Proposal to fix keyword compatibility issues between 1364 and P1800

Add new section 25.4, as follows:

25.4 `keywords, `endkeywords

SystemVerilog provides a pair of directives, **‘keywords** and **‘endkeywords**, that specify what identifiers are reserved as keywords within a block of source code, based on a specific version of the IEEE 1364 Verilog standard or the IEEE 1800 SystemVerilog standard.

The ‘keywords and ‘endkeywords directives only specify the set of identifiers that are reserved as keywords. The directives do not affect the semantics, tokens and other aspects of the Verilog and SystemVerilog languages.

The syntax of the ‘keywords and ‘endkeywords directives are:

```
keywords_directive ::= `keywords "version_specifier"

version_specifier ::=
     default
   | 1364-1995
   | 1364-2001
   | 1364-2005
   | 1800-2005

endkeywords_directive ::= `endkeywords
```

The ‘keywords directive can only be specified outside of a **module**, **primitive**, **interface**, **program** or **package**. It affects all modules, primitives, interfaces, programs or packages that follow the directive, even across source code file boundaries. Multiple ‘keywords directives are allowed, but the effects of the ‘keywords directive are not nested. The latest occurrence of the directive in the source code controls the set of identifiers that are reserved keywords for the source code that follows the directive.

Implementations and other standards are permitted to extend the ‘keywords directive with custom version specifiers. It shall be an error if an implementation does not recognize the *version_specifier* used with the ‘keywords directive.

The directives ‘endkeywords and ‘resetall shall terminate the effect of the ‘keywords directive. ‘endkeywords and ‘resetall have the same effect as specifying ‘keywords "default".

If no ‘keywords directive is specified, then the reserved keyword list shall be implementation dependent. The lack of a ‘keywords directive is the same as if the directive ‘keywords "default" had been specified.

The *version_specifier* "default" specifies that the implementation determines the set of reserved keywords that are in effect. For example, an implementation based on the IEEE 1364-2005 Verilog standard would most likely use the 1364-2005 set of reserved keywords as its default. Implementations may provide other mechanisms for specifying the set of reserved keywords to be used as the default. One possible use model might be that an implementation provides invocation options to establish its set of default keywords. Another possible use model might be the use of source file name extensions for determining a default set of reserved keywords to be used for each source file.

The *version_specifier* "1364-1995" specifies that only the identifiers listed as reserved keywords in the IEEE 1364-1995 standard are considered to be reserved words. These identifiers are listed in Table 25-1.

Table 25-1: IEEE 1364-1995 reserved keywords

always	for	output	supply0
and	force	parameter	supply1
assign	forever	pmos	table
begin	fork	posedge	task
buf	function	primitive	time
bufif0	highz0	pull0	tran
bufif1	highz1	pull1	tranif0
case	if	pullup	tranif1
casex	ifnone	pulldown	tri
casez	initial	rcmos	tri0
cmos	inout	real	tri1
deassign	input	realtime	triand
default	integer	reg	trior
defparam	join	release	trireg
disable	large	repeat	vectored
edge	macromodule	rnmos	wait
else	medium	rpmos	wand
end	module	rtran	weak0
endcase	nand	rtranif0	weak1
endmodule	negedge	rtranif1	while
endfunction	nmos	scalared	wire
endprimitive	nor	small	wor
endspecify	not	specify	xnor
endtable	notif0	specparam	xor
endtask	notif1	strong0	
event	or	strong1	

The *version_specifier* `"1364-2001"` specifies that only the identifiers listed as reserved keywords in the IEEE 1364-2001 standard are considered to be reserved words. These identifiers are listed in Table 25-2.

Table 25-2: IEEE 1364-2001 reserved keywords

always	event	noshowcancelled	specify
and	for	not	specparam
assign	force	notif0	strong0
automatic	forever	notif1	strong1
begin	fork	or	supply0
buf	function	output	supply1
bufif0	generate	parameter	table
bufif1	genvar	pmos	task
case	highz0	posedge	time
casex	highz1	primitive	tran
casez	if	pull0	tranif0
cell	ifnone	pull1	tranif1
cmos	incdir	pulldown	tri
config	include	pullup	tri0
deassign	initial	pulsestyle_onevent	tri1
default	inout	pulsestyle_ondetect	triand
defparam	input	rcmos	trior
design	instance	real	trireg
disable	integer	realtime	unsigned
edge	join	reg	use
else	large	release	vectored
end	liblist	repeat	wait
endcase	library	rnmos	wand
endconfig	localparam	rpmos	weak0
endfunction	macromodule	rtran	weak1
endgenerate	medium	rtranif0	while
endmodule	module	rtranif1	wire
endprimitive	nand	scalared	wor
endspecify	negedge	showcancelled	xnor
endtable	nmos	signed	xor
endtask	nor	small	

The *version_specifier* `"1364-2005"` specifies that only the identifiers listed as reserved keywords in the IEEE 1364-2005 standard are considered to be reserved words. These identifiers are listed in Table 25-3.

