Changes

- Resolutions
  - CH-60 – (SSWG) CH-113
  - CH-84 – (SSWG) CH-113
  - CH-98 – (SSWG) CH-113
  - CH-102 – (Handle -> CHandle) CH-113
  - CH-101 – (SSWG) CH-113
  - CH-96 – (clarification/syntax change) CH-113
  - CH-97 – (clarification) CH-113
  - CH-99 – (clarification, SSWG) CH-113
  - CH-100 –CH-113
  - CH-93 – CH-113
  - CH-94 – (Refer to 8.9 from 13.12, add statement to 8.9)
- Vote
  - CH-47 – vote
- New
  - CH-111 – Fixes to draft 3 for AI-64, AI-65, AI-71, AI-72
  - CH-113 – Rework changes to respond to above issues
  - CH-114 – Resolve editors notes
  - CH-115 – AI-67 (Class for linked list)
- Unresolved
  - CH-17 – need event resolution

CH-60

1 for, 6 against

- Section 13.13 Input sampling
- Comments:

Jay	Definition or existance of "Verification phase" is TBD.
Francoise	Replaces "If the input skew is zero then the value sampled corresponds to the signal value at the start of the verification phase." with: "If the input skew is zero then the value sampled corresponds to the signal value at the clock domain event"
Stefen	We still don't know what the verification phase is yet, but assuming it resembles what we've seen, this would mean that zero skew would capture the DUT outputs *after* NBAs have propagated. This would make sense if it was at the start of the design phase. Why does this matter? Your testbench sampling with zero delay won't work for both zero delay rtl and gate level sims! Any clk->q delay on flops in design will mean sampling before vs after clock edge in gate vs rtl versions of dut. If we sampled at beginning of design phase, we're ok.
Neil	What happens when we sample a signal within an expression? Do we wait for the clocking event? What if there are signals from different clockings in the same expression where each signal uses a different clocking event? I assume that the samples in these situations take place immediately without waiting for the clocking event.
Mehdi	[more clarification on sampling would help]
Arturo	Verification phase: Needs to reflect new names from SSWG
Brad	SSWG

- Resolution: CH-113

CH-84

2 for, 5 against

- Section 13.3
- Comments:

Jay	Definition or existence of "Verification phase" is TBD.
Francoise	Leave as it was previously said.
Stefen	same reason as CH-60
Arturo	Verification phase: Needs to reflect new names from SSWG
Brad	SSWG

- Resolution: CH113

CH-98

2 for, 5 against

- Section 13.14.1
- Comments:

Jay	Definition or existance of "Verification phase" is TBD.
Francoise	If the the conflicting drives are only for net and not for variables, I would insert the additional proposed paragraph before instead of after.
Neil	It looks like I am now confused. Didn't CH-96 do away with the non-blocking drive?
Mehdi	signal operation clarified.
Arturo	Verification phase: Needs to reflect new names from SSWG
Brad	SSWG

- Resolution: CH113

CH-101

6 for, 1 against

- Section 13.14.1, 13.14.2, 13.14.2
- Action: Merge Chapters 13 and 14 from Draft 2
- Comments:

Jay	Definition or existance of "Verification phase" is TBD.
Francoise	Does it mean that we are merging 13 and 14?
Mehdi	the combined chapters 13/14 resolves.

- Resolution: Changes to handle region names made in CH-113

CH-102

2 for, 2 against

- Sections 3.1 and 3.7
- Action: Add Handle
- Comments:

Arturo	Suggest the change: remove the sentence: , which represents a non-existent handle
Jay	1) I don't like this keyword, maybe something like Chandle would avoid more conflicts with existing names and the wording in CH-104 2) Has the sv-cc dealt with all the other layout compatibility issues from SV to 'C'. If not, adding this type is premature.
Francoise	Would like Cpointer name instead of handle. Would like to be able to have a C compatible struct contain a Cpointer type fields, otherwise you cannot construct a C list using the C interface and pass it back to Verilog. We should not allow to have packed structs and packed unions contain Cpointer member fields.

