Answer 3.1:

When we use blocking statements, all right hand side values get evaluated concurrently but no assignment takes place in the current simulation cycle. Assignment will always take place in the next cycle. So, even if the order of the statements changes, unless the value of 'in' is assigned to 'x', the right hand side evaluation of y and z will always be unknown. Similarly, unless 'y' has been assigned a value, z will be unknown. Hence, re-arrangement of statements will not affect the result shown in the example.

Answer 3.3:

Synchronous reset depends on the clock signal and ensures operations in sequential circuits take place at a known time. It can be synchronized to any edge of the clock. It provides a more controlled and timed design and helps avoid metastability issues.

Asynchronous reset is not synchronized to the clock and can occur at any point of time. It is useful when a quick and immediate reset of the system is required. May create metastability issues and needs careful handling.

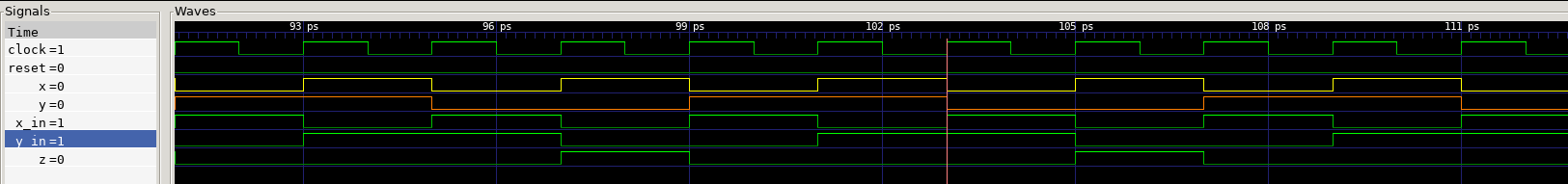
Answer 3.3:

A paper with writing on it

Description automatically generated

With blocking assignments, ‘x’, ‘y’ and ‘z’ get incremented in the same cycle. Which is why we do not have pipelined increments and rather a single stage operation.

Answer 3.4:



The waveform generated is as expected. For every positive edge of clock, input values of x and y are registered in ‘x\_in’ and ‘y\_in’ in every next cycle and ‘z’ gets assigned the new AND gate output in the following cycle since it uses ‘x\_in’ and ‘y\_in’. This way, all inputs and outputs of AND gate gets registered.