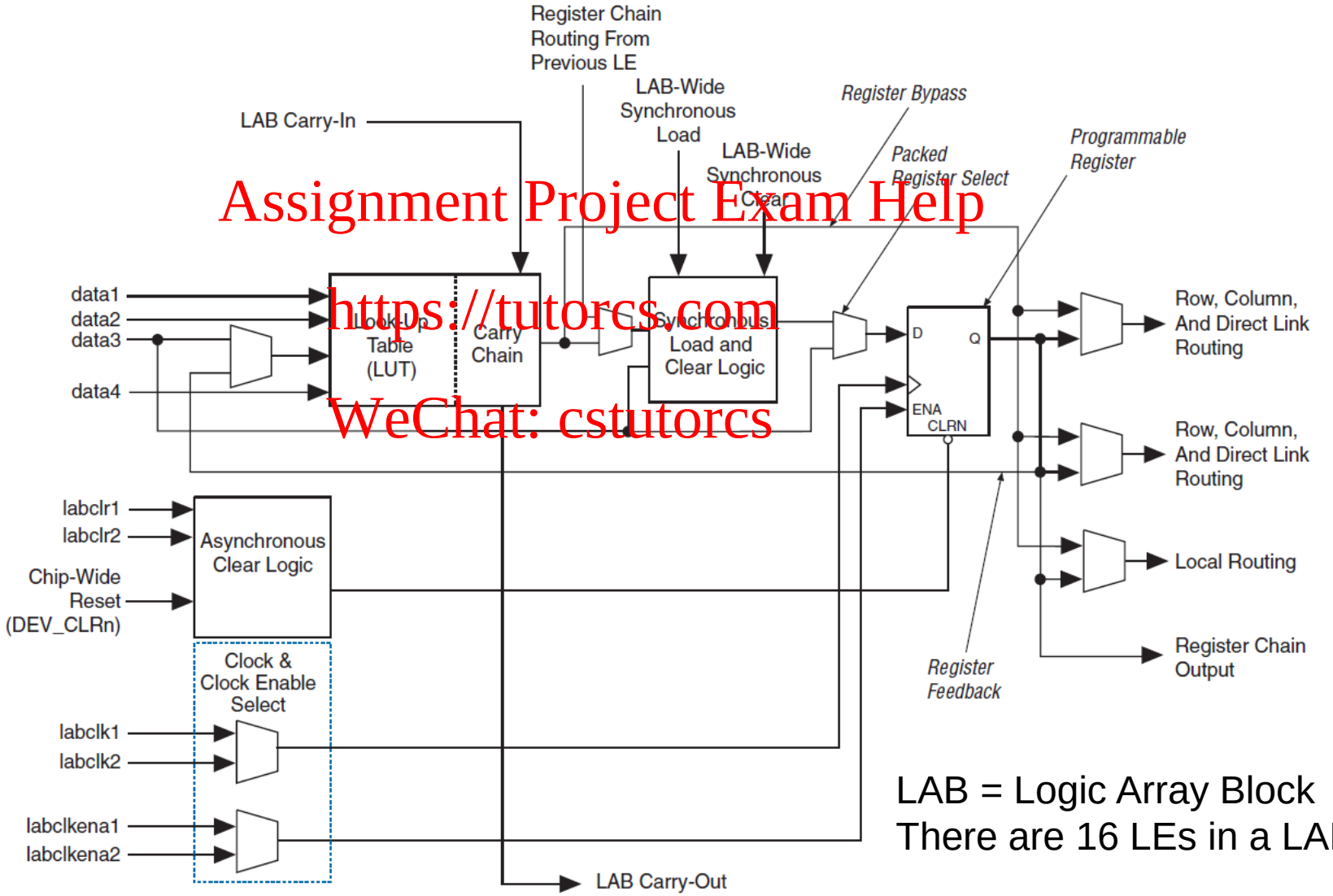
Altera's EP2C35F672C6N

Table 1–1. Cyclone II FPGA Family Features (Part 1 of 2)

Feature	EP2C5 *(2)*	EP2C8 *(2)*	EP2C15 *(1)*	EP2C20 *(2)*	EP2C35	EP2C50	EP2C70
LEs	4,608	8,256	14,448	18,752	33,216	50,528	68,416
M4K RAM blocks (4 Kbits plus 512 parity bits	26	36	52	52	105	129	250
Total RAM bits	119,808	165,888	239,616	239,616	483,840	594,432	1,152,000
Embedded multipliers *(3)*	13	18	26	26	35	86	150
PLLs	2	2	4	4	4	4	4

