

[Minutes from the meeting of July 22, 2002 - by Cliff Cummings](#)

Conference call information - courtesy of David Smith & Synopsys:

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

5 August 9:00am-11:00am SV-BC

5 August 11:00am-1:00pm SV-EC

PARTICIPANT CODE: 516134

Toll Free Dial In Number: (877)233-7845

International Access/Caller Paid Dial In Number: (505)766-5458

I have been out of town the past two weeks and four of the last five weeks, but I am mostly in town for the month of August.

I hope to get some minutes from our last meeting out to you just before the conference call starts at 9:00 AM PDT.

Regards - Cliff

Next meeting: August 19 - 9-11 AM PDT

Next IEEE Verilog Standards Group Meeting: August 12 - starts at 8:30 AM PDT - contact Mike McNamara for details.

Others -

16, 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.](#)

SV-BC Agenda:

Approve minutes sent by Stefen Boyd

(email: Date: Mon, 08 Jul 2002 11:58:07 -0700 / Subject: SV-BC: Minutes from July 8, 2002 meeting)

[Motion to approve: Dave Kelf](#)

[Second: David Smith](#)

[Passed unanimous](#)

Access to the SystemVerilog 3.0 Document - Does everybody have access? [- Yes](#)

Does anybody know when the next HDL++ (or whatever it is called) meeting is scheduled? [-](#)

[Vassilios - HDL+ - trying to form a SystemVerilog Committee](#)

[SystemVerilog 3.1 will come from the sv-ec committee \(combined with sv-bc\)](#)

[Which documents will be passed to the IEEE Verilog Working Group?](#)

How are SV-BC proposals supposed to be presented for HDL++ group approval?

SV-BC to write proposals and HDL++ committee to approve and Stu to incorporate?

Documentation is still an issue - Vassilios will work with committee chairs to resolve.

Amend voting rules? - Grandfather clause for SystemVerilog eligible voters??

Current rules

Proposal - anyone who attend any of the first four meetings will have voting rights and then the voting rules kick in - David

Second: Vassilios

Passed unanimously

Voting structure and rules

3/4 meetings must be attended or 75%

Vote on technical issues will be a simple majority of all attendees

(not limited to one per company)

SV-BC Item List

+ To be discussed at 7/22 meeting (BC5, BC7, BC8, BC11, BC17**new**)

SV-BC1 -Deprecation - **Complete** (no new proposals)

SV-BC2 -Time precision and timescale - AI - Peter - Discussion at 07/22 meeting?

SV-BC3 -Dynamic process control - Moved to SV-EC (David - has the EC accepted this?)

SV-BC4 -DSM (negative timing check) - AI - Dennis Brophy and Steven Sharp to make proposals.

+SV-BC5 -Data alignment and data packing issues - AI - Peter to make a proposal & Kevin to re-send proposal

+ From: Peter Flake <flake@co-design.com>

+ Date: Fri, 19 Jul 2002 09:54:32 +0100

+ Subject: Size of members of a packed union

SystemVerilog 3.0 - section 3.7 - 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. 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 by Tom Fitzpatrick

Second: Steve Sharp

Pass - unanimous

A packed union shall contain members that ~~are~~packed structures, ~~or~~packed arrays or integer data types 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.

Intention: to affirm that the bit order is specified (top of page 11 - first member sentence)

There may be an issue related to interfacing to C compilers - SV-CC should examine what we have done.

SV-EC: To address whether real numbers should be addressed by packed arrays.

AI - Kevin - may make a clarification proposal concerning alignment of bytes on integers.

SV-BC6 -Clarify auto increment/decrement AI - Peter to make a proposal about areas that were gray-areas in C

SV-BC7 -Cadence issues w/ Section 2 literals - AI - Steve Sharp to send list

+ From: Stefen Boyd <stefen@boyd.com>

+ Date: Mon, 08 Jul 2002 11:58:07 -0700

+ Subject: SV-BC: Minutes from July 8, 2002 meeting

BC7a - Section 2.3

In a self-determined context (such as $\$display("%b", '1)$ what is printed)

Proposed: Steve Sharp

Second: Cliff Cummings

Passed unanimous

Proposal - add to the last paragraph of section 2.3 the sentence.

In a self-determined context these literals have a width of 1 bit.

BC7b - Section 2.3

AI - Steve Sharp to send testcase to Superlog guys to test signed arithmetic - does everything else become unsigned?

BC7c - Section 2.7 & 2.8 & 7.9

Can strings be used?

How precisely must the types match?

AI - Peter to examine what Superlog does and report back

BC7c - Section 2.7 & 2.8

Where exactly are array and structure literals legal?

