

Hardware as Software

VHDL and Using FPGAs as Software

This VHDL code was extracted from a flight project. The first version apparently had a problem, and was commented out and rewritten:

```
--      HOLD2 <= (((HLD2TX_N_Q AND O_EN_Q(2)) OR
--      (HLDTX_N_Q AND O_EN_Q(1)) OR
--      (ROFRDY_N_Q AND O_EN_Q(0)))) AND
--      NOT(BYPASS_EN_Q AND (HLDTX_N_Q AND O_EN_Q(1))));

      HOLD2 <= (((((HLD2TX_N_Q AND O_EN_Q(2)) OR
(HLDTX_N_Q AND O_EN_Q(1)) OR
(ROFRDY_N_Q AND O_EN_Q(0)))) AND
NOT(BYPASS_EN_Q AND (HLDTX_N_Q AND O_EN_Q(1))))) OR
      (((HLD2TX_N_Q AND O_EN_Q(2)) OR (HLDTX_N_Q AND
O_EN_Q(1))) AND (BYPASS_EN_Q AND HLDTX_N_Q AND
O_EN_Q(1))));
```

Examining the New Code:

```
HOLD2 <= (((((HLD2TX_N_Q AND O_EN_Q(2)) OR
(HLDTX_N_Q AND O_EN_Q(1)) OR
(ROFRDY_N_Q AND O_EN_Q(0))) AND
NOT(BYPASS_EN_Q AND (HLDTX_N_Q AND O_EN_Q(1)))) OR
(((HLD2TX_N_Q AND O_EN_Q(2)) OR (HLDTX_N_Q AND O_EN_Q(1)))
AND (BYPASS_EN_Q AND HLDTX_N_Q AND O_EN_Q(1)))));
```

Simplify the code by replacing the variables:

```
a=HLD2TX_N_Q AND O_EN_Q(2), b=HLDTX_N_Q AND O_EN_Q(1)
c=ROFRDY_N_Q AND O_EN_Q(0), d=BYPASS_EN_Q
```

The equation then becomes:

```
HOLD2 <= (((((a) OR (b) OR (c)) AND NOT(d AND (b)))) OR ((a) OR (b))
AND (d AND b)));
```

Which reduces to `HOLD2<= a OR b OR c;`

This raises some questions:

- Is the new HOLD2 equation correct?
- If not, how much simulation would it take to catch this?
- If so, what happened to `d=BYPASS_EN_Q`?
- Is this the result of:
 - Typing error?
 - Just trying things to make it work without any real thought?
- How would this be different with schematics?
 - Do schematics encourage logic reduction more than VHDL?
 - Do schematics discourage “trying things” to make the simulation work out more than VHDL?

Some Lessons Learned:

- Reduce your equations.
- Don't just try things, **THINK** about what you're doing
- Rely more on your own logical thought processes than simulations

This code is currently flying.

Is it doing the right thing?