Logic design with FPGAs

Synchronous logic and logic synthesis

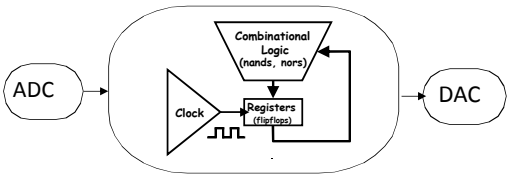
BASYS2 FPGA board

Spartan FPGA

Xilinx ISE WebPACK

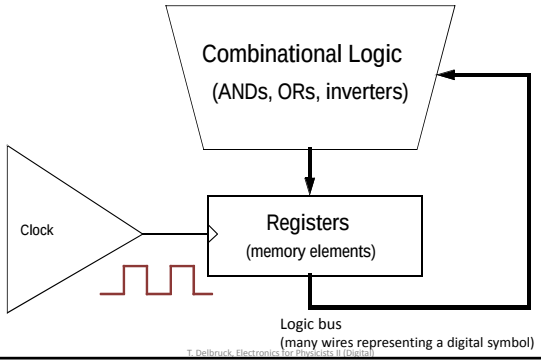
Verilog & Gateway "HelloWorld" exercise

Digital computation



1. Fast global clock
2. Bit-perfect deterministic logical state

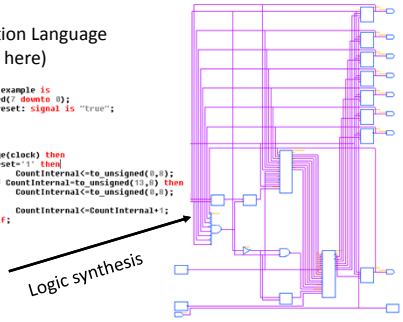
Synchronous logic



How logic is designed now

Hardware Description Language
(VHDL syntax used here)

```
architecture example_arch of example is
signal CountInternal: unsigned(7 downto 0);
attribute sync_set_reset of reset: signal is "true";
begin

        process(clock)
        begin
                if rising_edge(clock) then
                        if reset='1' then
                                CountInternal<=to_unsigned(0,8);
                        elsif CountInternal=to_unsigned(13,8) then
                                CountInternal<=to_unsigned(0,8);
                        else
                                CountInternal<=CountInternal+1;
                        end if;
                end if;
        end process;
        count<=CountInternal;
end example_arch;
```



By using HDLs, industry can design complex chips with >100 million logic elements

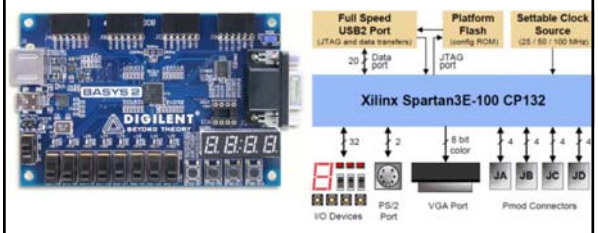
Field Programmable Gate Arrays (FPGAs)

- Reconfigurable logic chips with lots of registers, compared with **CPLD**s and **PAL**s.
- Range in price from $10 to $10,000 per chip depending on number of "gate equivalents".
- They **do not** provide you a processor with instructions, ALU and memory (although you can embed a processor in a larger FPGA).
- Sold by Xilinx, Altera, Lattice, Actel, etc

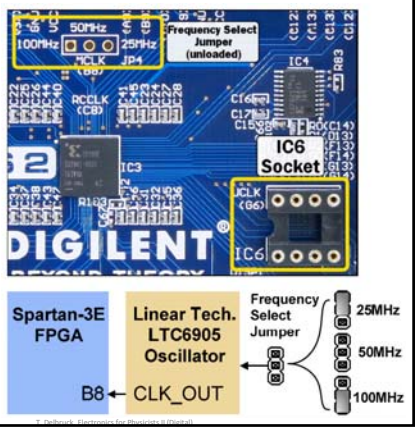
CPLD = Complex Programmable Logic Device
PAL = Programmable Array Logic

