PCI_TARGET_UNIT

PCI_TARGET32_SM

IRDY

pciu_pciif_irdy_in

pciu_pciif_irdy_reg_in

pci_irdy_in

pci_irdy_reg_in

```
trdy_critical (combinatorial)

assign pci_trdy_out = ~(trdy_w ||
(trdy_w_frm && ~pci_frame_in) ||
(trdy_w_frm_irdy && ~pci_frame_in &&
pci_irdy_in))
```

POTENTIAL PROBLEM #1:
Since TRDY is assigned based on the IRDYnon-registered input, then pci_trdy_out, even if it is subsequently registered, may be too short to allow the rest of the logic to work correctly when IRDY & TRDY are crowding CLK from opposite directions.

- stop_critical (combinatorial)
- devs_critical (combinatorial)
- backoff (registered)

(pci_irdy_reg_in used in these wires/assigns)

```
wire trdy_w_frm
assign load_medium_reg_out
assign sel_fifo_mreg_out
wire ctrl_en
assign fetch_pcir_fifo_out
assign load_to_pciw_fifo_out
wire fast_back_to_back
assign load_to_conf_out
```

```
assign load_to_conf_out =
((state_transfere_reg && cnf_progress &&
rw_cbe0 && ~pci_irdy_reg_in &&
~bckp_trdy_reg) || (state_transfere_reg
&& norm_access_to_conf_reg && rw_cbe0 &&
~pci_irdy_reg_in && ~bckp_trdy_reg)) ;
```

WHERE I STARTED FROM:
Basically, what happened is that I am unable to write to the command register in PCI space yet in order to set the memory enable bit, or change the BAR0. In searching why, I can see that load_to_conf_out does not assert. I am suspecting a timing problem because IRDY is asserted from the master initiating this configuration write and it de-asserts only 2NS after CLK33 rises.

PCI_IO_MUX

pci_io_mux_ad_load_crit.v
(one for each byte lane)

TRDY

load_on_transfer_in

load_in

load_out

POTENTIAL PROBLEM #2:
Since IRDY & TRDY are used in combinatorial logic sometimes, the fact that they are both asserted for only a few NS may cause the logic to work incorrectly.

PCI_IN_REG

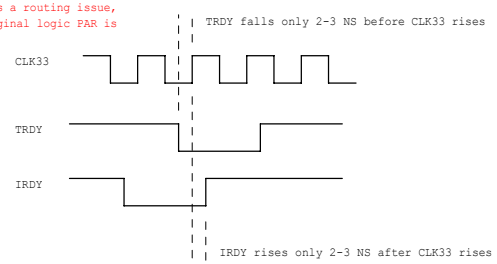
pci_irdy_in

pci_irdy_reg_out

clk

en

POTENTIAL PROBLEM #3:
Since IRDY is de-asserted a couple of NS after clk rises, the combinatorial logic in one of the connected blocks may not see IRDY before the clk signal internal to the FPGA as a result of PAR rises (this is a routing issue, but it is caused by the original logic PAR is attempting to route).



SUPPOSITION:
Perhaps both IRDY & TRDY should only be used as registered variables so they exist for one complete clock cycle for the combinatorial logic. Unfortunately, I am not familiar enough yet with the logic, nor am I sure I am right in the first place to make these changes.

Title: **IRDY/TRDY Relationships**		
Size: B	Document Number: Charles Krinke cfk@pacbell.net	Rev: A
Date: Friday, July 05, 2002	Sheet 1 of 1	