

- *mpc\_eqcond\_e / eqcond\_e*

SOLUTION:

- Let's analyze these signals for the specific case of a *beq* instruction (*br\_eq* = 1 and *cnd\_eq\_en*=1).

```
1895 // br_eq: Branch on equal (+L)
1896 assign br_eq = (mpc_ir_e[31:26] == 6'o04) || (mpc_ir_e[31:26] == 6'o24);
2613 // Condition will be true if edp_cndeq_e == 1
2614 assign cnd_eq_en = mpc_irval_e & (br_eq | br_le);
```

- If the equality compare result, computed at *m14k\_edp* as explained above, is 1 (*edp\_cndeq\_e*=1), then *eqcond\_e* and *mpc\_eqcond\_e* are set to 1, given that *cnd\_eq\_en*=1 for a *beq* instruction (omitting other signals also involved such as *mpr\_run\_ie*, *mpc\_irval\_e*, etc.).
- If the equality compare result is 0 (*edp\_cndeq\_e*=0), then *eqcond\_e* and *mpc\_eqcond\_e* are set to 0.

```
2623 assign eqcond_e = edp_cndeq_e ? ((cnd_eq_en | eq_cond_noce) & mpc_run_ie) :
2624 ((cnd_neq_en | eq_cond_noce) & mpc_run_ie);
2625 assign mpc_eqcond_e = (eqcond_e | edp_dsp_pos_ge32_e & dec_redirect_bposge32_e & mpc_irval_e & mpc_run_ie)
2626 & ~mpc_int_pref_phase1;
```