BASYS2 FPGA board

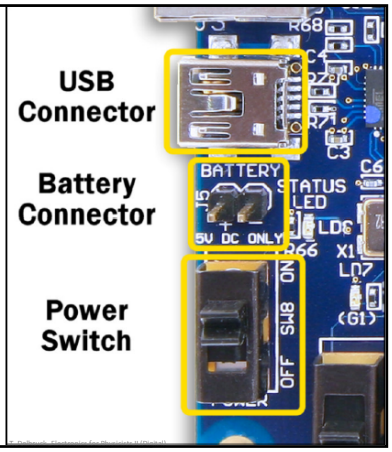
Xilinx Spartan 3E XCS100E



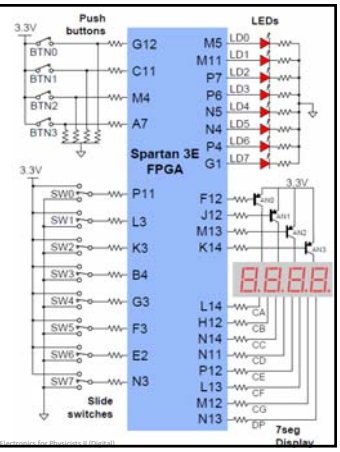
BASYS2 oscillator



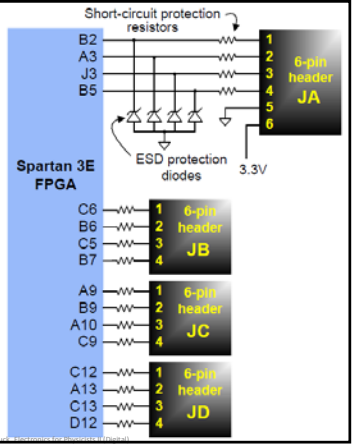
BASYS2 power



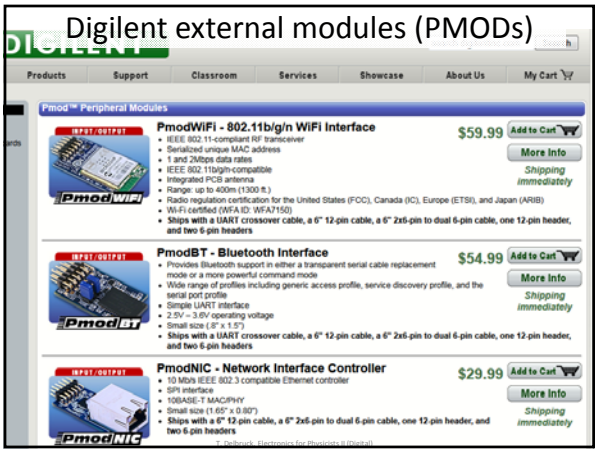
BASYS2 IO



BASYS2 PMOD connectors



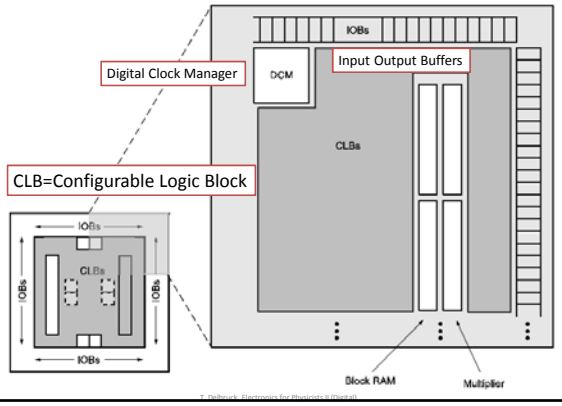
Digilent external modules (PMODs)



BASYS2 capabilities

- Xilinx Spartan 3-E FPGA, 100K or 250K gate
- FPGA features 18-bit multipliers, 72Kbits of fast dual-port block RAM, and 500MHz+ operation
- USB 2 full-speed port for FPGA configuration and data transfers (using Adept 2.0 software available as a free download)
- XCF02 Platform Flash ROM that stores FPGA configurations indefinitely
- User-settable oscillator frequency (25, 50, and 100 MHz), plus socket for a second oscillator
- Three on-board voltage regulators (1.2V, 2.5V, and 3.3V) that allow use of 3.5V-5.5V external supplies
- 8 LEDs, 4-digit seven-segment display, four pushbuttons, 8 slide switches, PS/2 port, and a 8-bit VGA port
- Four 6-pin headers for user I/Os, and attaching Digilent PMOD accessory circuit boards

