Lab 4 - Implementing a Cortex-M0 based SoC on the DE10-Lite Intel FPGA board

The latest and greatest Cortex-M0 (and Cortex-M3) Design Start package (rel2 at this time) can be downloaded after registering with ARM using the following link

https://www.arm.com/resources/designstart

For the reasons I detailed in lecture 7 (must watch before starting this lab) we are going to use the older rel0 package. Please download the Lab4 archive from BB and unzip it into a suitable location. Have the file ***Lab4_RTL.pdf*** to hand to see all the connections.

Assignment 1: Designing a peripheral device for the AMBA3 AHB-Lite bus (amend AHB2LED.v file)

You are going to amend the given peripheral that allows the SoC to operate LEDs and SWitches according to the diagram below

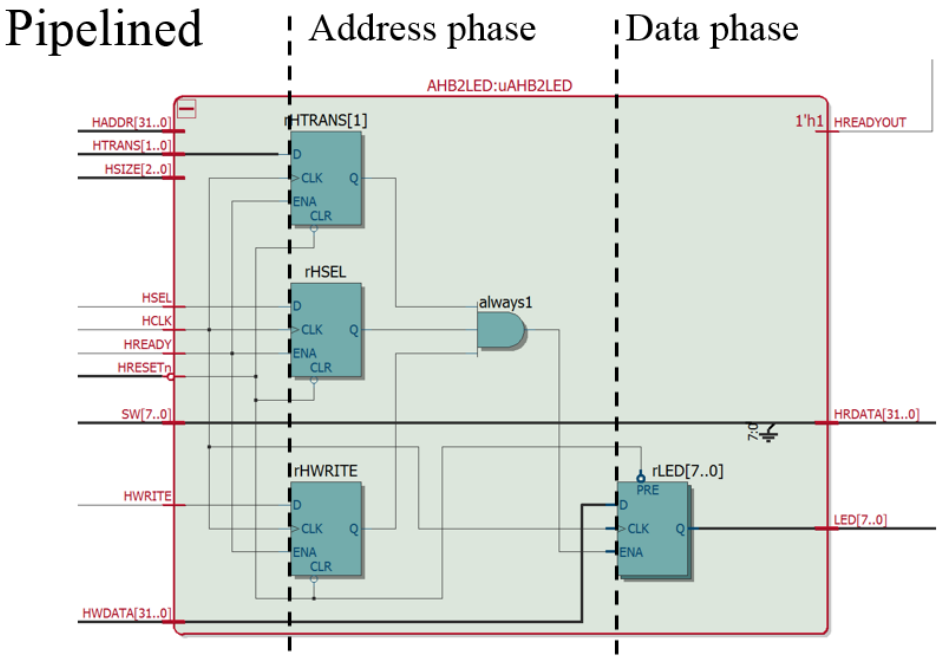
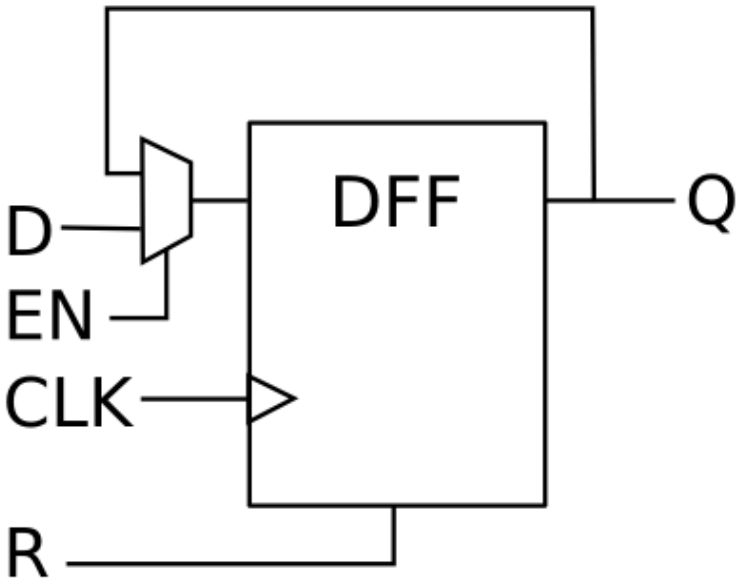


Figure 1 Block diagram of the peripheral to implement in your Lab4 design

The schematic diagram of the DFF with enable and reset, and appropriate Verilog code to implement it (from **https://www.eeweb.com/verilog-flip-flop-with-enable-and-asynchronous-reset/) are presented below:**




```
reg q;
always @ (posedge clk or posedge reset)
    if (reset)          // takes priority
        q <= 1'b0;
    else if (en)        //  no ELSE
        q <= d;
```

Figure 2 Schematic diagram of a D flip-flop with reset and enable

- Go through the file line by line, consult this lecture's notes and wiring diagram; replace all the **?...?** placeholders with the appropriate information
- show me (or better email me) this file for sign off

Assignment 2: Connecting constituents of a Cortex-M0 based SoC to the system bus (amend AHBLITE_SYS.v file)

You are going to amend the given top file in order to instantiate
- the CPU - the memory - the LED peripheral
- the address decoder, which generates peripheral select signals and controls the multiplexer at the input of the CPU
- the multiplexer that produces the system-wide ***HReady*** signal.

- Go through the file line by line, consult this lecture's notes and wiring diagram; replace all the **?...?** placeholders with the appropriate information

Assignment 3: Compile the amended design and run it on the FPGA board

Helpful hints:

- Make sure you go through both files without leaving any lines that require amends unedited
- Only when you fully amended both the files perform the compilation
- Then load the SOF file to the board
- Press KEY[0] to reset the SoC anytime
- TAKE YOUR TIME
- I will help you with the error messages
- When the design works, enter the last ten bits (least significant bits, LSBs) of your student ID number in binary using SW[9:0] and take a photograph of the board.

Lab 4

Personalisation: Your student ID number (in binary, 10 LSBs) =
Assignment 1 Screenshot of the Verilog code for the developed peripheral …
Assignment 2 Screenshot of the Verilog code for the **instantiation** of the memory module AHB2MEM_BRAM **ONLY** …
Assignment 3 Screenshot of the maximum clock frequency for the design … Photograph of the operating board where the 10 LSBs of your student ID number were set …
Your reflections based on the work that you have completed: - your overall assessment of the lab's difficulty level (leave only one) - Way too difficult Too difficult Comfortable Too easy Way too easy - what was more difficult to you (leave only one) – coding the peripheral OR connecting IPs - personal reflection on application of structural Verilog to SoC design … - any other comments regarding this lab …