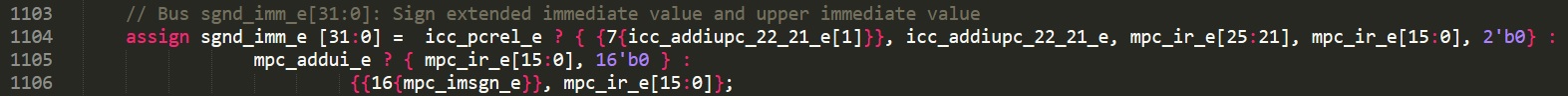
* + ***SUB* instruction**
    - The same adder as the one used for the *ADD* instruction is used here (*m14k\_edp\_add\_simple*).
    - In case of a SUB instruction, Source Operand B is *2’s complemented* before being provided to the adder:
      1. Signal ***mpc\_subtract\_e*** is set to 1 for SUB instructions in *m14k\_mpc\_dec* module.
      2. Source Operand B is inverted when ***mpc\_subtract\_e***=1 before going into the adder (*1s complement*):

***bop\_e***[31:0] = {32{***mpc\_subtract\_e***}} ^ ***bbus\_imm\_e***

* + - 1. A Carry-In is inserted in the adder for performing the *2’s complement*: *.ci(****mpc\_subtract\_e****)*.
  + ***ADDIU* instruction**
    - Source B is provided from signal ***sgnd\_imm\_e*** instead of signal ***edp\_bbus\_e*** (from the RF) at multiplexer *\_bbus\_imm\_e\_31\_0\_* (Figure 5).
    - Signal ***sgnd\_imm\_e*** is computed in the following multiplexer:



where:

* + - 1. ***icc\_pcrel\_e*** is always 0.
      2. ***mpc\_addui\_e***=0 for ADDIU instructions, as determined in module *m14k\_mpc\_dec*.

thus:

***sgnd\_imm\_e*** = *{{16{****mpc\_imsgn\_e****}},* ***mpc\_ir\_e****[15:0]}*

where:

* + - 1. ***mpc\_imsgn\_e*** is just equal to ***mpc\_ir\_e***[15] in this case.
  + ***SLT* instruction**
    - The result of a *slt* instruction is computed at module *m14k\_edp* in signal ***bit0\_m***, as follows:

assign bit0\_m = mpc\_signed\_m ? (pro31\_m ^ car31\_m) : ~car31\_m;

2 cases are possible: signed instructions (***mpc\_signed\_m***=1), such as *slt* or *slti*, and unsigned instructions (***mpc\_signed\_m***=0), such as *sltu* or *sltiu*.

* Signal ***mpc\_signed\_m*** is registered from signal ***signed\_e***, computed at ***m14k\_mpc\_dec*** at the E-Stage as:

assign signed\_e = (mpc\_ir\_e[29] & ~mpc\_ir\_e[26]) | (~mpc\_ir\_e[29] & ~mpc\_ir\_e[26] & ~mpc\_ir\_e[0]) | (~mpc\_ir\_e[29] & mpc\_ir\_e[26] & ~mpc\_ir\_e[16]);

* Note that ***car31\_m*** is the carry out of the Adder (module m14k\_edp\_add\_simple) and that:

pro31\_e = aop\_e[31] ^ bop\_e[31];

Note also that the adder performs a subtraction for any SLT-Type instruction, given that:

assign mpc\_subtract\_e =

(mpc\_ir\_e[31:27] == 5'h05) || // SLTI(U)

((mpc\_ir\_e[31:26] == 6'h1) && // RegImm

(mpc\_ir\_e[20:19] == 2'b01)) || // trap immed

((mpc\_ir\_e[31:26] == 6'h0) && // SPECIAL

(mpc\_ir\_e[5:1] != 5'b10000)); // Add(U)

* + - This result is assigned to signal ***res\_m*** in module *m14k\_edp\_buf\_misc\_pro* whenever we have this kind of instructions, computed by the following two multiplexers:

mvp\_mux2 #(32) \_udislt\_m\_31\_0\_(udislt\_m[31:0],mpc\_udisel\_m && edp\_udi\_present, {31'h0, bit0\_m}, UDI\_data\_m[31:0]);

mvp\_mux2 #(32) \_res\_m\_31\_0\_(res\_m[31:0],mpc\_udislt\_sel\_m, asp\_m[31:0], udislt\_m[31:0]);

Signal ***mpc\_udislt\_sel\_m*** is computed at *m14k\_mpc\_ctl* as follows:

assign mpc\_udislt\_sel\_m = slt\_sel\_m | mpc\_udisel\_m;

Signal ***slt\_sel\_m*** is registered from signal ***slt\_sel\_e***, computed at *m14k\_mpc\_dec* as follows:

assign slt\_sel\_e = (mpc\_ir\_e[31:27] == 5'b001\_01) || // slti, sltiu

special\_e && (mpc\_ir\_e[5:1] == 5'b101\_01) || // slt, sltu

special\_e && (mpc\_ir\_e[5:2] == 4'b110\_0) || // tge, tgeu, tlt, tltu

regimm\_e && (mpc\_ir\_e[20:18] == 3'o2); // tgei, tgeiu, tlti tltiu

Thus, for a slt instruction, control signal ***slt\_sel\_e***=1