

# NUMERIC TYPES FOR SYNTHESIS

Alex Zamfirescu

Vantage Analysis Systems Inc.

42808 Christy St. Suite 200

Fremont, CA. 94538

Phone : (510) 659 0901

## Summary

The effort for developing a standard VHDL library containing packages for synthesis has led to progress in the following areas:

1. Interpretation of logic types
2. Manipulating numeric types
3. Representing design constraints
4. Identifying other data needed by synthesis

The talk will present the two packages implementing a proposed solution for handling in a standard way numeric types in synthesis and simulation (item 2 above).

Definitions for overloaded arithmetic operations for `std.standard.bit_vector` and types defined in `ieee.std_logic_1164` package are introduced, the rationales for all major decisions are explained, and the body implementation alternatives are discussed.

Features that will benefit from the proposed changes in the VHDL '92 new standard are outlined.

Finally the users will learn about how a standard for numeric types in synthesis and simulation will benefit them, and how they can get involved and influence the process.

# NUMERIC TYPES FOR SYNTHESIS

October 20, 1992

Alex Zamfirescu

Vantage Analysis Systems, Inc.  
42808 Christy St. Suite 200  
Fremont, CA 94538  
USA  
(510) 659-0901  
anz@vas.com

(C) 1992 Vantage Analysis Systems, Inc.



## OUTLINE

The VHDL '92 requirements analysis phase left SSIG a clear objective

Numeric types effort covers RTL level "synthesizable" types, arithmetic operations (like "+", "-", "\*", "/", abs, mod, rem), comparison operators (like "<", "<=", ">", ">=", eq) and conversions between types

Today presenting

- main decisions taken so far
- a view of the synthesis library proposal
- future plans

---

*Acknowledgements: I would like to thank all SSIG members for their participation and inspired ideas. Special recognition is due to Ken Scott, Dave Ackley and Jean-Michel Bergé for their contributions*



## SSIG NUMERIC TYPES INITIAL REQUIREMENTS

Consensus reached via E-mail

1. Unconstrained bit-arrays *types* representing **signed** and **unsigned** integers
2. Signed should be 2's complement
3. Enable X propagation
4. *No need* for hex, floating point (provide for later addition), resource allocation
5. Support IEEE standard logic type (no subtypes)
6. *Possible* multiple packages, prototype using 1987 VHDL, vector to/from integer



## SOME INTERESTING DECISIONS

- 'LEFT, 'RIGHT versus 'HIGH, 'LOW for the order msb, lsb
  - VHDL array equality is defined based on 'LEFT – 'RIGHT order
  - > S'LEFT is the msb for any array object of subtype S
- Same size versus fill or extend
  - > use different sizes when possible
- Field size constraints for procedures
  - > for "+" and "-" result size is the largest operand size
  - > for "\*" result size is the sum of the operand sizes
  - > for "/" result size is the dividend size
- Strong, weak, or as computed result values
  - > only '0', '1', 'X', 'U' as elements of the result



## HOW X's INTERACT

### X PROPAGATION

	1 2 3 4 5 6 7 8 9	
	0 0 0 1 1 X 0 0 1	+
	0 0 0 1 0 1 0 0 1	
	-----	
CASE A	X X X X X X 0 1 0	
CASE B	X X X X X X X X X	
CASE C	0 0 1 X X X 0 1 0	
	^	

- > Support C
- > Multiplication X propagation based on addition propagation



## SIX ARRAY TYPES

Two symmetric packages

`BIT_VECTOR`

**package** NUMERIC\_BIT is

type `UNSIGNED_BV` is array (NATURAL range <>) of BIT;

type `COMP2_BV` is array (NATURAL range <>) of BIT;

...

**library** IEEE;

`STD_ULOGIC_VECTOR`

**use** STD\_LOGIC\_1164.all;

**package** NUMERIC\_STD is

type `UNSIGNED_LV` is array (NATURAL range <>) of  
STD\_ULOGIC;

type `COMP2_LV` is array (NATURAL range <>) of  
STD\_ULOGIC;