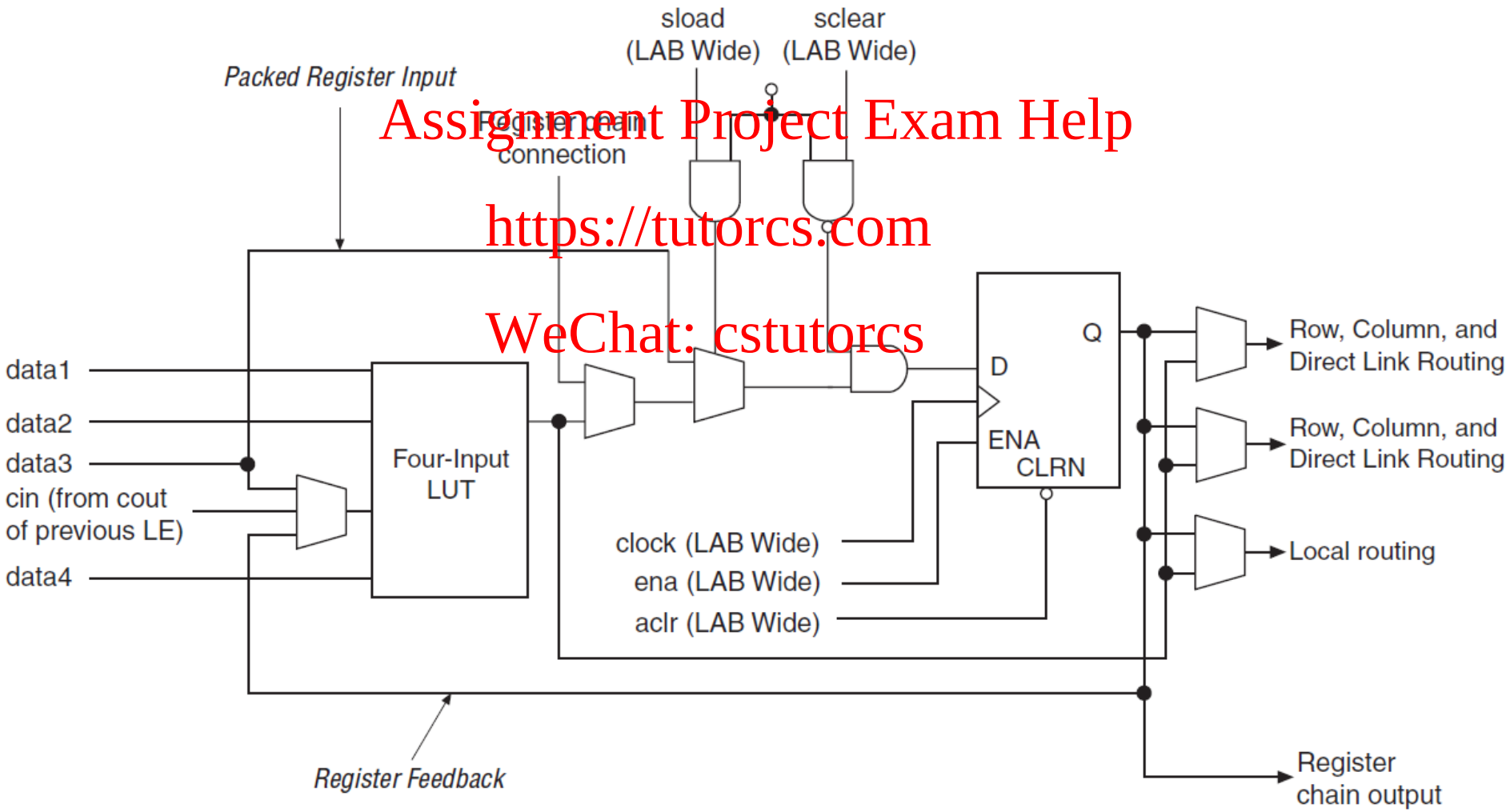
LE = Logic Element
PLL = Phased Locked Loop

Cyclone II LE (Logic Element)



LAB = Logic Array Block
There are 16 LEs in a LAB

LE (Logic Element) in Normal Mode



The Design Process

- Broadly speaking design can be categorised into three stages:
  - Conceptual Design
  - Embodiment Design
  - Detailed Design
- Design is iterative, designs evolve during the design process. Sometimes when problems crop up in the later design stages you have to go back and refine the Conceptual Design.
- One of the most important tasks in the design process is communication of the design because, in the “real world”, you’ll probably be working as a member of a design team rather than as an individually.
  - You need to be able to accurately communicate your design to colleagues and other engineers who may be working on the project. This communication will happen at all three stages of the design process.
  - In this module you need to be able to communicate your design to the assignment assessor.

Conceptual Design

- Think of this as the top level design.
- For this module we'll use block diagrams to show different blocks and label signals between the different blocks.
- Each block should be capable of being designed and tested independently.
- Use appropriate names for the blocks and signals which indicate their function.
- Try to be consistent with your signal and block names throughout the design process.
  - i.e. use the same names in the blocks, ASMs and Verilog.
  - Also make clear the "polarity" of the signal is it "active high" or "active low".
    - In the past some designs haven't worked because, for some modules, students have an active high reset and other modules have an active low reset.
    - Note that on the DE2 Board the four switches are "active low" i.e. when pressed the generate a low signal otherwise a high signal.

Embodiment Design

- This is where the conceptual design is expanded.
- The boundary between conceptual design and embodiment design can be a bit subjective.
- In this module you will produce ASMs or ASMDs during the embodiment stage. At least one ASM should be produced for each block from the Conceptual stage.
- However, during the Embodiment Stage, you may break down one block from the conceptual design to multiple blocks in the embodiment design.
- What will get passed to the “Detailed Design” is ASM or ASMD charts.

Detailed Design

- This is the final stage of the design process.
- In this module the inputs to the detailed design will be the ASM or ASMD charts.
- The outputs will be commented Verilog code, that can synthesise the full system.

Where is the design done?

- In the "top down" design process the bulk of the design work is undertaken during the conceptual and embodiment design stages.
  - i.e. on paper with an pen and pencil and not by writing Verilog code
- Once the ASMs have been designed it is relatively mechanistic to convert state machines to Verilog code.
- In fact Altera Quartus has a tool to convert State Machines to Verilog.
  - However you'll probably find it quicker to directly code in Verilog AFTER you have completed your ASMs rather than using the State Machine Wizard.
  - Also the State Machine Wizard code isn't very compact

Next Lecture

- Recap (or introduction) on Algorithmic State Machines (ASMs)