* ***mpc\_alufunc\_e***.

SOLUTION:

* + ***mpc\_alufunc\_e***, is computed within module *m14k\_mpc\_*dec. Note that, for special logic instructions (for which ***mpc\_ir\_e****[5]=1*): ***mpc\_alufunc\_e****=****mpc\_ir\_e****[1:0]*.

According to Table A.3 in document “MIPS Architecture For Programmers Volume II-A: The MIPS32 Instruction Set (MD00086)”:

* + - ***mpc\_ir\_e****[1:0]*=00 for an AND instruction.
    - ***mpc\_ir\_e****[1:0]*=01 for an OR instruction.
    - ***mpc\_ir\_e****[1:0]*=10 for an XOR instruction.
    - ***mpc\_ir\_e****[1:0]*=11 for an NOR instruction.

which is perfectly coherent with multiplexer “\_logic\_out\_e\_31\_0\_” in Figure 14.

* ***mpc\_sellogic\_m***.

SOLUTION:

* + ***mpc\_sellogic\_m*** is computed within module *m14k\_mpc\_ctl*. This signal is registered from the E-Stage through the Pipeline Registers (***ie\_pipe\_in***[`M14K\_IE\_SELLOG] 🡪 ***ie\_pipe\_out***[`M14K\_IE\_SELLOG]). It comes from signal ***sel\_logic\_e***, computed within module *m14k\_mpc\_dec*. This signal is 1 for special logic instructions (***mpc\_ir\_e***[5:2] == 4'b100\_1), and 0 for arithmetic instructions, which is perfectly coherent with multiplexer “\_edp\_alu\_m\_31\_0\_” in Figure 14.