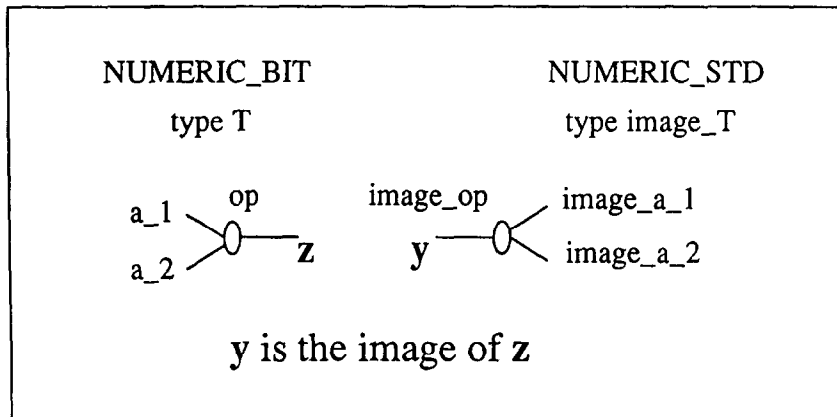


## A DESIRED PROPERTY

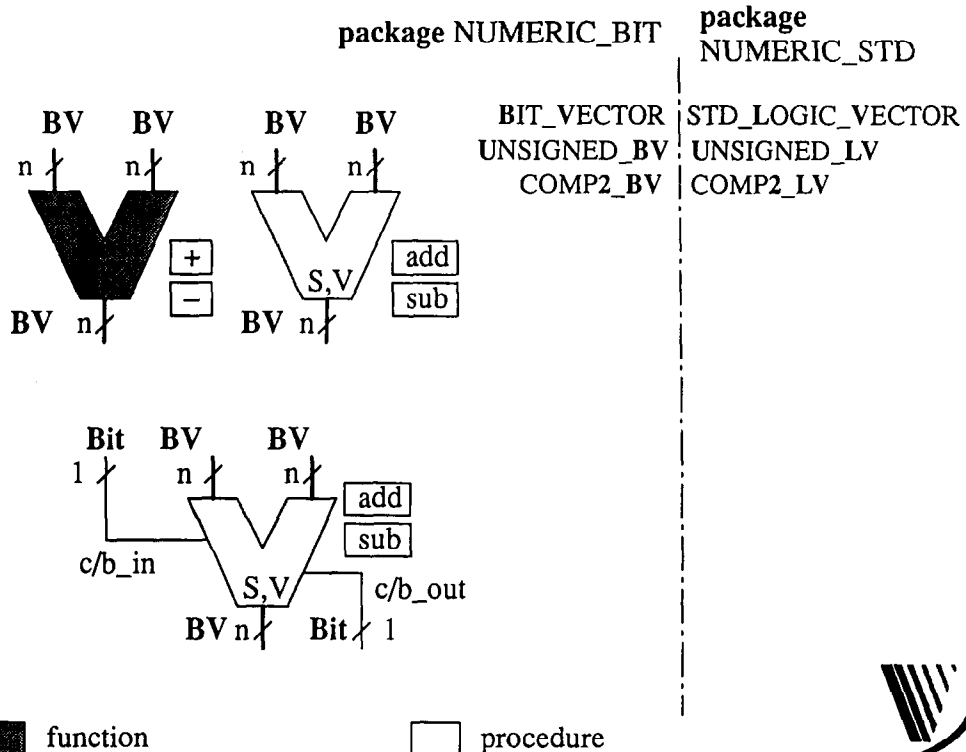
Corresponding operations are defined in both packages so that

$$\text{to\_std\_logic} ( \text{NUMERIC\_BIT.any\_op} ( \text{array\_1}, \text{array\_2} ) ) =$$

$$\text{NUMERIC\_STD.corresponding\_op} ( \text{to\_std\_logic} (\text{array\_1}),$$

$$\text{to\_std\_logic} (\text{array\_2}) )$$


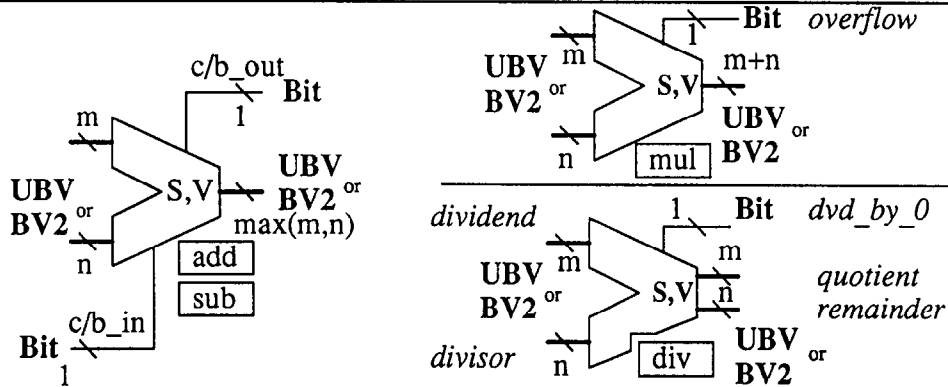
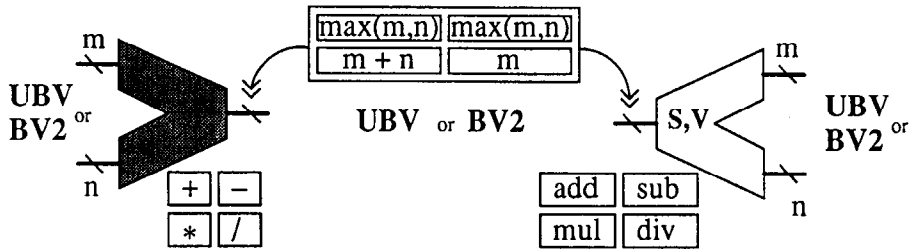
## SYNTHESIS LIBRARY-AT-A-GLANCE



