EEE4084F Digital Systems

C to VHDL translation tool language specification

(loosely based on the Handle-C syntax)

The C to VHDL translation tool supports a large portion of the ANSI C syntax standard. The supported datatypes and modifiers are listed in the table below. The **bit**, **byte**, and **short** datatypes are commonly used, together with the **in** and **out** modifiers. As in ANSI C, the unsigned, short and long keywords can be used as datatypes if used alone (e.g. int ix) or as a modifier if used with another dataype (e.g. unsigned int ux). **Floating point** values (e.g. float, double) are not supported.

Support for sized arrays but not for pointers or unsized arrays

Note: pointers not supported as either parameters or as variable declarations. But *arrays* are supported.

Examples:

SUPPORTED	NOT SUPPORTED		
void test (int p [10]);	void test (int* p);	OR	void test (int p[]);
int array1[10];	int* array1;	OR	int array1[];

Datatype sizing

Int and **unsigned** (or unsigned int) datatypes can have their their size (in number of bits) modified. All other datatypes (bit, bool, nibble, byte, char, unsigned char, llint, ulint, etc) cannot have their size modified.

The syntax for arbitrary sized declaration is as follows:

type size name;

Type: the datatype, namely: int, unsigned or unsigned int
Size: a positive integer (between 1 and 128)
Name: name of the variable

Examples:

SUPPORTED	NOT SUPPORTED
int 8 signedbyte;	char 8 signedbyte;
int 8 signedbyte;	char 8 signedbyte;
int 6 intarray[10];	int* 6 intarray;
unsigned 11 xu;	unsigned char 11 xu;
int 7 xu;	byte 7 xu;

Arbitrary sized integers/unsigned variables can be used as are normal integers / unsigned values. Only the least significant bits are processed/copied; for example (int 2 x = 4; // x is set to 0 as the two least significant bits of integer 4 are both 0.)

Example:

```
unsigned int 7 x = 20;
int y = 10;
x += 1; // x changed to 21
x = x + y; // x changed to 31
y = x;  // y set to 31
```

See the next page for list of datatypes.

Datatypes

C -> VHDL Translator datatype/modifier	**Default size**	**ANSI-C Standard equivalent keyword**	**Comments**
_in		N/A	Indicate input parameter (use only with function parameters)
_out		N/A	Indicate output parameter (use only with function parameters)
enum	1 to 4 bits	enum	Translator limited enums limited to sets of 16 items
bram		N/A	Use as datatype or as modifier (e.g. bram int x = 10;) Forces data into block RAM. “bram” without type => “bram int”
sram		N/A	Use as datatype or as modifier (e.g. sram int x = 10;) Forces data into SRAM if available; otherwise into BRAM. “sram” without type equates to “sram int”
dram		N/A	Use as datatype or as modifier (e.g. dram int x = 10;) Forces data into DRAM or external ram if available; else into BRAM. “dram” without type equates to “dram int”
ext		N/A	Use as datatype or as modifier (e.g. ext int x = 10;) Forces data into external memory if available; otherwise into BRAM. “ext” without type equates to “ext int”
rom		N/A	Use as datatype or as modifier (e.g. rom int x = 10;) Forces data into read only memory if available; otherwise into BRAM. “rom” without type equates to “rom int”. Do not confuse keyword “rom” with ASNI keyword “const” -- const a variable cannot be changed (e.g., “const bram y = 5;” means *y* is located in BRAM but cannot be changed by the C program)
int	32-bit	int	Signed 32-bit value
short	16-bit	short	Signed 16-bit value
unsigned	32-bit	unsigned	Unsigned 32-bit value
unsigned short	16-bit	unsigned short	Unsigned 16-bit value
char	8-bit	char	Signed 8-bit value (-128 to +127)
unsigned char	8-bit	unsigned char	Unsigned 8-bit value (0 to 255)
byte	8-bit	unsigned char	Unsigned 8-bit value (0 to 255)
nibble	4-bit	N/A	Unsigned 4-bit value (0 to 15)
bit	1-bits	N/A	Single bit (0 to 1)
bool	1-bit	N/A	Single bit (0 to 1) equivalent to bit
long	32-bit	long	Signed 32-bit value
unsigned long	32-bit	unsigned long	Unsigned 32-bit value
long long	64-bit	long long	Signed 64-bit value
unsigned long long	64-bit	unsigned long long	Unsigned 64-bit value
llint	64-bit	long long	Signed 64-bit value
ulint	64-bit	Unsigned long long	Unsigned 64-bit value