Email Voting Status

Person		Voting	CH-17	CH-18	CH-20	CH_47	CH-53	CH-60	CH-83	CH-84	CH-85	CH-86	CH-93	CH-94	CH-95	CH-96	CH-97	CH-98	CH-99	CH-100	CH-101
Arturo Salz	1																				
Brad Pierce	1																				
Cliff Cummings	1																				
Dennis Brophy	1																				
Francoise Martinolle	1	1	-13	1	1	-13	1	-13	1	-13	1	1	1	-13	1	-13	1	-13	-13	-13	
Jay Lawrence	1	1	-13	1	1	1	1	-13	1	-13	1	1	-13	-13	1	-13	-13	-13	-13	-13	-13
Mehdi Mohtashemi	1																				
Neil Korpusik	1																				
Stefen Boyd	1	1	-13	1	1	1	1	-13	1	-13	1	1	1	1	1	-13	-13	1	1	-13	1
Stu Sutherland	1																				
Totals	10	3	-39	3	3	-11	1	-39	3	-39	3	3	-11	-25	3	-39	-25	-25	-25	-39	-12

CH-17 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.

CH-18 Stefen The bnf requires brackets around expression, but examples 1&2 show without. Need to be consistent. Fixed in CH-54.

CH-47 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?

CH-53 Francoise Minor rewording: replaces statement with declaration. I believe that a default clocking is a declaration and not a statement. Fixed

CH-60 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.

CH-84 Jay Definition or existance of "Verification phase" is TBD.

Francoise Leave as it was previously said.

Stefen same reason as CH-60

CH-93 Jay This entire section needs to be integrated with Events and event control syntax

CH-94 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.

CH-96 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;'?

CH-97 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).

CH-98 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.

CH-99 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.

CH-100 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" 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. For example:

```
task setup;
  fork
    // start some background process of some sort (i.e. monitor)
  join_none
endtask

task foo;
  setup;
  fork
    // couple things started
  join_any
  disable fork; //$terminate();
endtask
```

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.
```

CH-101 Jay Definition or existance of "Verification phase" is TBD.

Francoise Does it mean that we are merging 13 and 14?