An Introduction to Lab1

Lecture 2 for Information Processing

Aaron Zhao, Imperial College London, a.zhao@imperial.ac.uk

What is in this lab?

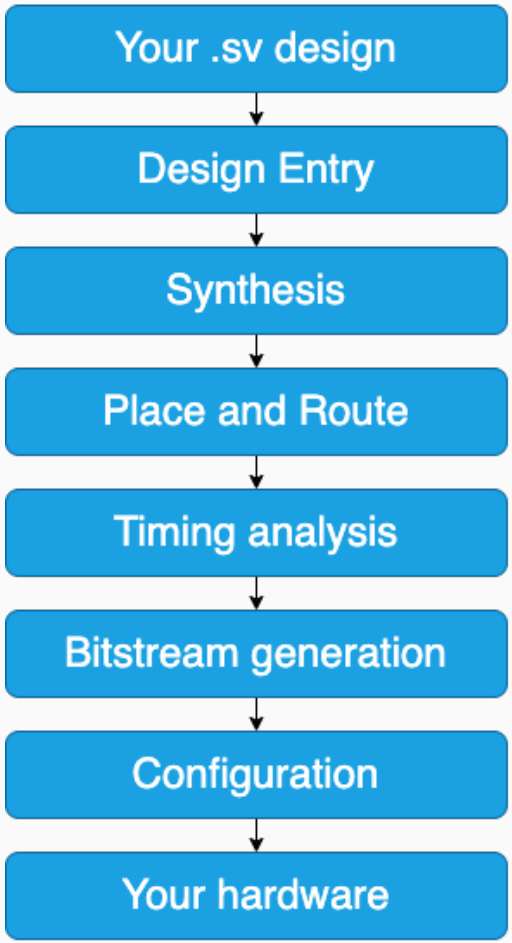
- Setting up Quartus Prime Lite to run.
- Create a directory structure for this and subsequent labs.
- Create a new project in Quartus and complete a basic 7-segment LED display decoder design using Quartus and Verilog.
- Program the MAX10 FPGA chip on the DE10-Lite board with your design.
- Understand the FPGA compilation process.
- Create another project for hex-to-BCD decoding.
- Explore and test your design.

Setting up your environment

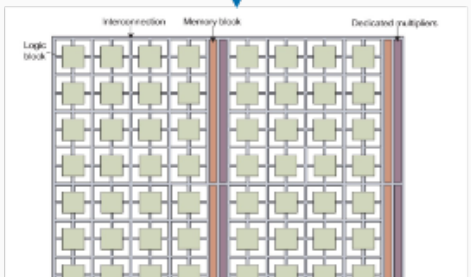
- Quartus Prime (preferred setup is in the VirtualBox or lab machine).
- Explore the programmer: this uses the USB-blaster to program the FPGA from your host machine.
- Explore the Ping Assignment tool.
- Netlist Viewer and Timing Analyzer.
- Be careful with your directory structure!

FPGA Compilation

You will have to go through a number of steps to map your design to the actual FPGA.

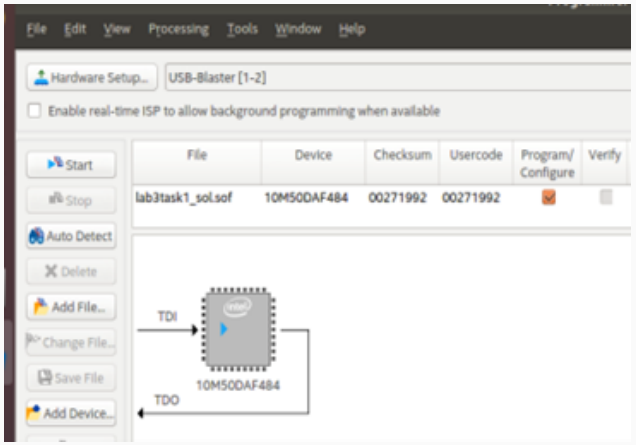



```
//-------------------------------
// Module name: task1_top
// Function: Top level module for Lab 3 Task 1
//           to display 4 switch on a 7-seg display
// Creator:  Peter Cheung
// Version:  1.0
// Date:     31 Oct 2020
//-------------------------------
module  task1_top (
     SW,               // input switches
     HEX0              // Hex output on 7 segment display
   );
  input    [3:0] SW;       // declare input/output ports
  output   [6:0] HEX0;

  hex_to_7seg    SEG0 (HEX0, SW[3:0]);

endmodule
```

