

Effect of Inertial Delay on Signal Drivers cont..

- Let us consider another example.

process

begin

ADDR_BUS <= 1 after 5 ns, 21 after 9 ns, 6 after 10ns,12 after 19 ns;

ADDR_BUS <= reject 4ns inertial 6 after 12 ns, 20 after 19 ns;

wait;

end process;


Effect of Inertial Delay on Signal Drivers cont..

- The summary of rules for adding a new transaction when inertial delay is used is
- 1. All transactions on a driver that are scheduled to occur at or after the delay of the new transaction are deleted (as in the transport case).
- 2 Add all the new transactions to the driver.

Effect of Inertial Delay on Signal Drivers cont..

3. For all the old transactions on the driver that occur at times between the time of the first new transaction (say F) and F minus the pulse rejection limit, delete the old transactions whose value is different from the value of the first new transaction.

Operator

Precedence	Operator Class	Operators					
Low	Logical	AND	OR	NAND	NOR	XOR	XNOR
	Relational	=	/=	<	<=	>	>=
	Shift	sll	srl	sla	sra	rol	ror
	Add	+	-	&			
	Sign	+	-				
	Multiply	*	/	mod	rem		
High	Miscellaneous	**	abs	not			

Logical Operator

- AND
- OR
- NAND
- NOR
- XOR
- NOT



Operate on bit BUT
can be overloaded
for operation
Bit_vector

- defined for the predefined types BIT and BOOLEAN.

Relational Operator

- **Compare the two values of the same base and return a Boolean value**
- **= (Equal)**
- **!= (Not equal)**
- **> (Greater than)**
- **>= (Greater than equal to)**
- **< (Less than)**
- **<= (Less than equal to)**
- **all relational operations is always BOOLEAN.**

Shift Operators

- Sll (Shift Left logical)
- Srl (Shift Right Logical)
- Sla (Shift Left Arithmetic)
- Sra (Shift Right Arithmetic)
- Rol (Rotate Left)
- Ror (Rotate right)

– A= 00001111

Adding Operator

- +
- -
- &

- Operators '+' and '-' are predefined for integer operands can be overloaded for operation on data of the type `bit_vector`.
- Concatenation operator '&' is predefined for one-dimensional array.
 - *SIGNAL A : BIT_VECTOR (0 TO 3);*
 - *SIGNAL B : BIT;*
 - *SIGNAL C : BIT_VECTOR(0 TO 4);*
 - *C <= B & A;*
- *IF A = "0000" AND B = '1' THEN C = "10000"*

Multiplying Operator

- *** (Multiply)**
- **/ (Divide)**
- **Rem**
- **Mod**
- predefined for both operands being of the same integer or floating point type.
- The result of a rem operation has the sign of its first operand and is defined as
- **$A \text{ rem } B = A - (A / B) * B$**
- The result of a mod operator has the sign of the second operand and is defined as
- **$A \text{ mod } B = A - B * N$**

Miscellaneous Operator

- **** (Exponent)**
- **Abs (absolute)**
- **Not**
- **The abs (absolute) operator is defined for any numeric type.**
- **The ** (exponentiation) operator is defined for the left operand to be of integer or floating point type and the right operand (i.e., the exponent) to be of integer type only.**
- **The not logical operator has the same precedence as the above two operators.**