Spartan FPGA architecture



Spartan 3 XC3S100E

Table 2: **Available User I/Os and Differential (Diff) I/O Pairs**

Package	VQ100 VQG100		CP132 CPG132		TQ144 TQG144		PQ208 PQG208	
Size (mm)	16 x 16		8 x 8		22 x 22		28 x 28	
Device	User	Diff	User	Diff	User	Diff	User	Dif
XC3S100E	66 (7)	30 (2)	83 (11)	35 (2)	108 (28)	40 (4)	-	-

Device datasheet (233 pages…)

XILINX®

DS312 (v3.8) August 26, 2009

Spartan-3E FPGA Family: Data Sheet

Product Specification

Module 1: Spartan-3E FPGA Family: Introduction and Ordering Information

DS312-1 (v3.8) August 26, 2009

- Introduction
- Features
- Architectural Overview
- Package Marking
- Ordering Information

Module 2: Functional Description

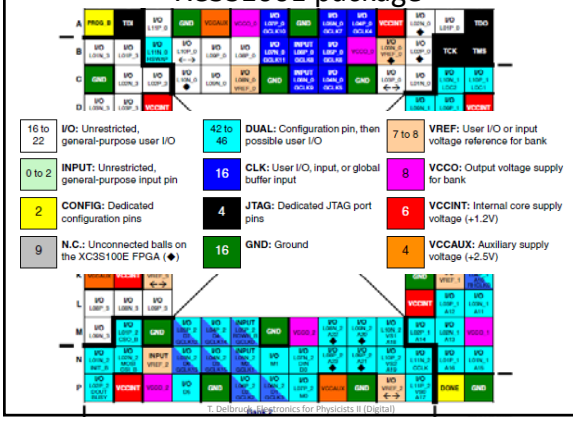
DS312-2 (v3.8) August 26, 2009

Module 3: DC and Switching Characteristics

DS312-3 (v3.8) August 26, 2009

- DC Electrical Characteristics
  - Absolute Maximum Ratings
  - Supply Voltage Specifications
  - Recommended Operating Conditions
  - DC Characteristics
- Switching Characteristics
  - I/O Timing
  - SLICE Timing
  - DCM Timing
  - Block RAM Timing
  - Multiplier Timing

XC3S100E package



16 to 22	**I/O:** Unrestricted, general-purpose user I/O	42 to 46	**DUAL:** Configuration pin, then possible user I/O
0 to 2	**INPUT:** Unrestricted, general-purpose input pin	16	**CLK:** User I/O, input, or global buffer input
2	**CONFIG:** Dedicated configuration pins	4	**JTAG:** Dedicated JTAG port pins
9	**N.C.:** Unconnected balls on the XC3S100E FPGA (◆)	16	**GND:** Ground

7 to 8	**VREF:** User I/O or input voltage reference for bank
8	**VCCO:** Output voltage supply for bank
6	**VCCINT:** Internal core supply voltage (+1.2V)
4	**VCCAUX:** Auxiliary supply voltage (+2.5V)

CLBs (Config. Logic Blocks)

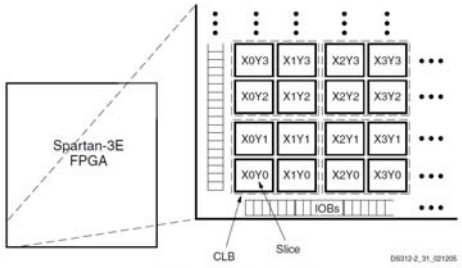


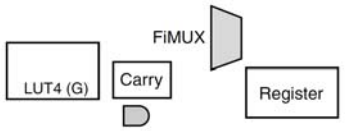
Figure 14: **CLB Locations**

Table 9: **Spartan-3E CLB Resources**

Device	CLB Rows	CLB Columns	CLB Total(1)	Slices	LUTs / Flip-Flops	Equivalent Logic Cells	RAM16 / SRL16	Distributed RAM Bits
XC3S100E	22	16	240	960	1,920	2,160	960	15,360

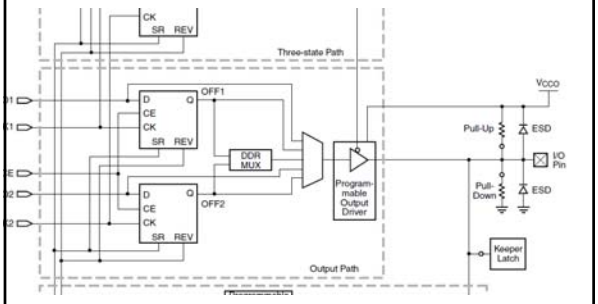
CLB structure

- Each CLB has 4 **slices**
- Each slice has 4-input LUT (look-up table) logic block and single bit latch (register), plus multiplexors, shift registers, carry bits, RAM, etc.
- One slice simplified: The LUT is programmed to compute any desired logic function of 4 inputs.

