BYOC\_HW2:

Tsahi Barshevsky 311334544

Kostya Lokshin 310765821

3.1) Draw on paper a block diagram describing the inside of HW2\_top entity. Specify all

connections of the sub components between themselves and between them and the i/o pins

of the HW2\_top entity. [This is meant to make you familiar with the i/o pin names, the signals,

the port mapping of the components, i.e., to understand what is connected to what]

It is similar to HW2\_top seen in Fig.2 of this document except you are not supposed to show

the inner parts of the components (e.g., the Fetch\_Unit), and you are supposed to show **all**

signals and i/o pins of HW2\_top and all components inside HW2\_top. The i/o pins names of a

component should be written inside the component (e.g., CK\_25MHz\_OUT in the

Clock\_Driver). The HW2\_top i/o pin names should be written outside the HW2\_top entity.

The signal names should be written next to the signal wires. You should gather bus lines

together and specify them accordingly (e.g., bus(15:0)\_example for the signals

bus0\_example, bus1\_example, … , bus15\_example). You may base your drawing on Fig.2 of

the 14\_BYOC\_HW\_infrastructure.doc file.

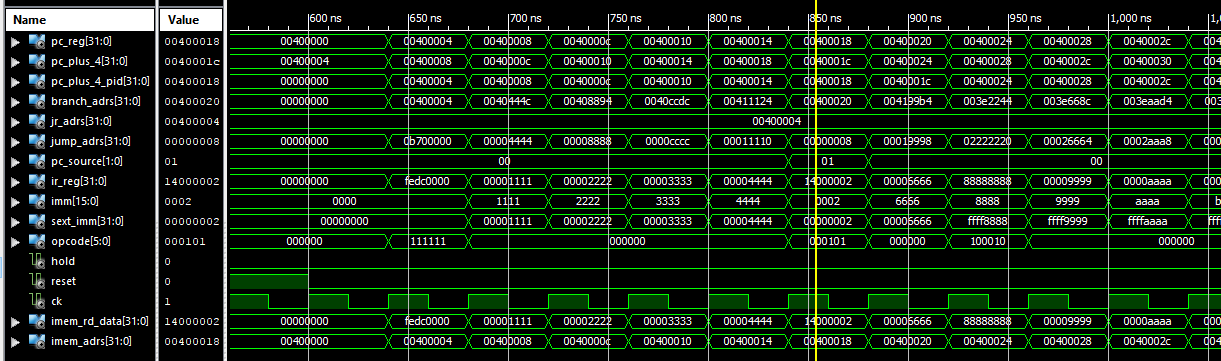
[The block diagram is a PDF file in this folder](HW2_Top.pdf)

3.2) You need to attach a doc file with screen captures describing the simulation you made.

All signals mentioned in section 1a above should be presented in the screen capture. Show

at least the 1st 10 ck cycles following the end of the reset pulse and make the values of all

signals readable.



3.3) Say we have a value V in location Adrs in the IMem. What will be the expected values of the

PC\_reg and PC\_plus\_4\_pID signals when the IR\_reg signal has the value V? Adrs? Adrs-4?

Adrs+4? Other? Explain your reply.

PC\_reg = Adrs

PC\_plus\_4\_pID = Adrs : whan PC\_source = 00 or 01 or 11

PC\_plus\_4\_pID = Adrs – 4 : when PC\_source = 10 (JR instruction = x"00400004"(

3.4) In the IMem we have the value of a bne instruction (x"14000002") in address x”400014”.

Write down all of the fields of this instruction (opcode, Rs, ..., etc.) Do we expect to actually

branch? Who checks the condition?

Opcode = 000101

Rs = 00000

Rt = 00000

imm = x"02"

because the condition of bne is not met then branch\_adr = PC + 4 = x"00400020"

because this is bne instruction then PC\_source = 01 therefore PC = branch\_adr = x"00400020"

3.5) Did we branch in the simulation? Check the simulation and write down the sequence of the

instructions (the PC\_reg values) from address x”400008” until 4 addresses after reaching the

branch target.

Did the branch occur in the address we expect it to occur in (if this was a single cycle or

multi-cycle implementation)? Explain what happened.

Following this write the assembly code of the smallest possible loop (that is not an endless

loop, i.e., it decrements or increments a counter) and explain how it works.

PC\_reg:

00400008 at 680 ns

0040000c

00400010

00400014

00400018 at 840 ns, opcode is 000101 : branch not equal code

Imm is 14000002: Rs=Rt=00000 therefore branch will not occur and branch target is PC=PC+4

0040001c

00400024

00400028

0040002c

1. li $t1, 0 # temp var counter for loop
2. li $t2, 10 # end of counter condition
3. Loop:
4. beq $t2, $t1, Next #at 10 we go to Exit, defined below
5. addi $t1, $t1, 1 #increment counter
6. j Loop # jumps back to the top of loop
7. Next: # continue program