

Языки описания схем

mk.cs.msu.ru → Лекционные курсы → Языки описания схем

Блок 18

Verilog:
Комбинационные выражения и операции

Лектор:
Подымов Владислав Васильевич
E-mail:
valdus@yandex.ru

ВМК МГУ, 2023/2024, осенний семестр

Комбинационные выражения

В непрерывном присваивании, блокирующем присваивании и некоторых других конструкциях используются выражения (будем называть их **комбинационными**), которые можно трактовать двояко:

- ▶ *Программно*: для заданных значений точек выражение имеет значение (**результат** вычисления) заданной ширины и заданной знаковости
- ▶ *Аппаратно*: значение выражения — это любая комбинационная схема, в которой:
 - ▶ имеется одна выходная шина, ширина которой задаётся программной трактовкой
 - ▶ входами являются все точки, использующиеся в выражении
 - ▶ значение выходной шины задаётся значениями входных шин согласно программной трактовке

Комбинационное выражение строится над точками, константами и **комбинационными операциями**, из которых обсудим только те, которые имеют аппаратную семантику

Основные комбинационные операции

	<code>x && y</code>		<code>x y</code>		<code>!x</code>	
<code>x + y</code>	<code>x - y</code>	<code>+x</code>	<code>-x</code>	<code>x * y</code>	<code>x / y</code>	<code>x % y</code>
<code>x == y</code>	<code>x != y</code>	<code>x > y</code>	<code>x >= y</code>	<code>x < y</code>	<code>x <= y</code>	
<code>x << y</code>	<code>x >> y</code>	<code>x <<< y</code>	<code>x >>> y</code>			
<code>x & y</code>	<code>x y</code>	<code>x ^ y</code>	<code>~x</code>			
<code>&x</code>	<code> x</code>	<code>~x</code>	<code>~&x</code>	<code>~ x</code>	<code>~^x</code>	
		<code>x ? y : z</code>				
	<code>x[i]</code>		<code>x[i:j]</code>			

Очень похоже на операции C/C++ — и это действительно так, но с поправкой на особенности семантики \mathcal{V} и на «технические тонкости» (ширина, знаковость и т.п.)

Логические операции

`x && y`

`x || y`

`!x`

Результат — беззнаковая шина ширины 1, значение которой определяется согласно таблицам

		x && y		
		0	\mathcal{X}/\mathcal{Z}	1
x	y			
	0	0	0	0
	\mathcal{X}/\mathcal{Z}	0	\mathcal{X}	\mathcal{X}
	1	0	\mathcal{X}	1

		x y		
		0	\mathcal{X}/\mathcal{Z}	1
x	y			
	0	0	\mathcal{X}	1
	\mathcal{X}/\mathcal{Z}	\mathcal{X}	\mathcal{X}	1
	1	1	1	1

		!x
x		!x
0		1
\mathcal{X}/\mathcal{Z}		\mathcal{X}
1		0

При вычислении результата аргументы сужаются до ширины 1

Арифметические операции

$x + y$

$x - y$

$+x$

$-x$

$x * y$

x / y

$x \% y$

Результат — естественный для арифметики с переполнением

Если хотя бы один из аргументов не определён,
то результат — ($\mathcal{X}\mathcal{X}\dots\mathcal{X}$)

Учёт знаковости при вычислении стандартный

Ширина выражения — это максимум среди

- ▶ ширины аргументов
- ▶ ширины левой части присваивания,
если «над» операцией стоит присваивание
- ▶ ширины «надвыражения»,
если это выражение есть и имеет ширину

Ширина результата равна ширине выражения

При вычислении результата

аргументы расширяются до ширины выражения

Арифметические отношения

$$x == y$$

$$x != y$$

$$x > y$$

$$x >= y$$

$$x < y$$

$$x <= y$$

Результат — беззнаковая шина ширины 1 со следующим значением:

- ▶ если оба числа-аргумента **определены**, то
 - ▶ 1, если числа входят в отношение, и
 - ▶ 0, если не входят
- ▶ иначе результат — \mathcal{X}

Учёт знаковости при вычислении **стандартный**,
но только результат беззнаковый

При вычислении результата
узкий аргумент расширяется до широкого

Сдвиговые операции

 $x \ll y$ $x \gg y$ $x \lll y$ $x \ggg y$

Результат имеет ту же ширину и знаковость, что и x

« \lll » и « \llll »:

сдвиг x влево на y разрядов с заполнением нолями

« \ggg »: (логический сдвиг)

результат — сдвиг x вправо

на y разрядов с заполнением нолями

« \gggg »: (арифметический сдвиг)

результат — сдвиг x вправо на y разрядов

с заполнением при помощи расширения шины согласно знаковости

Если число y не определено, то результат — $(\mathcal{X}\mathcal{X}\dots\mathcal{X})$

Многобитовые логические операции

$$x \ \& \ y$$

$$x \ | \ y$$

$$x \ \sim \ y$$

$$\sim x$$

Ширина результата равна ширине самого широкого аргумента

Учёт знаковости при вычислении **стандартный**

Узкий аргумент расширяется до широкого беззнаково

Каждый разряд результата

получается из соответствующих разрядов x и y

применением соответствующей операции: $\&$, $|$, $!$, $!$

Операции редукции

 $\&x$ $|x$ \hat{x} $\sim\&x$ $\sim|x$ $\sim\hat{x}$

Результат — беззнаковая шина ширины 1 со следующим значением:

- ▶ Для операции без « \sim »:
соответствующая логическая операция ($\&$, $|$, $!$)
применяется к паре младших разрядов, и затем,
итеративно до конца шины,
к результату предыдущего шага и следующему разряду
- ▶ Для операции с « \sim »:
отрицание (!) результата
соответствующей операции без « \sim »

Тернарный оператор

$x \ ? \ y \ : \ z$

Ширина результата — это максимум ширин y и z

Узкий аргумент y/z расширяется до широкого беззнаково

Результат зависит от **условия** — значения выражения « $x == 0$ »:

- ▶ $1 \Rightarrow$ результат совпадает с z
- ▶ $0 \Rightarrow$ результат совпадает с y
- ▶ $\mathcal{X}/\mathcal{Z} \Rightarrow i$ -й разряд результата — это
 - ▶ i -й разряд y , если i -е разряды y и z равны
 - ▶ \mathcal{X} иначе

Учёт знаковости при вычислении **стандартный**

Операции индексации

`x[i]`

`x[i:j]`

`x[i]` — разряд шины `x` с индексом `i`

`x[i:j]` — шина (`x[i] ... x[j]`)

Индексация может использоваться в левых частях присваиваний (~ `lvalue`) с сохранением категории типа

Операции конкатенации и репликации

$$\{x_1, x_2, \dots, x_n\}$$
$$\{N\{x_1, x_2, \dots, x_n\}\}$$

Результат **конкатенации** $\{x_1, x_2, \dots, x_n\}$

шин $x_i = (x_i[k_i-1] \dots x_i[0])$ — беззнаковая шина

$$(x_1[k_1-1] \dots x_1[0] \ x_2[k_2-1] \dots x_2[0] \ \dots \ x_n[k_n-1] \dots x_n[0])$$

Репликация $\{N\{x_1, x_2, \dots, x_n\}\}$

равносильна конкатенации N копий выражения $\{x_1, x_2, \dots, x_n\}$

Конкатенация может использоваться в левых частях присваиваний (\sim lvalue), с сохранением категории типа