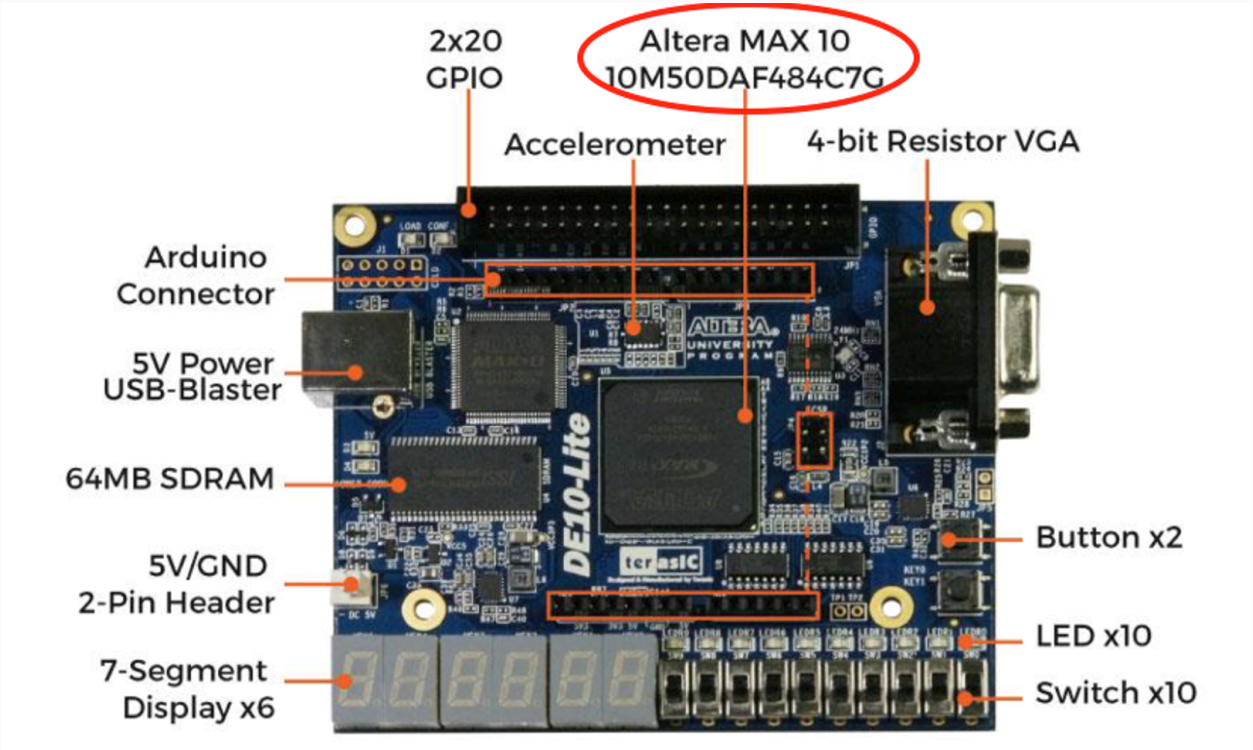


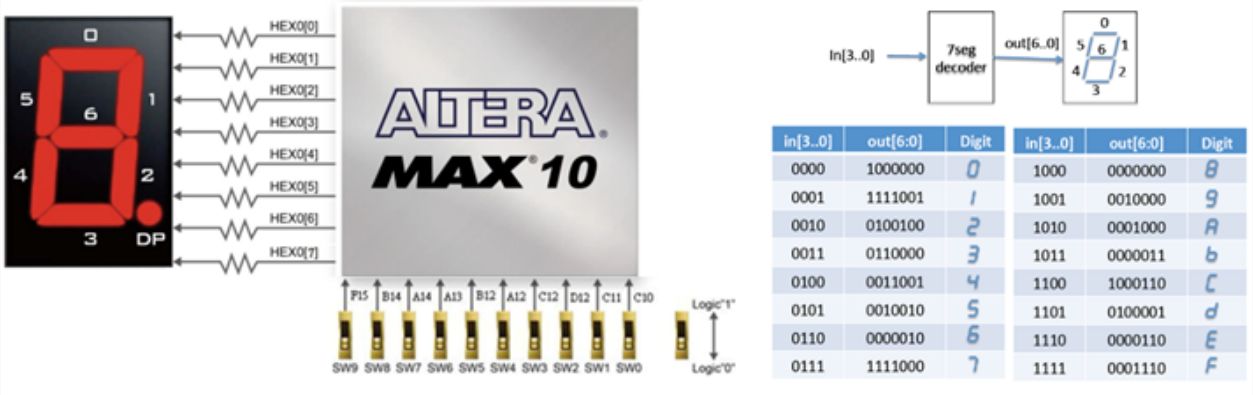
Blasting the FPGA

We have to transfer the bitstream generated on your local PC now to the FPGA device, so we need to use the Programmer in the Quartus toolchain to do this.



7-segment LED display





in[3..0]	out[6:0]	Digit	in[3..0]	out[6:0]	Digit
0000	1000000	0	1000	0000000	8
0001	1111001	1	1001	0010000	9
0010	0100100	2	1010	0001000	A
0011	0110000	3	1011	0000011	b
0100	0011001	4	1100	1000110	C
0101	0010010	5	1101	0100001	d
0110	0000010	6	1110	0000110	E
0111	1111000	7	1111	0001110	F

7-segment LED display - Putting this to Verilog

```
module hex_to_7seg    (out,in);

    output    [6:0] out;     // low-active outp
    input [3:0] in;        // 4-bit binary inp

    reg       [6:0] out;   // make out a varia

    always @ (*)
      case (in)
        4'h0: out = 7'b1000000;
        4'h1: out = 7'b1111001;      //  --0 --
        4'h2: out = 7'b0100100;      //  |     |
        4'h3: out = 7'b0110000;      //  5     1
