- **EXPLANATION**:

- **m14k_mpc_dec:**
  - Disable exception for the funct field that we will use (*funct = 011100*).

```
        assign spec_ri_e =    special_e && ((mpc_ir_e[5:0] == 6'o05) ||
                              (mpc_ir_e[5:0] == 6'o16) ||
                              (mpc_ir_e[5:2] == 4'b010_1) ||
                              (mpc_ir_e[5:0] == 6'b011_101) ||              // LWI
                              (mpc_ir_e[5:1] == 5'b011_11) ||               // LWI
                              (mpc_ir_e[5:1] == 5'b101_00) ||
                              (mpc_ir_e[5:2] == 4'b101_1) ||
                              (mpc_ir_e[0] && mpc_ir_e[5:2] == 4'b110_1) ||
        ((mpc_ir_e[5:3] == 3'o2) & ~edp_dsp_present_xx &
      (mpc_ir_e[0] ? (mpc_ir_e[12:11] != 2'b00) :          // 11, 13 (HI/LO 1-3)
                            (mpc_ir_e[22:21] != 2'b00))) ||
                              (mpc_ir_e[5:3] == 3'o7));
```

  - Control signals:
    - **load_e**:

```
assign load_e =         !(mpc_ldst_e & ~(mpc_ir_e[31:26] == 6'o23) ?
(mpc_ir_e[29] || mpc_atomic_m) :
(mpc_ir_e[31:26] == 6'o23) ? mpc_ir_e[3] :
(mpc_ir_e[23:21] == 3'b011) ? mpc_ir_e[9] : mpc_ir_e[23]) ||
mpc_lxs_e || lx_e || cop_e || synci_e || mpc_atomic_e || lwi_instr_e;     // LWI
```

    - **mpc_ldst_e**:

```
assign mpc_ldst_e =  lwi_instr_e || mpc_lxs_e || mpc_ir_e[31] || (regimm_e
&& (mpc_ir_e[20:16] == 5'o07)) || lx_e || (mpc_ir_e[31:26] == 6'o23) &&
!mpc_ir_e[5];
```

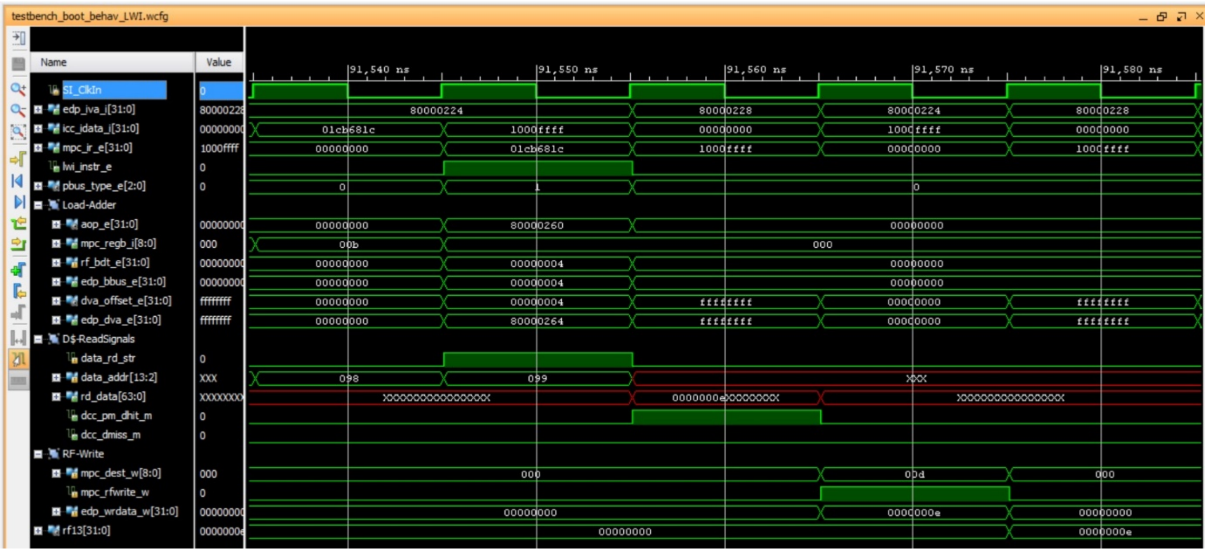
    - **lwi_instr_e**:

```
assign lwi_instr_e = special_e && (mpc_ir_e[5:0] == 6'b011_100);       // LWI
```

- **m14k_edp:**
  - Select register for lwi instruction:

```
        assign dva_offset_e[31:0] = lwi_instr_e ? edp_bbus_e :
!mpc_selimm_e ? bbusis_e
: (mpc_atomic_e || mpc_atomic_m) ? {{20{mpc_imsgn_e}}, mpc_ir_e[11:0]}
: {{16{mpc_imsgn_e}}, mpc_ir_e[15:0]};
```

- **EXAMPLE - SIMULATION**:



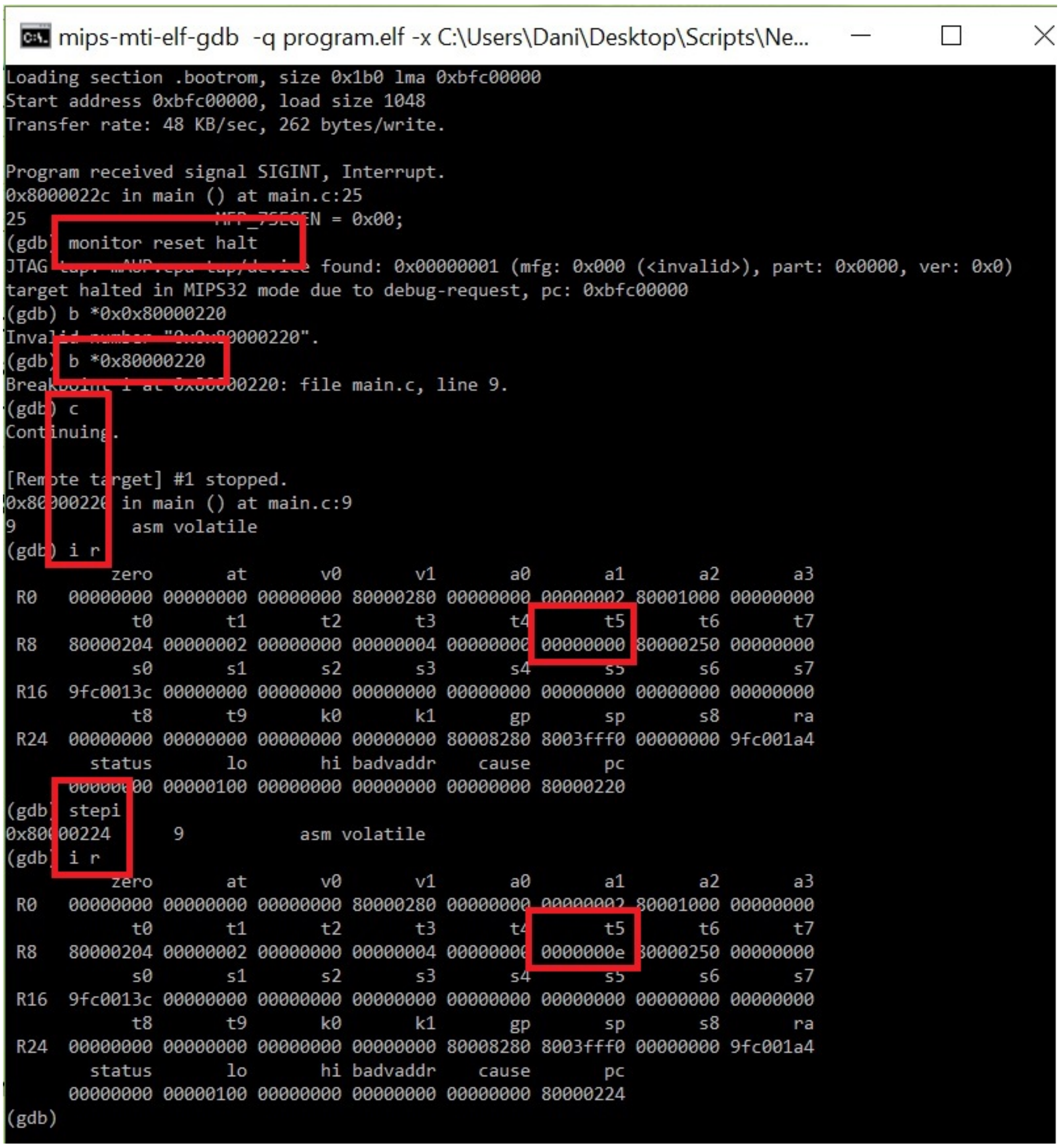
Observe that in the fifth cycle rf13=0xe, as this is the value read from the D$.

- **EXECUTION ON THE BOARD**:

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

- 7-seg displays=$t5, which in our example is 0x0000000e

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...

```
Loading section .bootrom, size 0x1b0 lma 0xbfc00000
Start address 0xbfc00000, load size 1048
Transfer rate: 48 KB/sec, 262 bytes/write.

Program received signal SIGINT, Interrupt.
0x8000022c in main () at main.c:25
25                MFP_7SEGEN = 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 *0x0x80000220
Invalid number "0x0x80000220".
(gdb) b *0x80000220
Breakpoint 1 at 0x80000220: file main.c, line 9.
(gdb) c
Continuing.

[Remote target] #1 stopped.
0x80000220 in main () at main.c:9
9           asm volatile
(gdb) i r
          zero       at       v0       v1       a0       a1       a2       a3
 R0   00000000 00000000 00000000 80000280 00000000 00000002 80001000 00000000
            t0       t1       t2       t3       t4       t5       t6       t7
 R8   80000204 00000002 00000000 00000004 00000000 00000000 80000250 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 80008280 8003fff0 00000000 9fc001a4
        status       lo       hi badvaddr    cause       pc
      00000000 00000100 00000000 00000000 00000000 80000220
(gdb) stepi
0x80000224      9           asm volatile
(gdb) i r
          zero       at       v0       v1       a0       a1       a2       a3
 R0   00000000 00000000 00000000 80000280 00000000 00000002 80001000 00000000
            t0       t1       t2       t3       t4       t5       t6       t7
 R8   80000204 00000002 00000000 00000004 00000000 0000000e 80000250 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 80008280 8003fff0 00000000 9fc001a4
        status       lo       hi badvaddr    cause       pc
      00000000 00000100 00000000 00000000 00000000 80000224
(gdb)
```