- Handled in CH-113

CH-96

1 for, 6 against

- Section 13.14
- Comments:

Jay	Much clearer but, Why aren't concatentations allowed? This still doesn't say when the drive occurs at that cycle (active event or NBA event). Is there an NBA equivalent to this drive?
Francoise	add bnf for event_count. What about using non blocking drives? Do they disappear?
Stefen	The syntax doesn't look like it allows brackets like the ## cycle operator. There should be a sentence stating so explicitly, and presumably, there are restrictions on the kind of expression allowed? If '32-1' were allowed then how would we deal with 'bus.data = ##2 -1-r;'?
Neil	a. I really don't like the use of [] on the '## [5]' type of controls, but if we are going to require it in some places it should be required everywhere for consistency. That means we need to add it in here. b. For the intra-assignment variation, we must define the specific behavior when a ## 0 is specified. c. How do we allow for driving values onto a clocking signal on both edges of a clock? I would like to see an example along with an explanation. (It looks like this is covered in CH-97). d. Are both forms blocking?
Mehdi	the syntax ##, ##[] is not necessarily the best, but shows intent.
Arturo	Requires more changes due to latest SSWG discussions
Brad	SSWG

- Resolution: CH-113

CH-96

- Section 13.14
- Clocking domain drives are always assigned as NBA's.
- Propose changing the syntax to use the non-blocking assignment:
  - **##[3] a <= b;**
  - **a <= ##[3] b;**

CH-97

4 for, 3 against

- Section 13.14.2
- Text:

  When the same variable is an output from multiple clocking domains, the last drive determines the value of the variable. This allows a single module to model multi-rate devices, such as a DDR memory, using a different clocking domain to model each active edge. Naturally, clock-domain outputs driving a net (i.e., through different ports) cause the net to be driven to its resolved signal value.

- Comments

Jay	I just don't understand this.
Stefen	The last sentence is unclear: "Naturally, clock-domain outputs driving a net (i.e., through different ports) cause the net to be driven to its resolved signal value." It's not clear if the resolved value is from the winning assignment driven onto the net (no driver contention from multiple clocking domain outputs) or that each clocking domain acts like a driver on the net (which is what I think I remember from the verbal explanation).
Neil	I wasn't sure if this should be flagged in CH-99 or CH-97... In the face-to-face meeting there was some discussion about a reg versus a wire with respect to resolution. Was that not true?

- Resolution: CH-113

CH-97

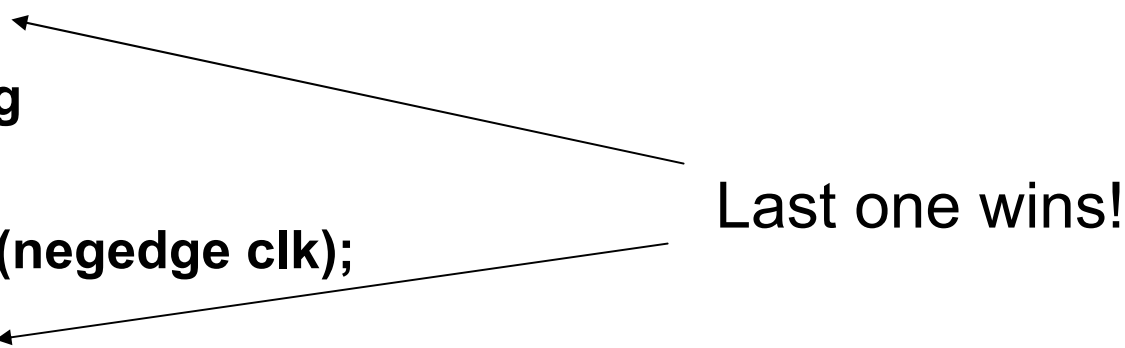
- Section 13.14.2

When the same variable is an output from multiple clocking domains, the last drive determines the value of the variable. This allows a single module to model multi-rate devices, such as a DDR memory, using a different clocking domain to model each active edge. Naturally, clock-domain outputs driving a net (i.e., through different ports) cause the net to be driven to its resolved signal value.

```
reg j;

clocking @(posedge clk);
    output j;
endclocking


clocking @(negedge clk);
    output j;
endclocking
```



CH-99

5 for, 2 against

- Section 13.14.2
- Text:

  When more than one synchronous drive is applied to the same clocking domain **output** (or **inout**) at the same simulation time, the driven values are checked for conflicts. When conflicting drives are detected a runtime error is issued, and each conflicting bit is driven to X (or 0 for a 2-state port).

- Comments:

Jay	How does this differ from CH_97?
Francoise	What is the verification phase? In the example: isn't bus.data = 0 supposed to be bus.data <= 0? I thought we were only allowing 0 delay non blocking drives.
Neil	Need to add 'inout'. Vote changed to yes.
Mehdi	inout addition is ok.

