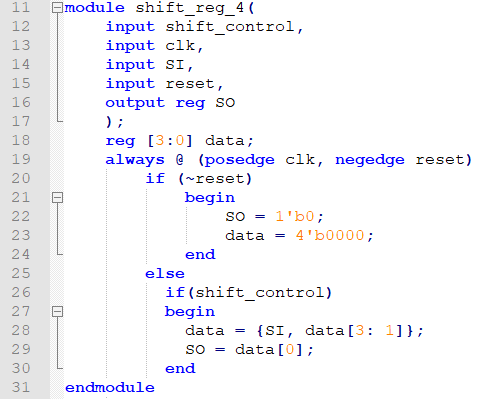
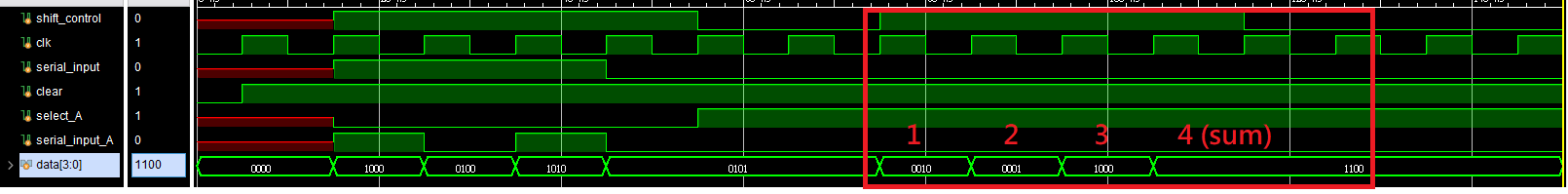


* Blocking實現shift right

程式碼如下。當採用「=」，表示當等號右邊的值改變，會立刻在”同一個”clock cycle反應。例如暫存器A (0101) 和暫存器B (0111) 右移最低位元 1和 1時，再搭配前一級進位Ci = 0可以得到做XOR運算結果0 (S)，並將S值直接補至暫存器A最左位元使其變成0010。因此，暫存器A更新四次後即可得到和的結果為1100。此為課本說明串列加法器的方式，但前提是在shift的方式採用blocking的做法，所以暫存器A變化為0010, 0001, 1000, 1100。





* Non-blocking實現shift right

程式碼如下。當採用「<=」，表示當等號右邊的值改變，會”下一個”clock cycle反應。例如暫存器A (0101) 和暫存器B (0111) 右移最低位元 1和 1時，再搭配前一級進位Ci = 0可以得到做XOR運算結果0 (S)，並於”下一個”clock cycle時將此S值補至暫存器A最左位元。因此，暫存器A最左位元第1次補的是S的初始值(0)，因此暫存器A第1次變成0010。暫存器A最左位元第2次補才是前一個clock cycle做XOR運算得到的S值(0)，因此暫存器A第2次變成0001，再以此類推暫存器A值的變化。因為有一個clock cycle時間差，暫存器A更新五次後可得到和的結果1100。因此，若shift的方式採用non-blocking的做法時，暫存器A變化為為0010, 0001, 0000, 1000, 1100。

