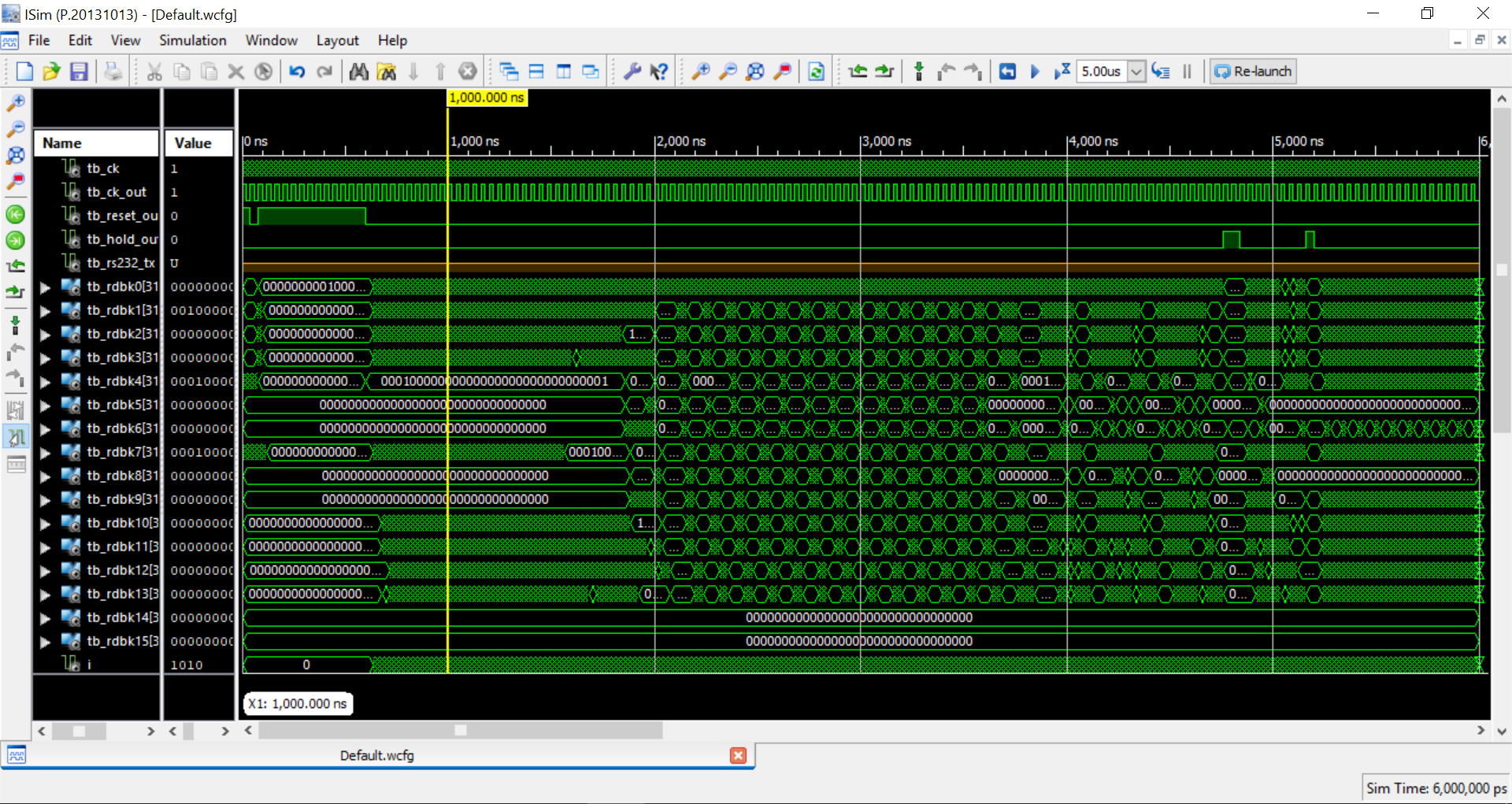
R-Type MIPS CPU

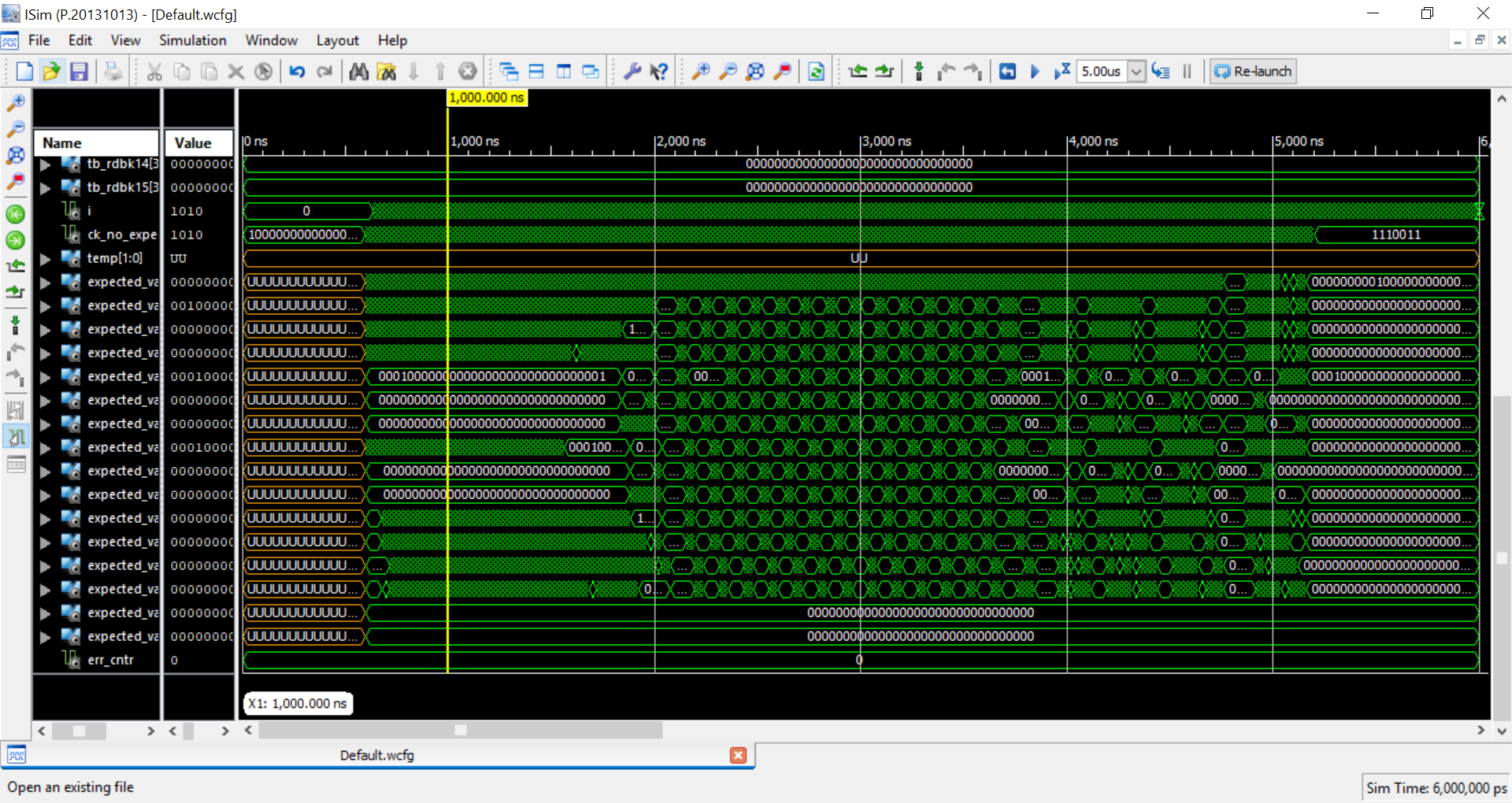
Simulation report

Otabek Sherman – 326868775

Lidor Cohen – 305782732

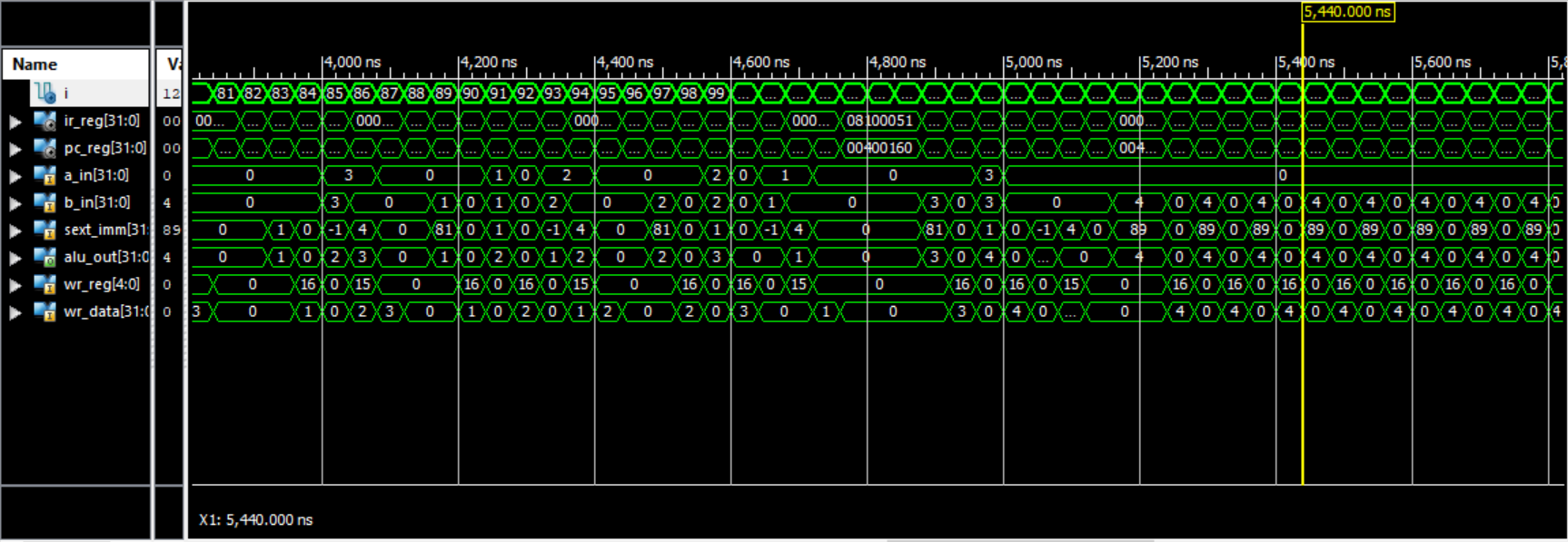
Question 3.1





The simulation for 6us

Question 3.2

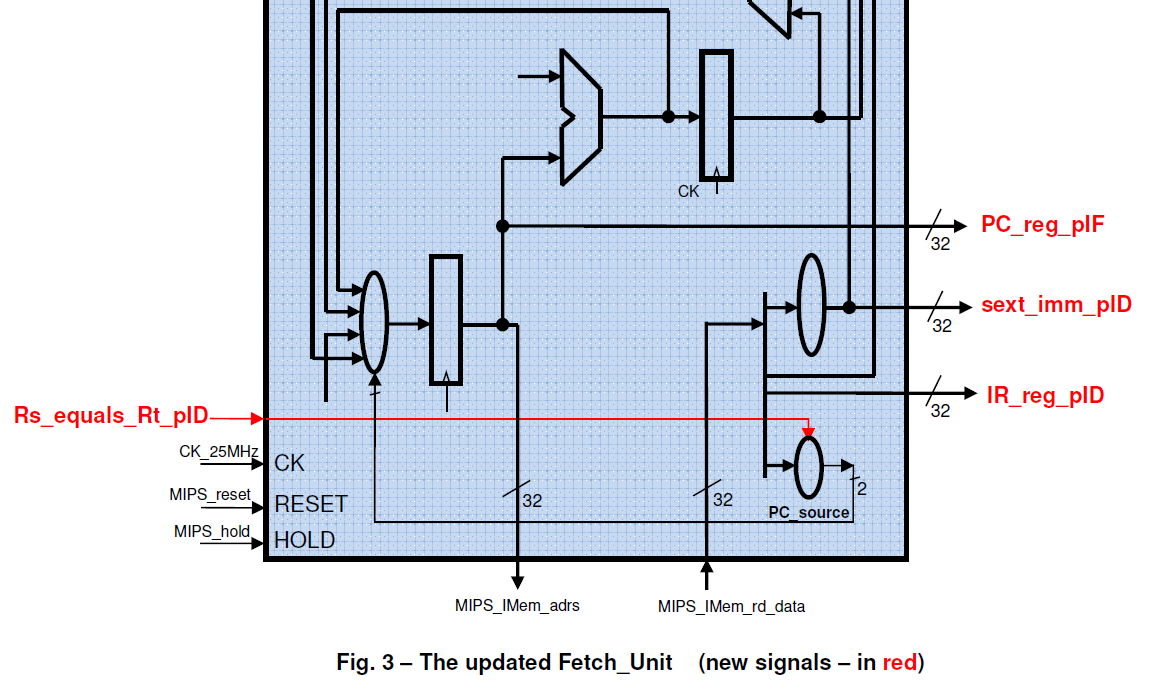


Ck cycles 70 to 118

\*The scheme with execution of the program is in the Excel file.

Question 3.3

To support beq and bne instructions we added 4 new signals in the Fetch\_unit and changed the PC\_source decoder process:



Input signals:

* Rs\_equals\_Rt\_pID

