

paltype pal1618  
palname requestor U501  
palid 1.2 84/08/30

PALBEGIN

% Inputs

1 INPUT p1.br0-  
2 INPUT p1.br1-  
3 INPUT p1.br2-  
4 INPUT p1.br3-  
5 INPUT dmareq  
6 INPUT dmareq.s  
7 INPUT as  
8 INPUT p1.bgin-  
9 INPUT eventvalid  
11 INPUT sysreset  
17 INPUT dtack  
16 INPUT berr

% outputs

19 OUTPUT event  
18 OUTPUT dmagrx  
15 OUTPUT dmagr-  
14 OUTPUT bbsy  
13 OUTPUT brout  
12 OUTPUT bgout-

10 GND

20 VCC

EQUATIONS

% Event means that the requestor has to make a decision.

% Three things can cause an event:

%

% 1) If we aren't asserting bbsy, meaning we don't have control of the  
% bus, only p1.bgin- is an event.

% 2) If we are asserting bbsy (we have the bus), dmareq is an event  
% signifying that we want to use the bus.

% 3) If we are asserting bbsy (we have the bus), any p1.br<sub>x</sub>- is an event  
% signifying that someone else wants to use the bus.

%

% For any event, there are two possible outcomes:

% Either we get the bus, or we don't get it.

%

% An event clocks the external dmareq synchronizer flip flop, which  
% decides whether or not dmareq was asserted. The event signal is also  
% fed through an external delay line and back into the pal as eventvalid.  
% The delay provides settling time for the synchronizer.

%

% After the delay, the pal makes the decision whether to take or  
% release the bus.

%

% Event has to be deasserted after the bus is acquired, in case an  
% external request comes in during our cycle. This guarantees that the second  
% event will eventually cause a rising edge on the event line, and it will  
% not be ignored.

% event = bbsy \* / dmagr \* ( dmareq + p1.br0 + p1.br1 + p1.br2 + p1.br3 )  
% + / bbsy \* p1.bgin

ASSERT event \* CONDITIONS to turn OFF event (De-Morganized)  
ENABLE ALWAYS

OR / bbsy / pl.bgin % idle state - we don't own the bus, no bus grant  
OR bbsy dmagr % After we've granted ourself the bus, turn off event

OR / pl.bgin dmagr % Probably redundant - if we've granted ourself the  
% bus, bbsy must be true, so pl.bgin must be false

OR bbsy / dmareq / pl.br0 / pl.br1 / pl.br2 / pl.br3  
% no event if we own the bus but nobody wants it

OR / pl.bgin / dmareq / pl.br0 / pl.br1 / pl.br2 / pl.br3  
% no event if nobody wants the bus and no grant

% bbsy = / sysreset \*  
% (  
% ( eventvalid \* dmareq.s )  
% + ( bbsy \* ( / eventvalid + dmareq.s ) )  
% )

ASSERT bbsy ( De-Morganized )  
ENABLE ALWAYS

% turn off on system reset  
OR sysreset

% stay off until a bus grant has clocked-in our request  
OR / dmareq.s / bbsy

% and until a flip-flop settling delay after the bus grant  
OR / eventvalid . / bbsy

~~OR / eventvalid . / bbsy~~  
% once on, don't turn off until the synchronized dmareq has  
% gone away and it has been validated

OR / dmareq.s & eventvalid & / pl.bgin

% br = / sysreset \* dmareq \* / bbsy

ASSERT brout % De-Morganized !  
ENABLE ALWAYS

OR sysreset % turn off on system reset  
OR / dmareq % no bus req. if we don't need the bus  
OR bbsy % no bus req. if we already have the bus  
% or after we have captured it

ASSERT dmagr-  
ENABLE ALWAYS

% on when our request has been granted and flip-flop has settled  
% and everybody is off the bus  
OR dmareq dmareq.s eventvalid / sysreset / as / dtack / berr  
% stay on until request goes away  
OR dmareq dmagr / sysreset

ASSERT dmagrx ( just an inverted version of dmagr- )  
ENABLE ALWAYS  
OR dmagr- ( dmagrx is low when dmagr- is high )

*Can demorganize above for speed up of 1 pal delay.*

ASSERT bgout- % daisy-chained bus grant out  
ENABLE ALWAYS

% pass grant if we don't want the bus, but first wait for f/f to settle  
OR pl.bgin eventvalid / dmareq.s

PALEND

bye

TIMING: NO-CLOCK

pl.br0- -----  
pl.br1- -----  
pl.br2- -----  
pl.br3- -----  
dmareq -----  
( pl.bgin- ----- )  
eventvalid -----  
dmareq.s -----  
% br -----

bgout- -----  
bbsy -----  
dmagr- -----

sysreset

PALEND