Lecture 9/10: VHDL

VHDL => Very-high-speed-integrated-circuit Hardware Description Language
- Developed for US Dept of Defense in 1983
- Models IC design of target assembly board
- Two kinds of design description: structural or behavioral
- Strong emphasis on logic concurrency and timing
- Uses standard set of libraries that describe design entities and constructs
- Top-down design method employed: recursive partitioning of components into subcomponents
- Subcomponents broken down into manageable design block VHDL description

Structural description: Instantiations of modules within other modules using signal identifiers
Behavioral description: Procedural statements that determine relationship between input and output signals

Entity declaration: describes interface of logical subcomponent/block
Architectural declaration: describes structural or behavioral operation of component

```
ENTITY latch IS
        PORT(s, r : IN BIT; q, nq : OUT BIT);
END latch;

ARCHITECTURE dataflow OF latch IS                    SR LATCH
BEGIN
        q <= r NOR nq;
        nq <= s NOR q;
END dataflow;
```

Use of COMPONENT or PORT MAP declarations can be used when subcomponents already defined.
PORT/PORT MAP declarations can also take array assignments using BIT_VECTOR and DOWNTO.
Examples:

```
        (latch already defined above)
ARCHITECTURE structure OF latch IS
  COMPONENT nor_gate
        PORT( a, b: IN BIT; c: OUT BIT);
END COMPONENT;
BEGIN
        N1: nor_gate
                PORT MAP (r, nq, q);
        N2: nor_gate
                PORT MAP (s, q, nq);
END structure;
```

```
        2x1 4-line Multiplexer
ENTITY mux2 IS
  PORT( a: IN BIT_VECTOR(3 DOWNTO 0);
        b: IN BIT_VECTOR(3 DOWNTO 0);
        c: IN BIT;
        d: OUT BIT_VECTOR(3 DOWNTO 0););
END mux2;
ARCHITECTURE mux2_arch OF mux2 IS
        BEGIN
                IF c = '0' THEN
                        d <= a;
                ELSE
                        d <= b;
                END IF;
END mux2_arch;
```

Take case of serial_adder: components are full_adder, flip_flop, counter, and shifter:
How would you do component description of serial adder? Exercise: fill in PORT arguments.

Lecture 9/10: VHDL (cont.)

```
ENTITY serial_adder IS
  PORT (a, b, start, clock : IN BIT; ready: OUT BIT; result: BUFFER BIT_VECTOR(7 DOWNTO 0));
END serial_adder;
ARCHITECTURE structure OF serial_adder IS
        COMPONENT counter IS
                PORT(…);
        END COMPONENT;
        COMPONENT full_adder IS
                PORT(…);
        END COMPONENT;
        COMPONENT flip_flop IS
                PORT(…);
        END COMPONENT;
        COMPONENT shifter IS
                PORT(…);
        END COMPONENT;
END structure;
```

SIGNAL statement can also be used to make BIT/BIT_VECTOR assignments.
PROCESS statement is used to make VARIABLE declarations and sequential statements.
Example:

```
SIGNAL x: BIT_VECTOR (7 DOWNTO 0);
…
…
PROCESS (x)
        VARIABLE p : BIT;
BEGIN
        p = '0';
        FOR I IN 7 DOWNTO 0 LOOP      <========= can use NEXT statement also
                p = p XOR x(i);
        END LOOP;                     <========= with NEXT WHEN <condition> here
END PROCESS;
```

WAIT UNTIL <condition> or EXIT WHEN <condition> statements can be used also inside LOOP.
Can use CASE <expression/condition> IS

```
                WHEN case1
                …
                WHEN case2
                …
        END CASE;
```

****

Exercise: Instead of structural component description of serial_adder, do its gory behavioral description.
Hint: don't segregate architecture statement into components.

****

Functional simulation – models how design functions without timing reference
Timing simulation – model internal delays present in real-time circuits

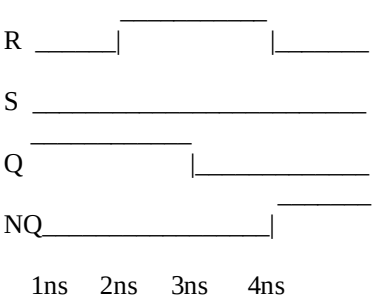
Lecture 9/10: VHDL (cont.)

Inertial delay model – timing simulation using AFTER clause in signal assignment
Transport delay model – timing simulation using AFTER clause that triggers on ANY signal state change

Example:

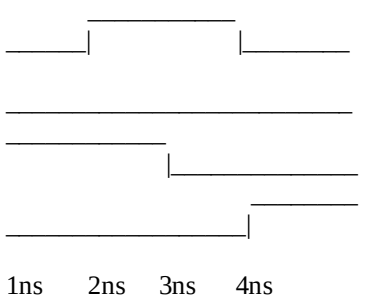
Inertial model

```
Q <= R NOR NQ AFTER 1 NS;
NQ <= S NOR Q AFTER 1 NS;
```

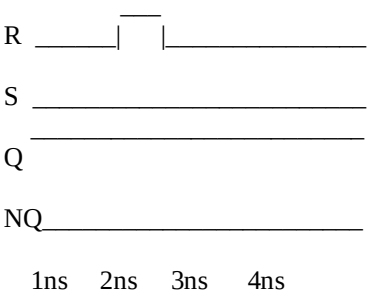


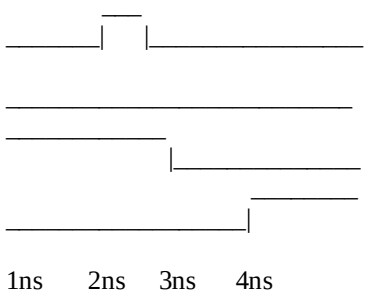
Transport model

```
Q <= TRANSPORT R NOR NQ AFTER 1 NS;
NQ <= TRANSPORT S NOR Q AFTER 1 NS;
```

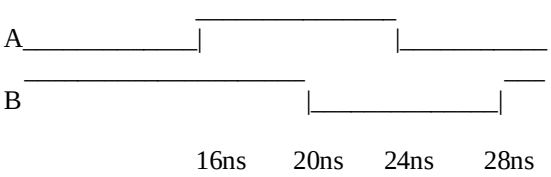


But what if R pulse is < 1ns :





Exercise: Now describe the following with VHDL.




```
ARCHITECTURE signals IS
        SIGNAL a,b : BIT;
BEGIN
        PROCESS
        BEGIN
        A <= '0', '1' AFTER 16 NS, '0' AFTER 24 NS;
        B <= '1', TRANSPORT NOT A AFTER 20 NS;
        END PROCESS;
END signals;
```

Can use subroutines in VHDL using PROCEDURE statement. Example:

```
PROCEDURE byte_to_int (ib:
IN BIT_VECTOR(7 DOWNTO 0); oi:
OUT INTEGER) IS
oi = 0;
BEGIN
        FOR i IN 0 TO 7 LOOP
                IF ib(i) = '1' THEN
                        oi = oi + 2**i;
                END IF;
        END LOOP;
END byte_to_int;
```

Exercise: use PROCEDURE statement in ARCHITECTURE signals to get same result for B using A as input.