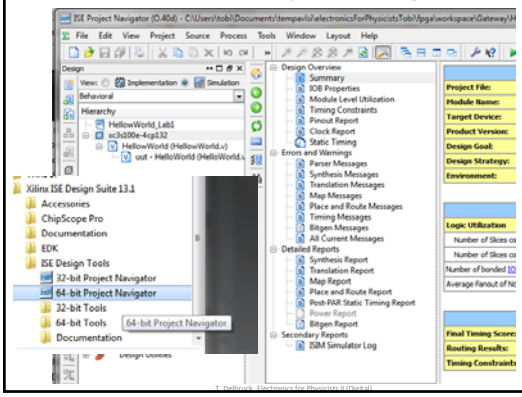


- Generally you don't need to know about this; just to be aware of what fabric your synthesized logic is finally mapped to.

Spartan IOB structure (part) (Input Output Buffer)



Xilinx ISE Project Navigator



HelloWorld.v

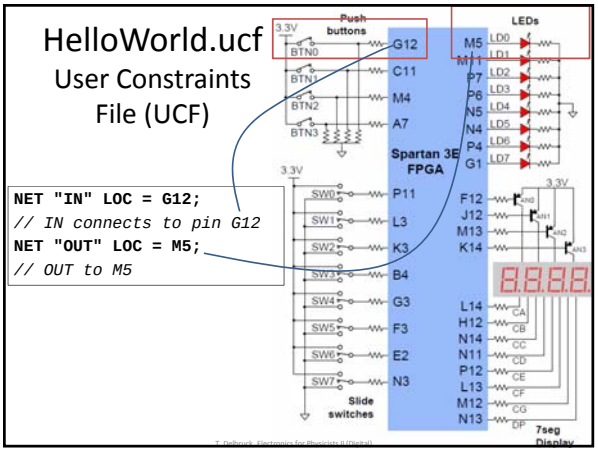
verilog hardware description language (HDL)

```
module HelloWorld(
    input IN,
    output OUT
    );

    assign OUT = IN;

endmodule
```

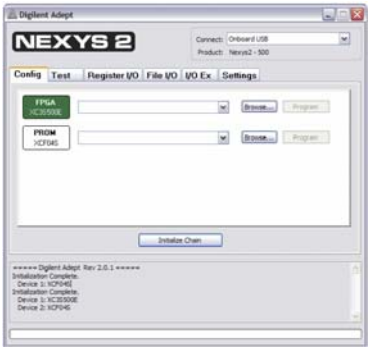
HelloWorld.ucf

User Constraints File (UCF)

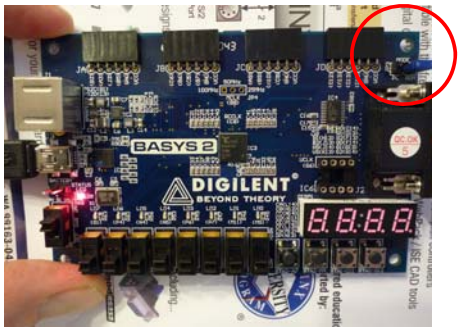
```
NET "IN" LOC = G12;
// IN connects to pin G12
NET "OUT" LOC = M5;
// OUT to M5
```



