BYOC\_HW4:

Tsahi Barshevsky 311334544

Kostya Lokshin 310765821

3.1) Attach screen captures showing ck cycles 70 to 118 (following the end of the reset pulse –

use the i signal value of the TB to count the CK cycles) and make the values of all signals

connected to rdbk0-15 readable. All of the signals mentioned in 2.c should be presented in

the screen capture. The signal names should be readable as well. These are not TB\_rdbk0 -

TB\_rdbk15 of the TB or rdbk0\_out\_to\_TB-rdbk15\_out\_to\_TB of HW4\_top\_4sim, but PC\_reg,

IR\_reg etc. You may use the signals rdbk3\_vec, rdbk4\_vec, rdbk7\_vec & rdbk13\_vec.

3.2) Explain in details what you see in this screen capture: You should show the execution of the

program using a scheme as described below. Specifically you should show the values of

registers $16 and $15 and to specify the point in time where they changes (in a similar

manner to the drawing below)

Val in $16 ?? 0

Val in $15 ?? 3

PC= 400130 inst= add $16,$0,$0

PC= 400134 inst= addi $15,$0,3

PC= 400138 ….

.

.

.

Put a line like that when $16 or $15 are updated

Also mark the ALUOUT\_reg value at the WB stage when $16 is written to in **RED** (in a similar

manner to the drawing above) on the screen capture. And mark the ALUOUT\_reg value at the

WB stage when $15 is written to in **BLUE**.

Also mark the IR value of branch instructions in **ORANGE** and state whether we branch or not at

that instruction. You can use Excel for drawing this and attach the XL file in the Doc directory.

The purpose of this is to make you familiar in looking at the waveforms and understanding what

happens in the design via simulations. This knowledge is required in HW5.

IF ID EX WB

IF ID EX WB

IF ID EX WB

IF ID EX WB

11

3.3) Explain in detail the changes you did in the Fetch\_unit to support beq and bne instructions.

Attach the relevant vhdl code.

3.4) Why do we have two nops in the following code (which is part of the code use in the

simulation phase):

x"004000A0" => x"00230820"

x"004000A4" => x"00000000"

x"004000A8" => x"00000000"

x"004000AC" => x"00810820"

x"004000B0" => x"00000000"

x"004000B4" => x"00000000"

x"004000B8" => x"00A10820"

x"004000BC" => x"00000000"

x"004000C0" => x"00000000"

x"004000C4" => x"00260820"

3.5) The program we load into the IMem checks many functions & instructions. We do check the

sign extension for example. Which of the instructions or functions of the Rtype MIPS CPU

you built are not checked by the program we have in the IMem (the program we disassembled

in the report of HW3 and which is part of the HW4\_BYOC\_Host\_Intf\_4sim.vhd

file) ?