

Языки описания схем

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

Блок 16

Verilog:

Арифметическая трактовка значений

Знаковость типов и значений

Расширение и сужение шин

Лектор:

Подымов Владислав Васильевич

E-mail:

valdus@yandex.ru

ВМК МГУ, 2024/2025, осенний семестр

У: Знаковость типов и значений

В У используются две арифметических трактовки значения в шине:

► **Беззнаковая**:

$$(\alpha_{n-1} \dots \alpha_0)_2 = \sum_{i=0}^{n-1} 2^i \cdot \alpha_i$$

► **Знаковая** (дополнительный код):

$$(\alpha_{n-1} \dots \alpha_0)_2^- = (\overline{\alpha_{n-1}} \alpha_{n-2} \dots \alpha_0)_2 - 2^{n-1}$$

У каждого типа и у каждого значения есть **знаковость** — то, какая арифметическая трактовка для него используется

Язык У устроен так, что *по умолчанию* всё оказывается беззнаковым за нечастыми (явно прописанными) исключениями

Арифметическая трактовка корректна, если значение состоит **только** из 0 и 1, а иначе соответствующее число **не определено**

Одноразрядная точка обычно отождествляется с шиной точек ширины 1

У: Знаковость типов и значений

При вычислении выражений знаковость результата может соотноситься со знаковостью выражений самыми разными способами

Эти способы подобраны «естественно»,
чтобы не думать слишком много,
но иногда незнание этих способов приводит к ошибкам

Поэтому в описании операций выражений будет явно рассказываться о способе учёта знаковости при вычислении,
и часто будет упоминаться **стандартный** способ:

- ▶ результат выполнения операции объявляется беззнаковым \Leftrightarrow хотя бы один из аргументов беззнаковый
- ▶ перед вычислением результата знаковость аргументов приравнивается знаковости результата

Если знаковость аргумента при вычислении выражения меняется, то это означает **только** изменение арифметической трактовки значения, но **не** изменение самих разрядов, образующих значение

У: Знаковость типов и значений

По умолчанию значения точек для типов, основанных на `wire` и `reg`, являются беззнаковыми

Чтобы значения точки по умолчанию трактовались как знаковые, можно добавить к типу в объявлении точки ключевое слово

`signed`

Например: `wire signed [1:0] x;`

Чтобы поменять трактовку значения *выражения* на знаковую, достаточно написать

`$signed(<выражение>)`

Аналогично для изменения трактовки на беззнаковую:

`$unsigned(<выражение>)`

У: Расширение и сужение шин

В семантику некоторых комбинационных операций включены механизмы **выравнивания** ширины аргументов: **расширения** слишком узких и **сужения** слишком широких

Сужение всегда устроено одинаково: отбрасываются старшие разряды

$$(x_{n-1}x_{n-2} \dots x_k x_{k-1} x_{k-2} \dots x_0) \rightsquigarrow (x_{k-1}x_{k-2} \dots x_0)$$

Расширение, как правило, относится к одному из двух видов:

1. **Беззнаковое**: в старшие разряды дописывается 0

$$(x_{k-1}x_{k-2} \dots x_0) \rightsquigarrow (00 \dots 0x_{k-1}x_{k-2} \dots x_0)$$

2. **Знаковое**: старший разряд «размножается» нужное число раз

$$(x_{k-1}x_{k-2} \dots x_0) \rightsquigarrow (x_{k-1}x_{k-1} \dots x_{k-1}x_{k-1}x_{k-2} \dots x_0),$$

то есть

$$(0x_{k-2} \dots x_0) \rightsquigarrow (00 \dots 00x_{k-2} \dots x_0)$$

$$(1x_{k-2} \dots x_0) \rightsquigarrow (11 \dots 11x_{k-2} \dots x_0)$$

$$(x x_{k-2} \dots x_0) \rightsquigarrow (x x \dots x x x_{k-2} \dots x_0)$$

$$(z x_{k-2} \dots x_0) \rightsquigarrow (z z \dots z z x_{k-2} \dots x_0)$$