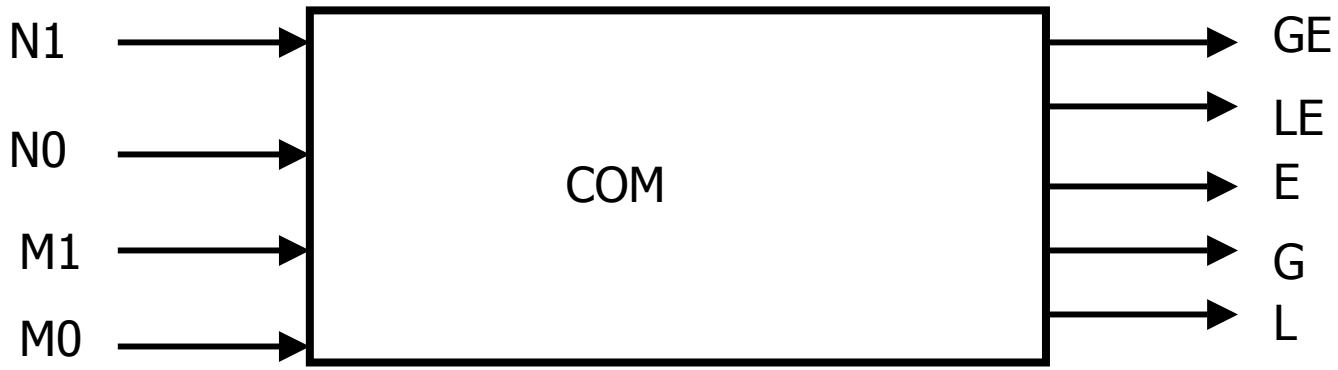
Binary Comparator [COM]



A device must compare two binary numbers and determine which is larger

INPUT: N=N1 N0 and M=M1 M0

OUTPUTS

GE	Binary output that is true when N is greater than or equal to M
LE	Binary output that is true when N is less than or equal to M
E	Binary output that is true when N is equal to M
G	Binary output that is true when N is strictly greater than M
L	Binary output that is true when N is strictly less than M

```
-- Device to compare two binary numbers
--
ENTITY com IS
        PORT (N1, N0, M1, M0 : IN BIT;
                GE, LE, E, G, L : OUT BIT;
END com;
```

TRUTH TABLE for 2-bit Comparator

N_1	N_0	M_1	M_0	GE	LE	E	G	L
0	0	0	0	1	1	1	0	0
0	0	0	1	0	1	0	0	1
0	0	1	0	0	1	0	0	1
0	0	1	1	0	1	0	0	1
0	1	0	0	1	0	0	1	0
0	1	0	1	1	1	1	0	0
0	1	1	0	0	1	0	0	1
0	1	1	1	0	1	0	0	1
1	0	0	0	1	0	0	1	0
1	0	0	1	1	0	0	1	0
1	0	1	0	1	1	1	0	0
1	0	1	1	0	1	0	0	1
1	1	0	0	1	0	0	1	0
1	1	0	1	1	0	0	1	0
1	1	1	0	1	0	0	1	0
1	1	1	1	1	1	1	0	0

VHDL MODEL

```
-- Standard model for truth table using ARRAY structure.
-- This model can be directly translated into a ROM
-- implementation.
package TRUTH4x5 is
  constant NUM_OUTPUTS: INTEGER:=5;
  constant NUM_INPUTS: INTEGER:=4;
  constant NUM_ROWS: INTEGER:= 2 ** NUM_INPUTS;
  type WORD is array(NUM_OUTPUTS-1 downto 0) of BIT;
  type ADDR is array(NUM_INPUTS-1 downto 0) of BIT;
  type MEM is array (0 to NUM_ROWS-1) of WORD;
  constant TRUTH: MEM :=
                ("11100", "01001", "01001", "01001",
                 "10010", "11100", "01001", "01001",
                 "10010", "10010", "11100", "01001",
                 "10010", "10010", "10010", "11100");
  function INTVAL(VAL:ADDR) return INTEGER;
end TRUTH4x5;
package body TRUTH4x5 is
  function INTVAL(VAL: ADDR) return INTEGER is
    variable SUM: INTEGER:=0;
  begin
    for N in VAL'LOW to VAL'HIGH loop
      if VAL(N) = '1' then
        SUM := SUM + (2 ** N);
      end if;
    end loop;
    return SUM;
  end INTVAL;
end TRUTH4x5;
--  Description of COM using table lookup.
use work.TRUTH4x5.all;
architecture TABLE of COM is
begin
  process (N1,N0,M1,M0)
    variable INDEX: INTEGER;
    variable WOUT: WORD;
  begin
    INDEX := INTVAL (N1&N0&M1&M0);
    WOUT := TRUTH (INDEX);
    GE <= WOUT(4) after D;
    LE <= WOUT(3) after D;
    E  <= WOUT(2) after D;
    G  <= WOUT(1) after D;
    L  <= WOUT(0) after D;
  end process;
end TABLE;
```