

## ADDENDA TO NOTES 2

There are several basic ways of describing the operation of clocked flip-flops in VHDL. The following is an example.

```
entity demo is port
  (a, b, c, clk: in bit; q1, q2, q3: out bit);
end demo;
architecture ffdemo of demo is
  signal q1_next, q3_next: bit;
begin
  -- this is a sequential process block which clocks
  -- both flip-flop 1 and 2
  process(a, b, c, clk, q1_next) begin
    if (clk'event and clk='1') then
  -- flip-flops 1 and 3 loaded with the "next" value to be loaded
      q1<=q1_next;
      q3<=q3_next;
  -- flip-flop 2 is loaded with the actual next value (as shown)

      q2<=(a and b) or c;
    end if;
  end process;
  -- this is a data-flow description. It describes what the
  -- next value is to be loaded into flip-flop 1.
  q1_next<=(a and b) or c;
  -- the next value for flip-flop 3 is specified with a
  -- behavioral description.
  process(a, b, c, q3_next)
  begin
    q3_next<='0';
    if c='1' or (a='1' and b='1') then q3_next<='1'; end if;
  end process;
end ffdemo;
```

There are two basic techniques: Specifying what is to be loaded into the flip-flop within the if clk'event... statement and specifying it elsewhere. For simple operations the first method is more compact and easier. For more complicated cases (such as the flip-flops in a FSM, the later method is easier to handle. Two examples of the later case are shown. The next state for Flip-flop 1 is described in a DATA-FLOW type description (not necessarily a useful way to do it). The next state for flip-flop 3 is in a sequential behavioral statement. Note that the first line in this unconditionally sets the next value to 0. But the "sequential" nature of the process block qualifies this to say it is when under certain conditions.

What is the effect on the final synthesis? Here is part of the report file.

```
q1.D = a * b + c; q2.D = a * b + c; q3.D = a * b + c
```