- Resolution: CH-113

CH-100

1 for, 6 against

- Sections 9.9.1, 9.9.2, and 9.9.3
- Resolution: CH-113 (remove $suspend_thread, clean-up 9.9, etc)
- Comments:

Jay	I like "wait fork;" but not exiting simulation when programs are done. I like "disable fork;" but there is still a bunch of text here about $terminate() does it belong?
Francoise	I suggest that we use: wait <block_name> so that if we just name the fork parallel block we can just use that name to wait for all the spawned processes to complete. Also use: disable <block_name> to disable the fork. I don't understand the difference between $terminate and disable. I don't see the need for $suspend_thread if this is equivalent to a #0, just use #0.
Stefen	Typo: "The disable form statement" should be "The disable fork statement" (Fixed). Still lots of references to $terminate in last paragraph of 9.9.2. This needs to be changed to use disable fork. Also shouldn't compare against fork, but fork <label> form of fork. I'm also not sure that I like "disable fork" instead of $terminate() because it's not clear that it would kill all child processes. It's more clear that $terminate would kill my monitor from task setup than 'disable fork'. Using disable is a good idea, though, but perhaps we should use the method notation we've started adopting: disable.child // same as $terminate disable.thread label // same as regular disable but on thread.
Neil	a. Does 'wait fork' only apply to the program block? b. Typos: 1) "function wait_device function" 2) "parentchild" should be parent-child c. I didn't really understand the note about $suspend_thread() versus the use of #0 being called after an NBA.
Mehdi	would like to make sure the equivalent operation $terminate, disable fork
Brad	$terminate --> disable fork

CH-93

6 for, 1 against

- Section 13.12
- Text:

  Explicit synchronization is done via the event control operator, **@**, operator, which allows a process to wait for a particular signal value change, or a clocking event (see section 13.9).

- Comments:

  Jay This entire section needs to be integrated with Events and event control syntax

- Resolution: CH-113

CH-94

4 for, 3 against

- Section 13.12
- Comments:

Jay	Larger issue than I want to give a quick Yes over email vote
Francoise	Should refer to regular event control but not include it here.
Neil	a. According to CH-93 this should now be called "the event control operator". ^^^^^^^^^^^^ b. Clocking-domain 'inout' also allowed? (only input mentioned) c. I would like to get some clarification on this change. Are we now saying that section 8.9 is being enhanced to allow signals contained in clocking-domains to be specified? If so, why don't we just say that?

- Resolution: CH-113 (Change 13.12 and 8.9)

CH-47

5 for, 2 against

- Section 13.3
- Text:
  - A skew must be a constant expression, and can be specified as a parameter. If the skew does not specify a time unit, the current time unit is used. If a number is used, the skew is interpreted using the timescale of the current scope.
  - Remove: When skews are not specified, input signals default to a skew of **1step**, and output signals default to a skew of **#0**.
- Comments:

Francoise		If you look at the bnf for constant_expression A.8.3, a constant expression can be much more that what you allow for skew value. Specifically constant_expression contains string. Constant expression also includes constant_primary and a constant_primary can be a concatenation, a function call, a genvar, a specparam, a parameter etc... I think that the sentence needs to be rewritten to say something like: it must be a constant_expression of type unsigned int or time. What happen if skews are not specified? What are their default values?
Neil	a.	Suggest the following re-wording of the first change. A skew is a constant expression that is optionally followed by a time unit. If a time unit isn't specified, the current time unit is used. A skew can be specfied as a parameter.
	b.	Second change (paragraph removal) – agree

CH-17

3 for, 4 against

- Section 12.6 and 12.7 of LRM
- Comments:

Jay	The persistent nature of events is still up for debate.
Francoise	I don't like the fact that persistent events are declared a different way (event bit) that regular events but the trigger operation is the same for persistent and regular events. There may be other ways of accomplish this without creating a new type of events.
Stefen	As I mentioned in http://www.eda.org/sv-ec/hm/0698.html, http://www.eda.org/sv-ec/hm/0696.html. There are significant problems with the change proposed. Instead of creating a new "event" as the text suggests, it really creates a new "bit" type. Needs to be fixed to really be an event type.
Neil	Flagged issues in this section for 2/10/03 meeting.
Mehdi	event synchronization useful for testbench model