4.3. Translate the following code to a case-when statement:

```
with state select
      data <=  "0000" when idle | terminate,
               "1111" when increase,
               "1010" when maintain,
               "0101" when decrease,
               "----" when others;
```

4.4. Translate the following code to two with-select-when statements:

```
case state is
   when idle => a <= "11"; b <= "00";
   when terminate | increase => a <= "01"; b <= "--";
   when maintain | decrease => a <= "10"; b <= "11";
   when others => a <= "11"; b <= "01";
end case;
```

4.5. Write the code to describe an 8-bit wide, four-to-one mux using a case-when statement. Inputs: a[7:0], b[7:0], c[7:0], d[7:0], s[1:0]; Outputs: q[7:0].

4.6. Write the code to compare two 8-bit buses, a and b. The output signal is agrb for "a greater than b".

4.7. Build a 4-bit magnitude comparator with three outputs (equals, less than, and greater than) using:

a. logical operators

b. relational operators

c. structural instantiation

d. a when-else statement

e. an if-then-else statement

4.8. Compile, synthesize, and simulate each version of the design in Problem 4.7.

4.9. Write the code for a 4-bit register. Inputs: clk, enable, data; output: q.

4.10. Design a 32-bit register bank using D-type flip-flops. The register bank has three-state outputs, controlled by a common output enable. Inputs: clk, register_enable, output_enable, data[31:0]; output: q[31:0].