Table 25-3: IEEE 1364-2005 reserved keywords

always	event	noshowcancelled	specify
and	for	not	specparam
assign	force	notif0	strong0
automatic	forever	notif1	strong1
begin	fork	or	supply0
buf	function	output	supply1
bufif0	generate	parameter	table
bufif1	genvar	pmos	task
case	highz0	posedge	time
casex	highz1	primitive	tran
casez	if	pull0	tranif0
cell	ifnone	pull1	tranif1
cmos	incdir	pulldown	tri
config	include	pullup	tri0
deassign	initial	pulsestyle_onevent	tri1
default	inout	pulsestyle_ondetect	triand
defparam	input	rcmos	trior
design	instance	real	trireg
disable	integer	realtime	unsigned
edge	join	reg	use
else	large	release	uwire
end	liblist	repeat	vectored
endcase	library	rnmos	wait
endconfig	localparam	rpmos	wand
endfunction	macromodule	rtran	weak0
endgenerate	medium	rtranif0	weak1
endmodule	module	rtranif1	while
endprimitive	nand	scalared	wire
endspecify	negedge	showcancelled	wor
endtable	nmos	signed	xnor
endtask	nor	small	xor

The *version_specifier* `"1800-2005"` specifies that only the identifiers listed as reserved keywords in the IEEE 1800-2005 standard are considered to be reserved words. These identifiers are listed in Table 25-4.

Table 25-4: IEEE 1800-2005 reserved keywords

alias	endmodule	matches	small
always	endpackage	medium	solve
always_comb	endprimitive	modport	specify
always_ff	endprogram	module	specparam
always_latch	endproperty	nand	static
and	endspecify	negedge	string
assert	endsequence	new	strong0
assign	endtable	nmos	strong1
assume	endtask	nor	struct
automatic	enum	noshowcancelled	super
before	event	not	supply0
begin	expect	notif0	supply1
bind	export	notif1	table
bins	extends	null	tagged
binsof	extern	or	task
bit	final	output	this
break	first_match	package	throughout
buf	for	packed	time
bufif0	force	parameter	timeprecision
bufif1	foreach	pmos	timeunit
byte	forever	posedge	tran
case	fork	primitive	tranif0
casex	forkjoin	priority	tranif1
casez	function	program	tri
cell	generate	property	tri0
chandle	genvar	protected	tri1
class	highz0	pull0	triand
clocking	highz1	pull1	trior
cmos	if	pulldown	trireg
config	iff	pullup	type
const	ifnone	pulsestyle_onevent	typedef
constraint	ignore_bins	pulsestyle_ondetect	union
context	illegal_bins	pure	unique
continue	import	rand	unsigned
cover	incdir	randc	use
covergroup	include	randcase	uwire
coverpoint	initial	randsequence	var
cross	inout	rcmos	vectored
deassign	input	real	virtual
default	inside	realtime	void
defparam	instance	ref	wait
design	int	reg	wait_order
disable	integer	release	wand
dist	interface	repeat	weak0
do	intersect	return	weak1
edge	join	rnmos	while
else	join_any	rpmos	wildcard
end	join_none	rtran	wire
endcase	large	rtranif0	with
endclass	liblist	rtranif1	within
endclocking	library	scalared	wor
endconfig	local	sequence	xnor
endfunction	localparam	shortint	xor
endgenerate	logic	shortreal	
endgroup	longint	showcancelled	
endinterface	macromodule	signed	

In the example below, it is assumed that the definition of module `m1` does not have a `‘keywords` directive specified prior to the module definition. Without this directive, the set of reserved keywords in effect for this module shall be the implementation’s default set of reserved keywords.

```
module m1;  // module definition with no ‘keywords directive
  ...
endmodule
```

The follows example specifies a `‘keywords = "1364-2001"` directive. The source code within the module uses the identifier `logic` as a variable name. The `‘keywords` directive would be necessary in this example if an implementation uses the 1800-2005 SystemVerilog standard as its default set of keywords, because `logic` is a reserved keyword in SystemVerilog. Specifying that the 1364-1995 or 1364-2005 Verilog keyword lists should be used would also work with this example.

```
‘keywords "1364-2001"   // use IEEE 1364-2001 Verilog keywords
module m2 (...);
  reg [63:0] logic;     // OK: "logic" is not a keyword in 1364-2001
  ...
endmodule
‘endkeywords            // change to implementation’s default keyword set
```

The next example is the same code as the previous example, except that it explicitly specifies that the IEEE 1800-2005 SystemVerilog keywords should be used. This example shall result in an error, because `logic` is reserved as a keyword in the SystemVerilog standard.

```
‘keywords "1800-2005"   // use IEEE 1800-2005 SystemVerilog keywords
module m2 (...);
  reg [63:0] logic;     // ERROR! "logic" is a keyword in 1800-2005
  ...
endmodule
‘endkeywords            // change to implementation’s default keyword set
```

The following example specifies a `‘keywords` directive on an **`interface`** declaration. The directive specifies that an implementation shall use the set of reserved keywords specified in the IEEE 1800-2005 SystemVerilog standard.

```
‘keywords "1800-2005"   // use IEEE 1800-2005 SystemVerilog keywords
interface if1 (...);
  ...
endinterface
‘keywords "default"     // change to implementation’s default keyword set
```

The next example is nearly identical to the one above, except that the `‘keywords` directive specifies that the IEEE 1364-2005 Verilog set of keywords shall be used. This example shall result in errors, because the identifiers `interface` and `endinterface` are not reserved keywords in the IEEE 1364-2005 Verilog standard.

```
‘keywords "1364-2005"   // use IEEE 1364-2005 Verilog keywords
interface if2 (...);    // ERROR! "interface" is not a keyword in 1364-2005
  ...
endinterface            // ERROR! "endinterface" is not a keyword in 1364-2005
‘keywords "default"     // change to implementation’s default keyword set
```