Output signals:

* IR\_reg\_pID
* sext\_imm\_pID
* PC\_reg\_pIF

-- PC\_source decoder (create the PC\_source signal)

process (opcode, funct, Rs\_equals\_Rt\_pID)

begin

case opcode is

-- j, jal - 11.

when b"000010" => PC\_Source <= b"11";

when b"000011" => PC\_Source <= b"11";

-- beq, bne - 01.

when b"000100"=>

if Rs\_equals\_Rt\_pID = '1' then

PC\_Source <= b"01";

else

PC\_Source <= b"00";

end if;

when b"000101" =>

if Rs\_equals\_Rt\_pID = '1' then

PC\_Source <= b"01";

else

PC\_Source <= b"00";

end if;

-- RType.

when b"000000" =>

-- jr - 10.

if funct = b"001000" then

PC\_Source <= b"10";

-- Any other instruction - 00.

else

PC\_Source <= b"00";

end if;

when others => PC\_Source <= b"00";

end case;

end process;

Question 3.4

We have two nops in the following code because we need to wait for done the branch instruction, that takes 2 cycles. I takes 2 cycles because in one clock cycle we store the value of the R1 and after that we need to wait 2 command to use the R1 value again.

Question 3.5

The program we have in IMem doesn’t check the next instructions/functions: sub, and, or, xor, slt, bne.