Source:
Pedroni, *Circuit Design and Simulation with VHDL*, 2/e, MIT Press

GENERATE

GENERATE is a concurrent statement). It is equivalent to the sequential statement FOR/LOOP (in the sense that it allows a section of code to be repeated a number of times, thus creating several instances of
the same assignments. Its regular form is the FOR / GENERATE construct, with the syntax shown below. Notice that GENERATE must be labeled.

FOR / GENERATE:

```
label: FOR identifier IN range GENERATE
  (concurrent assignments)
END GENERATE;
```

An irregular form is also available, which uses IF/GENERATE. Here ELSE is not allowed. In the same way that IF/GENERATE can be nested inside FOR/GENERATE, the opposite can also be done.

IF / GENERATE nested inside FOR / GENERATE:

```
label1: FOR identifier IN range GENERATE
   ...
   label2: IF condition GENERATE
      (concurrent assignments)
   END GENERATE;
   ...
END GENERATE;
```

Example:

```
SIGNAL x: STD_LOGIC_VECTOR (7 DOWNTO 0);
SIGNAL y: STD_LOGIC_VECTOR (15 DOWNTO 0);
SIGNAL z: STD_LOGIC_VECTOR (7 DOWNTO 0);
...
G1: FOR i IN x'RANGE GENERATE
      z(i) <= x(i) AND y(i+8);
END GENERATE;
```

One important remark about GENERATE (and the same is true for LOOP) is that **both limits of the range must be static**.