Eligible voters		Closed 12-Jan-2006								
Last Name	**First Name**	**2028**	**2043**	**2050**	**2062**	**2064**	**2065**	**2068**	**2069**	**2071**
Ashenden	Peter	1	1	1	1	1	1	1	1	1
Aynsley	John	1 [JA1]	1	1	1	1	1 [JA2]	1	1	1
Bailey	Stephen	1 [SB1]	1	1	1	1	1	1	1	1
Lewis	Jim	1	1	1	1	1	1	1	1	1
Molenkamp	Bert	1	1	1	1	1	1	1	1	1
Myers	Robert	1	1	1	1	1	1	0 [RM1]	1	1
Ries	John	1	1	1 [JR1]	0 [JR2]	1	A [JR3]	1	1	1
Shankar	Sukrit	A	1	1	A	A	1	1	A	1
Shields	John	1 [JS1]	1	1	1	1	1	1	1	1
Swart	Chuck	1	1	1	1	1	1	1	1	1
Varikat	Ajayharsh	1	1	1	1	1	1	1	1	1

[JA1] I approve of the substance of the proposed change, but the wording needs some more thought. In particular, I agree with Steve Bailey that the statement "If a force or deposit was scheduled for any signal, the force or deposit is no longer scheduled for the signal." needs to be clarified.

[JA2] I agree with the originator of this issue that it is "a really annoying limitation" and regret that the VASC-ISAC were unable to find at least a partial solution.

[SB1] After reviewing the draft LRM, I now understand that it only allows one given force or deposit for a signal or driver to be pending at any given time. Also, the signal update phase accounts for the force and deposit. So, the unscheduling of forces and deposits does make sense.

I also note that there is a separate function for scheduling transactions on a driver which is used for future activity via VHPI.

So, if the IR is updated to include appropriate wording to capture the dependency on the current state of the D2 LRM, then my vote is changed to positive.

[RM1] Confusion may be in not that the space is allowed, but whether or not there is consistency in tool suites sometimes "tossing"/deleting it during parsing. Example: some tools may process "name . element" as equivalent to "name. element" or "name.element", while others may not consider these three to be equivalent. Further clarification is needed, in my opinion.

[JR1] The proposal is better than what is currently in the LRM, but its not clear what the value of S'Last_value is for composite signals. The 87 version, defines S'Last_value is the S'Last_value of each scalar element. The 93 and the proposed definition appear to be saying S'Last_value of a composite signal is the value the signal at the most recent time that a scalar element changed.
For simple example

```
type r is record
  f1: integer;
  f2: integer;
end record;

signal s : r := (0,0);
...
process
begin
   s.f1 <= 1 after 1 ns;
   s.f2 <= 1 after 10 ns;
   wait for 11 ns;
   -- what is s'last_value here.  In 87 it would be (0, 0),  in 93 I believe that it would be (1, 0) because
  -- the last event on s was at time 10, and at that time s.f1 had the value of 1.
```

The 93 defintiion is actually more constent with other attributes like 'last_event.

[JR2] This fixes the problem for generics because they are globally static primary, but it misses the case of signal ports, they are not globally static primaries but their type must be computable at elaboration time also just like the generic's.

[JR3] I'm not sure if we are voting here to change the others rules or keep them as is.

[JS1] My comment is with respect to release semantics, which are also mentioned here. Since they are mentioned here, I want to point out that we discussed release semantics of force for VHDL as a result of the analysis of fast track item FT-07 proposing a signal_release that should have the same semantics as VHPI. There was a possible conflict in the fast track desire vs. the VHPI semantics. The issue was whether release should simply "let go" of the force state of the net and let normal signal propagation update it the next time there is activity on the net or whether release should trigger an update of the net based the current state of the drivers of the net. VHPI currently is defined to "let go" of the force state.
The resolution in VHPI was that we would not change the semantics now(even though we agree that an enhancement to vhpi's release mechanism to support the other semantic is needed) for the sake of stability and because we expected to be done and in ballot before the FT work completes. One should assume there will be another release mode(e.g., a vphiReleasePropagate) in the future (for signals, not for drivers).