EE540 Spring 2016, Assignment5, Due 11.5.2016

Synthesis & FPGA Implementation of the Microprocessor

The purpose of this CAD assignment is to synthesize the microprocessor you designed in the previous cad assignments and to actually implement your design on an FPGA. In this assignment you are given a sample code that you will run and see the results on the 7-segment displays of the Nexys3 board. You will execute each instruction one by one by pressing a button. Use the debouncer circuit from assignment 3 to generate the board_clk signal input of the mikrop module. The four 7-segment displays will be used to display the program counter (PC) and R0 register contents. Use two LEDs on the left for displaying the least significant byte of PC and two LEDs on the right for displaying the least significant byte of R0.

Send all your project files as an e-mail attachment. There is no report required for this assignment. If you can show that your design can be synthesized and can be implemented on the FPGA physically, you will get the full credit.

Your microprocessor will be wrapped inside the module mikrop given below:

```
module mikrop (RESET, SW2, board_clk, S);
input RESET;
input SW2;
input board_clk;
output [6:0] S;

…
//d input of the binary-to-BCD converter is the least significant bits of the contents of the R0
//SW2 input is connected to debouncer circuit.
//The output of the debouncer is the clk of the processor.
//The clock signal of the debouncer circuit is the 100 MHz signal generated on the Nexys3 board.
//RESET input is the global input to reset the microprocessor.
…
endmodule;
```

Test Instructions

Write the instructions below to your instruction memory to test your microprocessor:

---------Instructions to test the synthesized micro-processor----------
--Implemented on Xilinx Spartan 6 FPGA
--Copy the given 16-bit instructions to the corresponding addresses
--of your instruction RAM (Just the 16-bit number).

--	Instruction address	16-bit instruction	
MOVI hex(00) R0	00	1101 0000 0000 0000	displays 0
ADDI hex(01) R0	01	0101 0000 0000 0001	displays 1
SUBI hex(01) R0	02	1001 0000 0000 0001	displays 0
MOVI hex(02) R1	03	1101 0001 0000 0010	displays 0
MOV R1 R0	04	0000 0000 1101 0001	displays 2
ORI hex(03) R0	05	0010 0000 0000 0011	displays 3
ANDI hex(02) R0	06	0001 0000 0000 0010	displays 2
XORI hex(07) R0	07	0011 0000 0000 0111	displays 5
MOVI hex(06) R1	08	1101 0001 0000 0110	displays 5
OR R1 R0	09	0000 0000 0010 0001	displays 7
MOVI hex(01) R1	10	1101 0001 0000 0001	displays 7
XOR R1 R0	11	0000 0000 0011 0001	displays 6
ANDI hex(04) R0	12	0001 0000 0000 0100	displays 4
LSHI hex(01) R0	13	1000 0000 0000 0001	displays 8
AND R1 R0	14	0000 0000 0001 0001	displays 0
MOVI hex(01) R1	15	1101 0001 0000 0001	displays 0
MOVI hex(04) R0	16	1101 0000 0000 0100	displays 4
LSH R1 R0	17	1000 0000 0100 0001	displays 8
MOVI hex(09) R1	18	1101 0001 0000 1001	displays 8
MOV R1 R0	19	0000 0000 1101 0001	displays 9
MOVI hex(02) R1	20	1101 0001 0000 0010	displays 9
SUB R1 R0	21	0000 0000 1001 0001	displays 7
MOVI hex(0A) R1	22	1101 0001 0000 1010	displays 7
MOVI hex(00) R2	23	1101 0010 0000 0000	displays 7
STOR R1 R2	24	0100 0001 0100 0010	displays 7
LOAD R0 R2	25	0100 0000 0000 0010	displays A
CMPI hex(0A) R0	26	1011 0000 0000 1010	displays A
CMP R0 R1	27	0000 0000 1011 0001	displays A
BEQ hex(0B)	28	1100 0000 0000 1011	displays A
OR R0 R0	29	0000 0000 0010 0000	displays A
MOVI hex(0) R0	30	1101 0000 0000 0000	
MOVI hex(1) R1	31	1101 0001 0000 0001	
MOVI hex(2) R2	32	1101 0010 0000 0010	
MOVI hex(3) R3	33	1101 0011 0000 0011	
MOVI hex(4) R4	34	1101 0100 0000 0100	
MOVI hex(5) R5	35	1101 0101 0000 0101	
MOVI hex(6) R6	36	1101 0110 0000 0110	
MOVI hex(7) R7	37	1101 0111 0000 0111	
MOVI hex(8) R8	38	1101 1000 0000 1000	
MOVI hex(9) R9	39	1101 1001 0000 1001	
MOVI hex(0B) R0	40	1101 0000 0000 1011	displays B
MOVI hex(0C) R0	41	1101 0000 0000 1100	displays C
MOVI hex(0D) R0	42	1101 0000 0000 1101	displays D
MOVI hex(0E) R0	43	1101 0000 0000 1110	displays E
MOVI hex(0F) R0	44	1101 0000 0000 1111	displays F