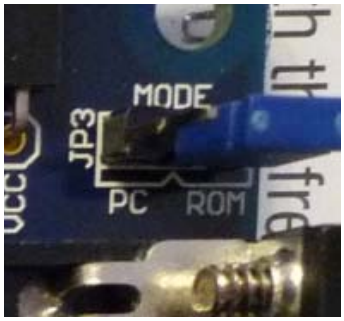
Programming BASYS2 with Adept



Programming with Adept



Programming with Adept

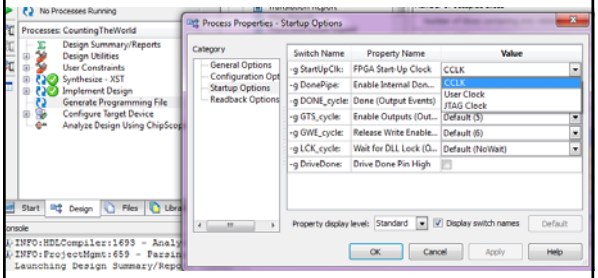


Two programming possibilities

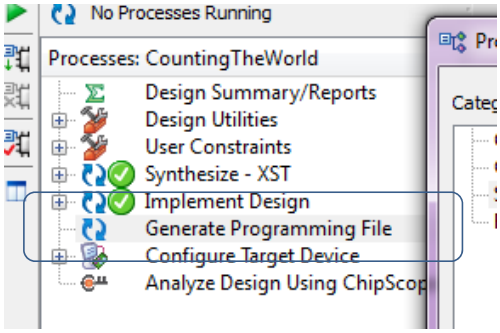
1. **PC** – configuration is loaded to FPGA RAM
   - Very fast
2. **ROM** – configuration is written to flash memory
   - Nonvolatile but slow

You must use different clock setup in project settings

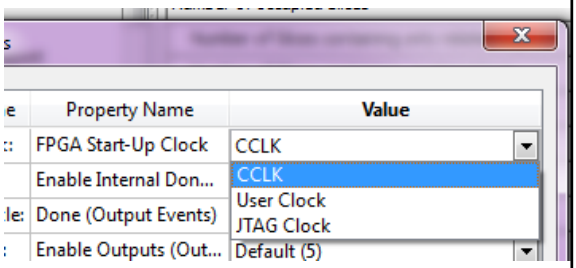
Choosing the clock source for programming with Adept



Choosing the clock source for programming with Adept



Use *CCLK* for ROM, *JTAG Clock* for RAM



"Gateway" lab exercises

1. HelloWorld - Making a new project with a *module* and controlling a single LED with a button.
2. Hello lots of Worlds - making a *bus* to wire all switches to all LEDs; the UCF (User Constraints File).
3. HelloWorldSynchronous - using *registers* and *wires*, simulation with a verilog *test fixture*; the *sensitivity list* in always@ in simulation. "If within a module you have a signal that is on the left hand side of an assignment within an 'always@(…)' statement, then it needs to be defined as a register ('reg')".
4. ShiftingTheWorld - synthesizing a shift register with *fd* D-FlipFlops using gate level and behavioral level design; *register transfer level (RTL)* design; *module instantiation*; signal*concatenation*; introduction to *generate*.
5. ShiftingManyWorlds - 2d array of shift registers (memory); simulation exercise.
6. CountingWorlds - simple arithmetic, *multiplexing*.

"Gateway" lab exercises

1. HelloWorld - Making a new project with a ***module*** and controlling a single LED with a button.
2. Hello lots of Worlds - making a ***bus*** to wire all switches to all LEDs; the UCF (User Constraints File).
3. HelloWorldSynchronous - using ***registers*** and ***wires***, simulation with a verilog ***test fixture***; the ***sensitivity list*** in ***always@*** in simulation. "If within a module you have a signal that is on the left hand side of an assignment within an 'always@(…)' statement, then it needs to be defined as a register ('reg')".
4. ShiftingTheWorld - synthesizing a shift register with *fd* D-FlipFlops using gate level and behavioral level design; ***register transfer level (RTL)*** design; *module instantiation*; signal ***concatenation***; introduction to ***generate***.
5. ShiftingManyWorlds - 2d array of shift registers (memory); simulation exercise.

"Gateway" lab exercises

6. CountingWorlds - simple arithmetic, ***multiplexing***.
7. TimingTheWorld - a second-counter watch using two counters, one clocking the other, both up/down with enable.
8. DecodingTheWorld - Number representation; 7-segment display *decoder* (see BASYS2 manual). See 7seg for the code for this exercise.
9. TimingTheWorldInDecimel - multiple counters, using ***generics*** to instantiate modules with parameters; revisit *generate*.

"Gateway" lab exercises

10. ColourTheWorld - *parameters* in generics; VGA display control to generate sync signals and RGB colors (see BASYS2 manual).
11. WorldOfStateMachines - making ***state machines*** using sequential and combinational blocks (switch/case statements) and using ROM modules (**$readmemb**).
12. WorldOfLinkedStateMachines - multiple state machines linked by a master state machine.
13. Snake - a complete snake game.