        4'h4: out = 7'b0011001;      //  |     |
        4'h5: out = 7'b0010010;      //  --6 --
        4'h6: out = 7'b0000010;      //  |     |
        4'h7: out = 7'b1111000;      //  4     2
        4'h8: out = 7'b0000000;      //  |     |
        4'h9: out = 7'b0011000;      //  --3 --
        4'ha: out = 7'b0001000;
        4'hb: out = 7'b0000011;
        4'hc: out = 7'b1000110;
        4'hd: out = 7'b0100001;
        4'he: out = 7'b0000110;
        4'hf: out = 7'b0001110;
      endcase
endmodule
```

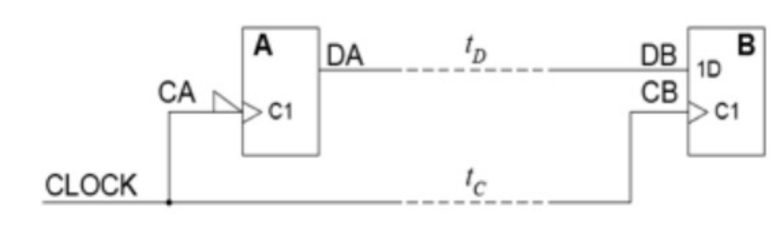
7-segment LED display - Ping assignments

Pin assignment is required, it defines how your block connects to outside world

Signal Name	Pin Location
HEX0[6]	PIN_C17
HEX0[5]	PIN_D17
HEX0[4]	PIN_E16
HEX0[3]	PIN_C16
HEX0[2]	PIN_C15
HEX0[1]	PIN_E15
HEX0[0]	PIN_C14
SW[3]	PIN_C12
SW[2]	PIN_D12
SW[1]	PIN_C11
SW[0]	PIN_C10

Timing Analysis

- Signal Propagation
- Sequential design: How fast can you clock it?
- Timing models in place
- Tools check all possible paths



Questions?

Questions?