

Minutes from the SV-BC Meeting of August 19, 2002 - Cliff Cummings

The following information is for use in connecting to the list committee meetings (all times are West Coast):

19 August 9:00am-11:00am SV-BC
19 August 11:00am-1:00pm SV-EC

Toll Free Dial In Number: (877)233-7845
International Access/Caller Paid Dial In Number: (505)766-5458
PARTICIPANT CODE: 516134

SV-BC Minutes from the August 5th meeting were not available and were not approved. Cliff to compile and send.

Next SystemVerilog-BC meeting: September 4th - Face-to-face meeting at Synopsys
Next IEEE Verilog Standards Group Meeting: August 26 - starts at 8:30 AM PDT - contact Mike McNamara for details.

Others -

16 September ****CANCELLED**** (Cliff will be unavailable to lead the call)
30 September
14, 28 October
11, 25 November
9, 16 December
6, 20 January
3, 17 February

NOTE: all emails will now only be sent to the sv-bc@eda.org reflector.

This is my list of attendees and voting status - please submit corrections:

(aaaaa)	Cliff Cummings	(Sunburst Design)	*
(aaaaa)	David Smith	(Synopsys)	*
(aa--a)	Heath Chambers	(HMC)	
(aaaaa)	Karen Pieper	(Synopsys)	*
(aaa-a)	Kevin Cameron	(NSC)	*
(aa---)	Medi Mohtashemi	(Synopsys)	
(aa---)	Paul Graham	(Cadence)	
(aaaaa)	Peter Flake	(Co-Design)	*
(aaaa-)	Simon Davidmann	(Co-Design)	*
(aaaaa)	Stefen Boyd	(Boyd Technology)	*
(aaa-a)	Steven Sharp	(Cadence)	*
(-aaa-)	Dave Kelf	(Co-Design)	*
(-aaa-)	Dennis Brophy	(Model Technology)	*
(-aa--)	Mike McNamara	(Verisity)	
(-aaaa)	Tom Fitzpatrick	(Co-Design)	*
(--aaa)	Vasisilios Gerousis	(Seimens)	*
(---aa)	Francoise Martinolle	(Cadence)	
(----a)	Don Mills	(LCDM Engineering)	
(----a)	Gord Vreugdenhil	(Synopsys)	

* indicates eligible to vote on consensus issues

Discontinue tracking the following engineers

(a----	Rog Armoni	(Intel)	
(-a----	Kurt Takara	(Zero-In) sv-bc reflector??	

2.3 Integer and logic literals

Literal integer and logic values can be sized or unsized, and follow the same rules for signedness, truncation and left-extending as Verilog-2001.

SystemVerilog adds the ability to specify unsized literal single bit values with a preceding apostrophe ('), but without the base specifier. All bits of the unsized value are set to the value of the specified bit, and the value is treated as unsigned.

```
'0, '1, 'X, 'x, 'Z, 'z // sets all bits to this value
```

Proposal, add the wording as shown.

Proposed: Tom Fitzpatrick

Second: Steve Sharp

Passed unanimously.

Hi All,

In Systemsim, the answer is:

test literal is unsigned which is the same as if the "0" were replaced with "b0", which is defined to be unsigned.

Thanks for the example Steve. Very clever.

-t

> -----Original Message-----

> From: owner-sv-bc@server.eda.org [<mailto:owner-sv-bc@server.eda.org>]On

> Behalf Of Steven Sharp

> Sent: Wednesday, July 24, 2002 4:03 PM

> To: sv-bc@server.eda.org

> Subject: Test program for literal signedness

```
module top;
  reg [1:0] rs, ru, rt;

  initial begin
    rs = 1'sb1 | 1'sb0;
    ru = 1'sb1 | 1'b0;
    rt = 1'sb1 | '0;
    if (rs !== 2'b11 || ru !== 2'b01)
      $display("signed arithmetic broken!");
    if (rt === rs) $display("test literal is signed");
    else if (rt === ru) $display("test literal is unsigned");
    else
      $display("test literal is broken");
  end
endmodule
```

=====

From the minutes of July 22nd, we still need proposed wording to be most likely added to the end of section 3.7

AI - Steve Sharp to propose document wording. Steve to write example and Tom to test it.

BC7d -

Since data types can be passed into modules or interfaces, it may not be possible to distinguish an array/structure literal from an ordinary concatenation until elaboration time. This probably doesn't place too much of a burden on the implementation. It does either delay error checking until elaboration, or require duplication of error checking so that errors can be produced as early as possible.

