# Change Summary: PFET ACK to follow PFET ENABLE instead of getting tied off to 1.

# Steps to drive ack\_port in non-UPF (non-power aware) simulations

Following is a list of recommendations and steps that can be followed to drive ack\_port during non-UPF simulations.

pfet\_en comes from PMC and ack to PMC is generated by the IP. [Stitching requirements is out of scope of this document].

## Testbench Changes

1. Use a global macro to enable driving of the ack port.   
   Macro name to use: **ENABLE\_NONUPF\_ACK**
2. ack\_port needs to be driven such that it follows the control port as power BFM expects an ack for proper handshake. This is also needed for proper sequencing.
3. ack\_port is driven to follow control port when this macros is enabled.

NOTE: Do not place the ack ports in test island as they will get prompted to the full chip. Full chip handling of ack\_port is different.

`ifdef ENABLE\_NONUPF\_ACK

assign <BFM path to PMC ack port>/ack\_port = <BFM path to PMC control port>/pfetenb; //loopback

`else

assign <BFM path to PMC ack port>/ack\_port = <RTL path to IP ack port>/ip\_pmc\_pg#\_fet\_en\_ack\_b;

`endif

## Environment Changes

Two VCS executables need to be created. One with UPF and the other without UPF. The model without UPF will have *+define+ENABLE\_NONUPF\_ACK*.

Some of the ace changes that would be needed are shown below. The total ace changes shown below are not comprehensive but provides a guideline on what needs to be done.

my $PROJECT = "myip\_lib";  
%audf = (  
 HDLSpec => {  
 $PROJECT => {  
 export => {  
 libs => ['myip\_rtl\_lib', 'myip\_upf\_lib' ],  
 },  
 libs => {  
 myip\_rtl\_lib => {  
 -hdl\_spec => [ "ace/myip.hdl", ],  
 },  
 myip\_upf\_lib => {  
 -hdl\_spec => [ "ace/myip.hdl", ],  
 -vlog\_opts => [ "+define+ENABLE\_NONUPF\_ACK", ],  
 },  
 }, # libs  
 }, # $PROJECT  
 }, # HDLSpec  
); # %audf

Each of the libs shown can be used in an ace “model”.

models => {  
 myip\_upf=> {  
 top => "<path>",  
 power\_intent => {  
 power\_top => 'pch\_tb/pch',  
 upf\_files => ['tools/collage/work/upf.DNT/outputs/pch.upf'],  
 },

<….. rest of the code>  
 libs => [ “myip\_upf\_lib” ],

….<rest of code>

}  
 myip\_rtl=> {  
 top => "<path>",  
<….. rest of the code>  
 libs => [ “myip\_rtl\_lib” ],

….<rest of code>

}

}

“myip\_upf” and “myip\_rtl” are 2 ace models each built with UPF (myip\_upf\_lib) and non-UPF (myip\_rtl\_lib) respectively.

Ace compiles and elaborates “myip\_upf” and “myip\_rtl” into 2 simvs.  
  
Specific simv can be picked to run simulations using ***ace –x –t <> -m <ace model name>***

Created by Satya Ayyagari for SPT; Last Update: 11/07/2012  
Acknowledgements: Sudheendra Kadri, Alamelu Ramaswamy, Rashid Nazir, Azam Asad