- **<u>Detailed explanation</u>**:

All changes related to the new instruction are tagged with comment: *// ADDIUPC* in the soft-core. We must perform the following actions:

- Inhibit "*Reserved Instruction Exception*":

Change:

assign maj_ri_e = … (mpc_ir_e[31:28] == 4'b011_0) ||

For:

assign maj_ri_e = ...
((mpc_ir_e[31:28] == 4'b011_0) && (mpc_ir_e[27:26] != 2'b00)) || *// ADDIUPC*

- Change signals:
  - ***alu_sel_e***: This signal is registered to the next stage through the Main Pipeline Registers. Then, at m14k_edp, when it is 1, it selects the ALU for output at the M-Stage.
    Set this signal to 1 when an addiupc is found:

((mpc_ir_e[31:26] == 6'o30)) || *// ADDIUPC*

  - ***maj_vd_e***: This signal controls the RF write strobe at A/W-Stage. Set to 1 when addiupc found:

((mpc_ir_e[31:26] == 6'o30)) || *// ADDIUPC*

- New signal:
  - ***addiupc_instr***: Set to 1 when addiupc found:

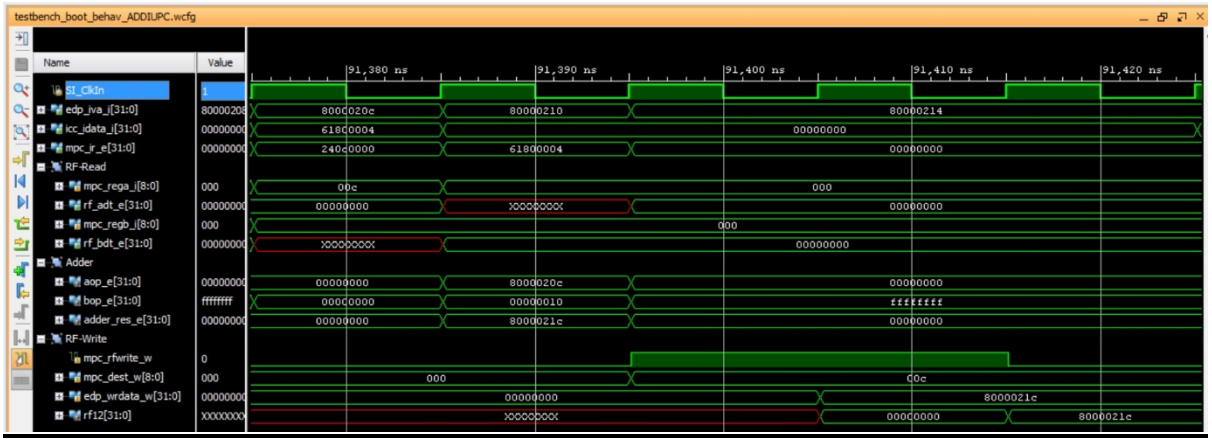
assign addiupc_instr = (mpc_ir_e[31:26] == 6'o30);

- Change signal ***dest_e*** based on the new signal: We must select Rs as Destination Register:

assign dest_e [5:0] = addiupc_instr ? { 1'b0, mpc_ir_e[25:21] } : ((special_e | spec2_e | mpc_cnvt_e | rwpgpr_e | dest_cnvt_dsp_e | lx_e) ? { rwpgpr_e & ~rdpgpr_e, pdest_e } : ((lnk31_e) ? 6'h1f : { 1'b0, src_b_e[4:0] })); // ADDIUPC

- Incorporate new functionality:
  - We must insert the new operands as an input to the "*m14k_edp_add_simple*" adder. ***aop_e*** and ***bop_e*** are the inputs to the adder:
    a. ***aop_e***: Incorporate the PC as an input. The PC is already available at E-Stage in signal ***iva_e***.
    *assign **aop_e** [31:0] = addiupc_instr ? iva_e : (icc_pcrel_e ? (iva_e & ~{30'b0, {2{icc_pcrel_e}}}) : edp_abus_e); // ADDIUPC*
    b. ***bop_e***: Incorporate the SignExt(Imm<<2) as an input.
    *assign **bop_e** [31:0] = addiupc_instr ? { {11{mpc_ir_e[18]}}, mpc_ir_e[18:0], 2'b0} : {32{mpc_subtract_e}} ^ bbus_imm_e; // ADDIUPC*

- **EXAMPLE - SIMULATION**:



