3.6.1 Integers

An integer number can be written in the following two forms.

i. Simple decimal
ii. Base format

Simple Decimal Form

An integer in this form is specified as a sequence of digits with an optional + (unary) or a – (unary) operator. Here are some examples of integers in the simple decimal form.

```
32          is decimal 32
- 15        is decimal -15
```

An integer value in this form represents a signed number. A negative number is represented in two's complement form. Thus 32 is 100000 in a

6-bit binary, 0100000 in 7-bit binary; –15 is 10001 in 5-bit binary, and is 110001 in a 6-bit binary.

The number of bits in an integer of this form is at least 32 bits.

Base Format Notation

The format of an integer in this form is:

```
[ size ] ' [ signed ] base value
```

where the *size* specifies the size of the constant in number of bits, *signed* is one of **s** or **S**, *base* is one of **o** or **O** (for octal), **b** or **B** (for binary), **d** or **D** (for decimal), **h** or **H** (for hexadecimal) and *value* is a sequence of digits that are values from the *base*. The values **x** and **z** and the hexadecimal values **a** through **f** are case-insensitive. The integer is a signed integer when the signed qualifier is present, else it is an unsigned integer.

Here are some examples.

```
5'O37       5-bit octal
4'D2        4-bit decimal
4'B1x_01    4-bit binary
7'Hx        7-bit x (x extended), that is, xxxxxxx
4'hZ        4-bit z (z extended), that is, zzzz
8 'h 2A     Spaces are allowed between size and '
            character and between base and value
8'sh51      8-bit signed number 01010001
6'so72      6-bit signed number 111010 which is
            decimal -6 which is same as 6'sh3A

// Examples of not legal integers:
4'd-4       Not legal: value cannot be negative
3' b001   Not legal: no space allowed between ' and base b
(2+3)'d10 Not legal; size cannot be an expression
```

Note that an **x** (or **z**) in a hexadecimal value represents four bits of **x** (or **z**), **x** (or **z**) in octal represents three bits of **x** (or **z**), and **x** (or **z**) in binary represents one bit of **x** (or **z**).

The size specification is optional in an integer of this form. If no size is specified in an integer, the size of the number is at least 32 bits. Here are some examples.

```
'o721      32-bit octal, unsigned
'hAF       32-bit hex, unsigned
'sb1011    32-bit binary, signed -5
```

If the size specified is larger than the number of bits specified for the constant, the number is padded to the left with 0's for unsigned numbers and with the leftmost bit for signed numbers. For the case where the leftmost bit is a **x** or a **z**, a **x** or a **z** respectively is used to pad to the left. For example,

```
10'b10       Padded with 0 to the left, 0000000010
10'bx0x1     Padded with x to the left, xxxxxxx0x1
8'sb101101   Padded with sign bit (1) to the
             left, 11101101
```

If the size specified is smaller than the number of bits specified for the constant, then the leftmost bits are appropriately truncated. For example,

The sign bit is not preserved during truncation.

```
3'b1001_0011      is same as 3'b011
5'H0FFF           is same as 5'H1F
3'sb10100         is same as 3'sb100
```

Similarly, when no size is specified and the target size required is larger, then the number is padded with 0's for unsigned numbers and with the leftmost bit for signed numbers. For the case when the leftmost bit is a **x** or a **z**, a **x** or a **z** respectively is used to pad to the left.

```
reg [15:0] pbus_select;

pbus_select = 'h5;   // Pads 0 to the left and
                     // assigns 0000 0000 0000 0101.
pbus_select = 'hx11; // Pads x to the left and
                     // assigns xxxx xxxx 0001 0001.
pbus_select = 'hz51; // Pads z to the left and
                     // assigns zzzz zzzz 0101 0001.
```

The ? character can be used as an alternate for value z in a number. It may be used to improve readability in cases where the value z is interpreted as a don't-care value (see Chapter 8).