They are apparently only supposed to be legal on initializers. - THIS STATEMENT WAS INCORRECT

Proposed: Tom Fitzpatrick

Seconded: Steve Sharp

Passed - unanimous

Section 2.7

Arrays literals are syntactically similar to C initializers, but with the replicate operator ($\{\{\}\}$) allowed.

Section 2.8

Structure literals are syntactically similar to C initializers. Structure literals must have a type, either from context or a cast.

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.

SV-BC8 -Cadence issues w/ Section 3 - AI - Steve Sharp to send list

+ From: Stefen Boyd <stefen@boyd.com>

+ Date: Mon, 08 Jul 2002 11:58:07 -0700

+ Subject: SV-BC: Minutes from July 8, 2002 meeting

+ From: "Tom Fitzpatrick" <fitz@co-design.com>

+ Date: Fri, 19 Jul 2002 10:52:43 -0400

+ Subject: Fitz's Action Items for 7/22 Meeting

SV-BC9 -Section 3.1, parameterized data types - ????

SV-BC10 -Displaying enumerated types, affect on VCD - Nobody assigned yet
SV-BC11 -Section 4: are elements of a signed packed array signed? - AI - Steve Sharp (???)
+ From: Stefen Boyd <stefen@boyd.com>
+ Date: Mon, 08 Jul 2002 11:58:07 -0700
+ Subject: SV-BC: Minutes from July 8, 2002 meeting

SV-BC12 -Constant expressions - Diff between parameters, localparams, constants? - AI - Paul Graham (???)
SV-BC13 -Change BNF to simplify attributes--for 1364 committee? - Cliff to send to IEEE committee
SV-BC14 -Section 9: process execution efficiency - AI - maybe Erich Marshner? and Kevin
SV-BC15 -Interleaving, event scheduling - clarify? Atomic operation - Nobody assigned yet
SV-BC16 -interfaces - Interface enhancements/simplifications - Nobody assigned yet

*** NEW ***

SV-BC17 -Other sections reviewed by Cadence
+ From: Stefen Boyd <stefen@boyd.com>
+ Date: Mon, 08 Jul 2002 11:58:07 -0700
+ Subject: SV-BC: Minutes from July 8, 2002 meeting

+ From: Steven Sharp <sharp@cadence.com>
+ Date: Mon, 8 Jul 2002 14:14:16 -0400 (EDT)
+ Subject: Further issues (mostly for SV-BC)

More Complete Descriptions Follow:

SV-BC1 -Deprecation follow on - do committee members have
* waiting for anyone to propose additional deprecation items

SV-BC2 -Time precision and timescale in general - cleaning up the description
* AI-Peter - will propose new wording for the 07/22 meeting

SV-BC3 -Dynamic process control - (Kevin) processes can be created but there is no mechanism to kill or suspend them
disable questions
Kevin, Steve
* AI-David - Moved to the EC committee

SV-BC4 -DSM - Deep Sub-Micron (negative timing check)
Negative timing check descriptions need improvement - Dennis Brophy
* AI - Dennis Brophy and Steven Sharp will talk to companies for proposals
This is on the Basic list because it is also part of the Verilog-2001 (shoring up language capability)
After proposals are made, a decision will be made concerning which committee has responsibility.

SV-BC5 -Data alignment and data packing issues - related to pointers C/C++ interfacing - cast from one type to another or pass into external system (binary write)
Kevin, (maybe coordinate with C-committee)
* AI - Peter to make a proposal and then to discuss it.
* AI - Kevin to re-send his earlier proposal

SV-BC6 -Clarify auto increment/decrement - (Karen) do you clarify when the increment happens - what happens with multiple per line
Karen
* (per Peter) has been implemented left-to-right
* AI - Peter to make a proposal about areas that were gray-areas in C

SV-BC7 -Cadence issues w/ Section 2 literals
Steve Sharp

SV-BC8 -Cadence issues w/ Section 3
Steve Sharp

SV-BC9 -Section 3.1, parameterized data types - ???

SV-BC10 -Displaying enumerated types, affect on VCD - VCD overhaul to include automatic variables, interfaces, structs, etc.

SV-BC11 -Section 4: are elements of a signed packed array signed?
Steve Sharp

SV-BC12 -Constant expressions - Lack of clarity about difference between parameters, localparams, constants (latter initialized at simulation time)
Paul Graham

SV-BC13 -Change BNF to simplify attributes--for 1364 committee?

SV-BC14 -Section 9: process execution efficiency - maybe Erich Marshner?
Kevin

SV-BC15 -Interleaving, event scheduling - clarify? Atomic operation

SV-BC16 -interfaces - Interface enhancements/simplifications - add more explanation about distinction between interface and module