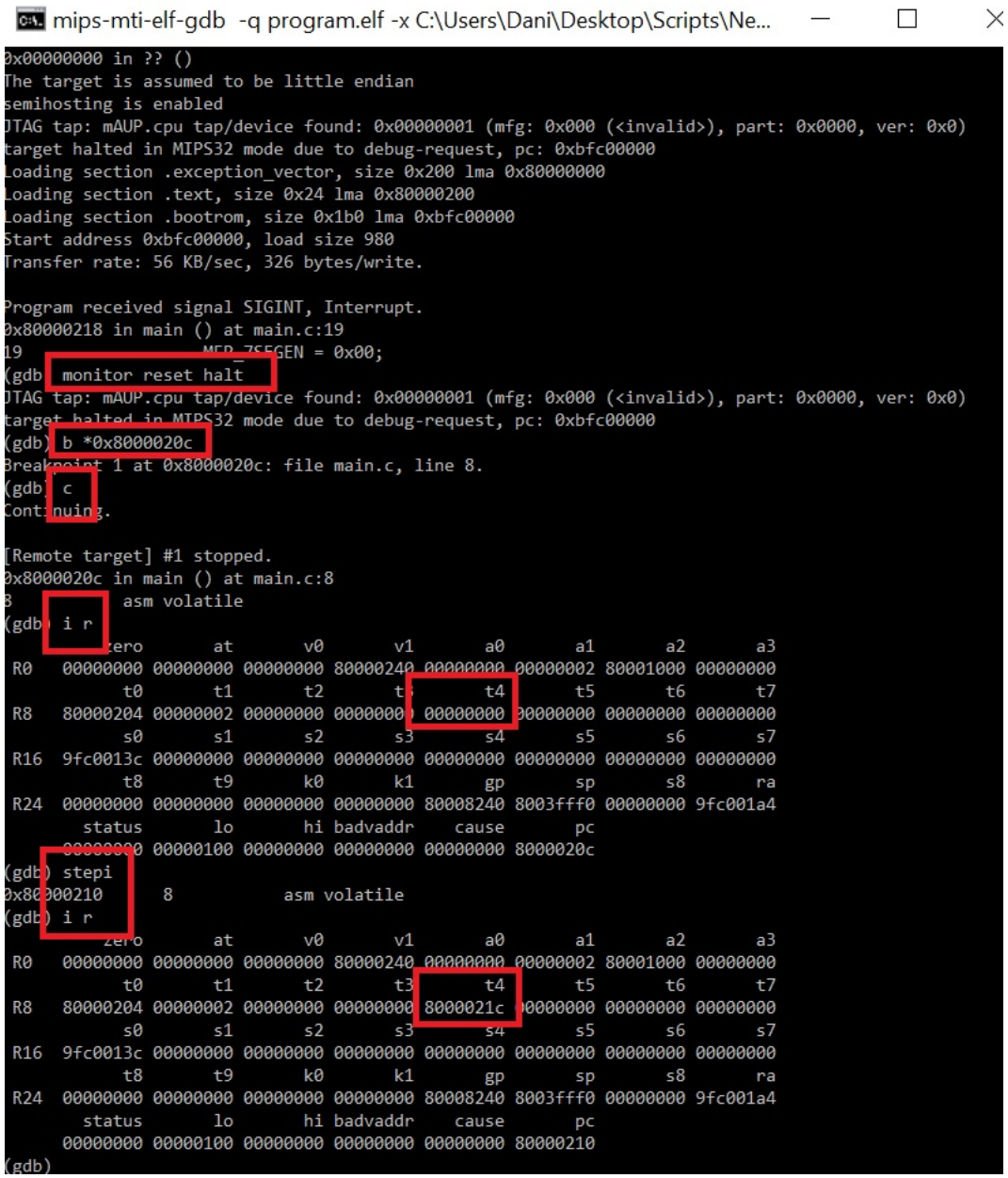
Observe that in the 5th cycle rf12($t4)=0x800002ac.

- **EXAMPLE – EXECUTION ON THE BOARD**:

When the program is downloaded on the board, you should see on the 7-seg displays:

- 7-seg displays=$t4, which in our example is 0x8000021c

Then, when you debug the program following the steps stated in the document, you should observe the following:


mips-mti-elf-gdb -q program.elf -x C:\Users\Dani\Desktop\Scripts\Ne...
0x00000000 in ?? ()
The target is assumed to be little endian
semihosting is enabled
JTAG tap: mAUP.cpu tap/device found: 0x00000001 (mfg: 0x000 (<invalid>), part: 0x0000, ver: 0x0)
target halted in MIPS32 mode due to debug-request, pc: 0xbfc00000
Loading section .exception_vector, size 0x200 lma 0x80000000
Loading section .text, size 0x24 lma 0x80000200
Loading section .bootrom, size 0x1b0 lma 0xbfc00000
Start address 0xbfc00000, load size 980
Transfer rate: 56 KB/sec, 326 bytes/write.
Program received signal SIGINT, Interrupt.
0x80000218 in main () at main.c:19
19 MFP_ZSEGEN = 0x00;
(gdb) monitor reset halt
JTAG tap: mAUP.cpu tap/device found: 0x00000001 (mfg: 0x000 (<invalid>), part: 0x0000, ver: 0x0)
target halted in MIPS32 mode due to debug-request, pc: 0xbfc00000
(gdb) b *0x8000020c
Breakpoint 1 at 0x8000020c: file main.c, line 8.
(gdb) c
Continuing.
[Remote target] #1 stopped.
0x8000020c in main () at main.c:8
8 asm volatile
(gdb) i r
zero at v0 v1 a0 a1 a2 a3
R0 00000000 00000000 00000000 80000240 00000000 00000002 80001000 00000000
t0 t1 t2 t3 t4 t5 t6 t7
R8 80000204 00000002 00000000 00000000 00000000 00000000 00000000 00000000
s0 s1 s2 s3 s4 s5 s6 s7
R16 9fc0013c 00000000 00000000 00000000 00000000 00000000 00000000 00000000
t8 t9 k0 k1 gp sp s8 ra
R24 00000000 00000000 00000000 00000000 80008240 8003fff0 00000000 9fc001a4
status lo hi badvaddr cause pc
00000000 00000100 00000000 00000000 00000000 8000020c
(gdb) stepi
0x80000210 8 asm volatile
(gdb) i r
zero at v0 v1 a0 a1 a2 a3
R0 00000000 00000000 00000000 80000240 00000000 00000002 80001000 00000000
t0 t1 t2 t3 t4 t5 t6 t7
R8 80000204 00000002 00000000 00000000 8000021c 00000000 00000000 00000000
s0 s1 s2 s3 s4 s5 s6 s7
R16 9fc0013c 00000000 00000000 00000000 00000000 00000000 00000000 00000000
t8 t9 k0 k1 gp sp s8 ra
R24 00000000 00000000 00000000 00000000 80008240 8003fff0 00000000 9fc001a4
status lo hi badvaddr cause pc
00000000 00000100 00000000 00000000 00000000 80000210
(gdb)