

Hi, All -

I have coded the large example previously sent out and the example in the SystemVerilog draft 4. I have also addressed Peter's proposals.

Regards - Cliff

Attached are the following untested (no compiler recognizes the syntax yet) FSM designs (13 files total):

**fsm\_cc8\_diagram.pdf - FSM Diagram for the 10-state cc8 design**

One always block method for doing the cc8 design  
Registered outputs (147 lines each)  
-----

fsm\_cc8new1.v - abstract enumerated types  
fsm\_cc8new1a.v - enumerated types with state definitions

Two always block method for doing the cc8 design  
Combinational outputs (79 lines each)  
-----

fsm\_cc8new2.v - abstract enumerated types  
fsm\_cc8new2a.v - enumerated types with state definitions

Three always block method for doing the cc8 design  
Registered outputs (81 lines each)  
-----

fsm\_cc8new3.v - abstract enumerated types  
fsm\_cc8new3a.v - enumerated types with state definitions

**DRAFT 4 - FIGURE 9-1 -- New SystemVerilog Examples**

One always block method for doing figure 9-1  
Registered outputs (26 & 29 lines)  
-----

fsm1.v - abstract enumerated types  
fsm1a.v - enumerated types with state definitions

Two always block method for doing figure 9-1  
Combinational outputs (25 & 28 lines)  
-----

fsm2.v - abstract enumerated types  
fsm2a.v - enumerated types with state definitions

Three always block method for doing figure 9-1  
Registered outputs (30 & 33 lines)  
-----

fsm3.v - abstract enumerated types  
fsm3a.v - enumerated types with state definitions

At 12:32 PM 3/25/02 +0000, Peter Flake wrote:

>Cliff,

>

>I have looked at your proposals and examples, and I have the following suggestions:

>

>1. Replace the "**state**" keyword and syntax with "**transition enum**" and the **enum** syntax, including optional encodings.

>

>Note that an ordinary **enum** does not create its own name space, so the symbols must be unique within the current scope, whereas a **transition enum** creates its own name space, and so the state names can only be used where the state variables are known.

>Then current and next state variables can be declared.

After doing the coding, I like the idea of just declaring **enum** (no **transition enum**).

It is highly unusual that someone would put multiple FSM designs in the same file for the following reasons:

- (a) In Verilog, it is typically recommended to keep FSM designs in separate files for modularity and documentation purposes.
- (b) Commercial tools can manipulate FSM designs if they are in separate files.
- (c) To do multiple FSM designs in the same file using Verilog, naming conventions would already have been used to avoid parameter name collisions. I do not see the big advantage of separate "transition enums" just to reuse a state name.

>The enum\_onehot is a possible extension.

This is still a glaring omission in all HDLs today and an opportunity to really make a valuable contribution to HDL design with a new SystemVerilog syntax. It should probably be postponed until SystemVerilog 3.1 to make sure we get it right. Like I mentioned before, this is not addressed efficiently even by VHDL.

-----

>2. Remove the timing control as you suggest so that a separate always block can be used to clock next into current.

>**always** @(posedge clock or negedge reset)

> if (reset) next <= current;

> else next <= S0;

Agreed.

-----

>3. Retain the begin-end because this is the same syntax as the case item, which eliminates syntactic ambiguity between a statement and a case expression.

Diminishes the value of adding a new case-like keyword, **transition**, but more on this later.

-----

>4. Add a new syntax for transition blocks as you suggest so that you can write **transition** (current: ->> next), which then allows blocking assignments of the form:

```

> ->> S1; // means next = S1;
>
>and a new syntax transition (current: ->>> next), which then allows nonblocking
assignments in the transition items:
> ->>> S1; // means next <= S1;
>
>Note that the transitions can be labeled as now:
> T0_1 ->> S1;
>
>and can be used outside a transition statement in a hierarchical way
> ->> next.S1;

```

Agreed, but let's remove the unnecessary ":" from the syntax (all examples assume the : is removed).

-----

>BTW indexing into a vector can already be done from an enumeration e.g  
myvec[int'(myenum)].

Interesting, but this still requires significant code modifications to turn an FSM design into an efficient onehot FSM design.

>Peter

-----

Additional proposals:

**PROPOSAL:** Remove the combined transition/output assignment syntax used in the S1 transition of the first FSM example shown on page 32 of draft 4. It only benefits the case of only one output assignment for each transition in the one always block examples. Not worth cluttering the syntax for a small subset of actual usage assignments.

We need the ability to declare enumerated types with the value of 'x for synthesis efficiency purposes (shown in all of the fsm\_\_\_a.v examples).

Which of the following is true?

// transition (state) is equivalent to which of the following?

// transition (state ->> state)

// transition (state ->>> state)

// ILLEGAL:

For clarity, I believe the syntax should require an explicit syntax and therefore **transition (state)** should now be illegal.

**PROPOSAL:** remove all discussion of hierarchical FSMs from the SystemVerilog standard until at least version 3.1. We have not yet established any value to using this capability.