# ARITHMETIC SUBPROGRAMS for ENCODED VECTORS

package NUMERIC\_BIT

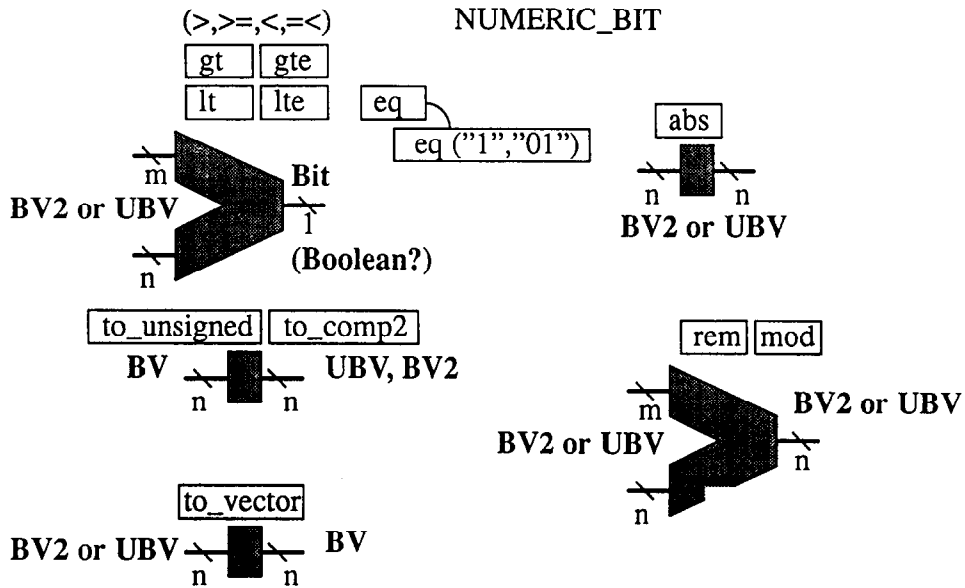
P  
a  
c  
k  
a  
g  
e  
  
N  
U  
M  
E  
R  
I  
C  
-  
S  
T  
D



function procedure

# SYN LIB EXTRAS

NUMERIC\_BIT



function

## NEW TOOL CHALLENGES

### SYNTHESIS

- Support as much as possible
- Selection criteria that improve quality

### SIMULATION

- Special treatment expected
  - speed
    - acceleration using foreign interfaces
    - pre-compiled library
  - friendly interface
    - to the library
    - run time special support

### FORMAL TECHNIQUES

- Take advantage of the relative reduced problem complexity
- Formally validated procedures



## VHDL '92 FEATURES

LCS #	TOPIC	IMPACT	prediction measure
05	private types.....	possible use to enable optimizations	<div style="width: 100%; height: 10px; background-color: black;"></div>
08	new aliasing.....	use aliases for procedures in allocations	<div style="width: 90%; height: 10px; background-color: black;"></div>
10	xnor.....	affects 1164 directly	<div style="width: 10%; height: 10px; background-color: black;"></div>
14	language-d att.....	use 'ASCENDING, 'VALUE, 'IMAGE	<div style="width: 90%; height: 10px; background-color: black;"></div>
16	logging messages.....	use in implementation and testing	<div style="width: 90%; height: 10px; background-color: black;"></div>
18	concatenation.....	affecting by definition shifts and rotate	<div style="width: 10%; height: 10px; background-color: black;"></div>
22	access to pred oper....	(uses LCS 08) use "=" instead of eq	<div style="width: 90%; height: 10px; background-color: black;"></div>
24	shift and rotate.....	default for bit arrays, overload for std	<div style="width: 10%; height: 10px; background-color: black;"></div>
26	foreign interface....	enable optimizations, validation needed	<div style="width: 90%; height: 10px; background-color: black;"></div>
41	run-t constraint check.....	increased robustness	<div style="width: 10%; height: 10px; background-color: black;"></div>
46	global variables.....	possible store flags	<div style="width: 100%; height: 10px; background-color: black;"></div>



## WHAT'S NEXT

Prototype package implementations

Test, validate usability, check portability issues

Propose it as a standard and follow legal IEEE procedures

Possible split and ballot before other outcome from SSIG

Get involved !



## CONCLUSIONS

- A strong start backed by a big momentum
- SSIG has the required expertise – no new inventions needed
- VHDL '92 has little impact on the proposed library
- Much work still ahead
- All pertinent input taken into account – consensus so far
- Another level where industry-wide cooperation and competition are orthogonal

