IEEE Standards Interpretation for IEEE Std 1685™-2009 IEEE Standard for IP-XACT, Standard Structure for Packaging, Integrating, and Reusing IP within Tools Flows

This is an interpretation of IEEE Std 1685-2009.

November 2011

Interpretation Request #1
Topic: Interaction between volatile and modifiedWriteValue elements **Clause:** 6.10.9.2 A and 6.10.9.2 D

In IEEE Std 1685-2009, 6.10.9.2.d, The modifiedWriteValue element allows IP-XACT to describe fields that have unconventional write behavior such as oneToClear, or zeroToToggle. Any field flagged with a modifiedWriteValue when read after writing is not guaranteed to return the value written.

It could therefore be assumed that any field with a modifiedWriteValue should also have a volatile element with a true value, however in the second sentence of IEEE Std 1685-2009, 6.10.9.2.a, it states that "The [volatile] element implies there is some additional mechanism by which these registers can acquire new values other than reads/writes/resets **and other access methods known to IP-XACT**." The oneToClear write mechanism is known to IP-XACT, so that would imply volatile should be false or omitted on such fields unless there was some additional capability for the field to change value (such as in response to a counter overflow).

Should the second sentence of IEEE Std 1685-2009, 6.10.9.2.a be interpreted as illustrative and therefore fields with a modifiedWriteValue should always have a true volatile element? Or should the second sentence of IEEE Std 1685-2009, 6.10.9.2.a be interpreted to provide an exception to the behavior described in the first sentence in cases where the behavior of the field is fully described in IP-XACT?

Interpretation Response #1
1 Volatile and modifiedWriteValue Elements in IEEE Std. 1685-2009

1.1 Request for Interpretation
The request for interpretation focused on IEEE Std 1685-2009, 6.10.9.2A and 6.10.9.2D.

1.1.1 IEEE Std 1685-2009, 6.10.9.2A
"volatile (optional) when true indicates in the case of a write followed by read, or in the case of two consecutive reads, there is no guarantee as to what is returned by the read on the second transaction or that this return value is consistent with the write or read of the first transaction. The element implies there is some additional mechanism by which this field can acquire new values other than by reads/writes/resets and other access methods known to IP-XACT. If this element is not present, it is presumed to be false. The volatile element is of type boolean."

1.1.2 IEEE Std 1685-2009, 6.10.9.2D
"modifiedWriteValue (optional) element to describe the manipulation of data written to a field. The value shall be one of oneToClear, oneToSet, oneToToggle, zeroToClear, zeroToSet , zeroToToggle, clear, set, or modify. If the modifiedWriteValue element is not specified, the value written to the field is the value stored in the field."

1.2 Possible Interpretations
There are two possible interpretations.

First Interpretation:
IEEE Std 1685-2009, 6.10.9.2A could be interpreted as stating a field is volatile in IP-XACT if the field does not follow normal ram like behaviour. See the excerpt below from IEEE Std 1685-2009, 6.10.9.2A:

"a) **volatile** (optional) when **true** indicates in the case of a write followed byread, or in the case of two consecutive reads, there is no guarantee as to what is returned by the read on the second transaction or that this return value is consistent with the write or read of the first transaction...".

This interpretation of volatile would be the same as ANSI C. It is consistent with a software developer's use of the volatile keyword in the C programming language (a keyword to direct the compiler not to apply optimizations that assume that the next value read will match the last value read or written). It would imply any field which uses a modifiedWriteValue (e.g. oneToSet) must have volatile set to true.

Second Interpretation:
Alternatively, one could interpret IEEE Std 1685-2009, 6.10.9.2A and 6.10.9.2D as stating a field is volatile if the field's value is different from one access to the next and if the difference is caused by an access mechanism not known to IP-XACT. This is supported by

the following statement in IEEE Std 1685-2009, 6.10.9.2A:

"The element implies there is some additional mechanism by which this field can acquire new values other than by reads/writes/resets and other access methods known to IP-XACT."

With this interpretation the meaning of volatile in IP-XACT is different to in ANSI C. This means a field with a modifiedWriteValue is not volatile, unless there is some other mechanism unknown to IP-XACT which can alter the fields value.

1.3 Resolution

The second interpretation above applies to the IP-XACT 1685 standard. The same rules also apply to readAction.