

the reset signal are part of these sequences, the sequences must run during reset and thus “disable iff (\sim reset_l)” is not used.

9.6 Exercise Problems – Assertions

9.1 Using Problem 5.1 as a problem definition, write an assertion to check that each sequence of five bits has two and only two 1-bits.

9.2 Using Problem 5.2 as a problem definition, write the following assertions:

- A) Assert that max**Value** is 0 as a result of reset (rst).
- B) Assert that after reset start precedes done.
- C) Assert that after done is asserted, max**Value** is 0.
- D) Assert that max**Value** remains 0 after reset or done until the first non-zero value is on the inputs and the input sequence has started.
- E) Assert that done is not asserted while start is asserted.
- F) Assert that done is asserted in the state following start becoming not-asserted.
- G) Assert that max**Value** is correct based on the inputs that the assertion statement sees.

9.3 Using Problem 5.3 as a problem definition, write the following assertions.

- A) Assert that done and error are never asserted together.
- B) Assert that error is asserted at the correct time.
- C) Assert that error stays asserted until the next start.

9.4 Using Problem 5.4 as a problem definition, write the following assertions.

- A) Assert that c.**Free** is the first signal to be asserted after reset (i.e., p.**put** isn't first).
- B) Assert that the c.**Free** and p.**put** operate in the correct order.