- Just an expression of concern by Steve Sharp - no action proposed.

There exists an ambiguity between structure literal and concatenation when passing as an argument where the formal is a union of a structure and a bit field - the known resolution - if it is a packed union, any assignment will always be treated as a bit-vector (no matter what the contents are) - if unpacked union then it will be treated as the first union member declared in the declaration. - Clarification may need to go into union section.

=====

- > Section 3 - Issues with new data types and keywords
- > (Steve/Cadence) [Basic]
- > - actual utility of char, shortint, longint, byte, shortreal

While it's true that there is no interface to C defined in SystemVerilog, these datatypes were added to SUPERLOG because it does have an interface to C. Before we decide to remove or alter these datatypes and keywords, we should wait and see what the SV-CC does in this area.

These data types were already added to the SystemVerilog 3.0 document. Consensus opinion of the BC is that the SV-CC committee should look into these data types for augmented descriptions or no change.

AI - Cliff to send this issue to the SV-CC committee. Committee members on both the SV-BC and the SV-CC committee should convey discussions of reserved-word types vs implementation using typedefs. Please contact Steve Sharp for his list of concerns.

=====

A review of an action item from the last conference call:

- > Section 3.6 - Implications of Enum type I/O (Steve/Cadence) [Basic]
- > - need to detail what is expected of Verilog I/O routines to support this
- > - [also vcd enhancements]

> In section 3.6, it is stated that enumerated types can be displayed using the enumerated names. Is this supposed to imply some capability of the Verilog I/O routines? If so, these capabilities need to be defined. If it is just a speculation about what the user interface to some tool might allow during debugging, then this should be made plain.

When printed in string format ("%s"), the enum value identifier shall be printed as a string. I do not believe that this implies any additional requirements on I/O routines.

AI - Franciose to propose added wording to section 3.6

AI - Cliff to put together new examples for section 3.6: \$display / %s / and integer data types

Look into example enum tags within the same value? Not legal (per Tom & Peter) - may need to be specified - per Kevin, C allows two names with the same value.

=====

Section 2 - (discussed at the July 22nd SystemVerilog-BC Committee meeting)

=====
Section 3 - Data packing issue (Kevin/NSC) [Basic]

- it is impossible to implement "union" from the current LRM description
 - there are many ways to do it which are not compatible
 - encoding of logic types is a factor, and "big-endian" vs. "little-endian"
 - unions should have either all logic or all bit as the base type of all elements
 - if packing is defined then 'packed' union syntax is redundant
 - may be desirable to state the packing/alignment explicitly for software compatibility
- [See email on the reflector archive for details from Kevin.]

From the SystemVerilog 3.0 Document Section 3.7, top of page 11.

Like a packed array, a packed structure can be used as a whole with arithmetic and logical operators. The first member specified is the most significant- and subsequent members follow in decreasing significance. The structures are declared using the packed keyword, which can be followed by the signed or unsigned keywords, according to the desired arithmetic behavior, which defaults to unsigned:

Proposal, add the wording as shown.
Proposed: Gord Vreugdenhil
Second: Steve Sharp
Passed unanimously.

From the SystemVerilog 3.0 Document Section 3.7, bottom of page 11.

A packed union contains members that are packed structures or arrays of the same size. This ensures that you can read back a union member that was written as another member. If any member is 4-state, the whole union is 4-state. A packed union can also be used as a whole with arithmetic and logical operators, and its behavior is determined by the `signed` or `unsigned` keyword, the latter being the default.

AI - Peter - to reword for clarity the sentence, "If any member is 4-state, the whole union is 4-state." (define read and write semantics)

Shall we permit packed unions with members of different sizes?
Opposed: Tom, Peter, David
Abstain: Stafen, Karen, Steve Sharp, Cliff
Favor: Kevin
Motion fails.

=====
Section 3 - Type use before definition (Steve, Paul/Cadence) [Basic]

- forces type checking to be post-elaboration
 - cause unnecessary complication of analysis, particularly separate analysis
 - useful only with pointer types
- [This has been discussed before - no additional detail is involved.]

AI - Peter - to contact Co-Design Aps Engineer for better reasoning/support for not requiring declaration of typedefs before use. Co-Design Aps Engineer to email the sv-bc reflector.

=====
Meeting adjourned at 10:55 AM - Cliff Cummings