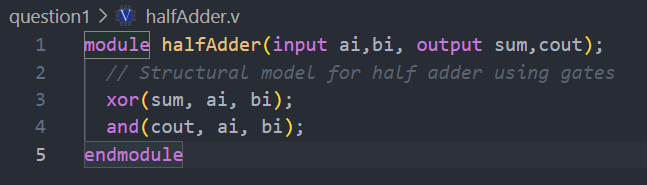


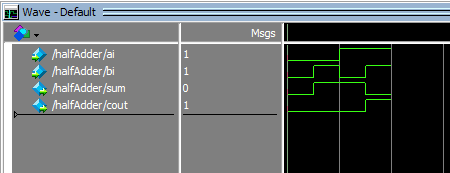
# Half adder (a) using gates (b) using dataflow model (using assign statement).

## Solution:

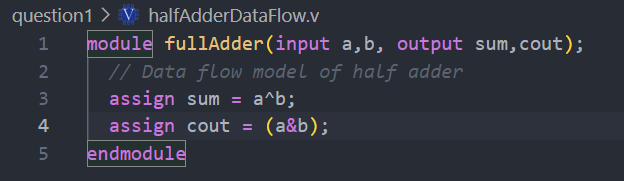
### Verilog code using gates



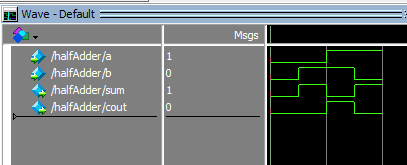
### Simulation waveform using gates



### Verilog code using dataflow model



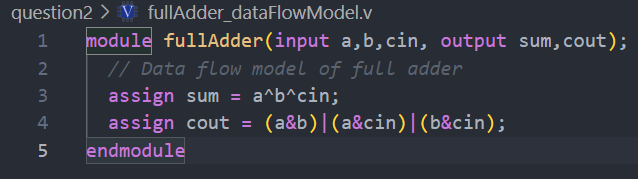
### Simulation waveform using dataflow model



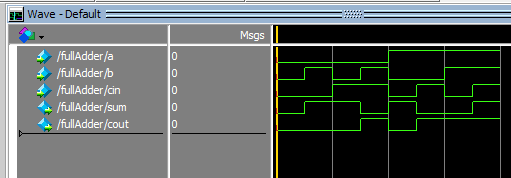
# Full adder (a) using dataflow model (using assign statement) (b) using two half adders and an OR gate.

## Solution:

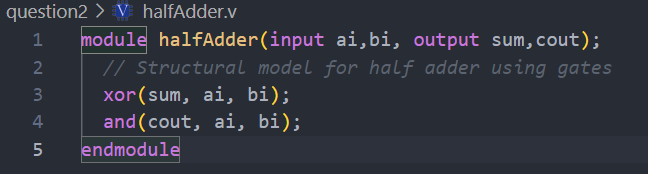
### Verilog code using dataflow model

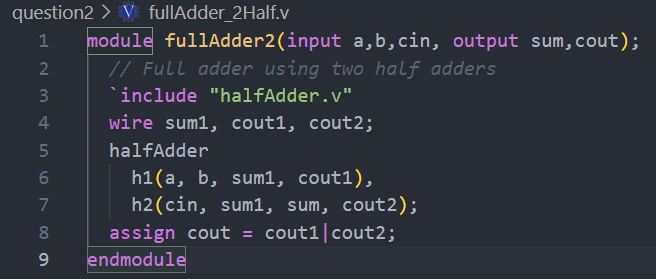


### Simulation waveform using dataflow model

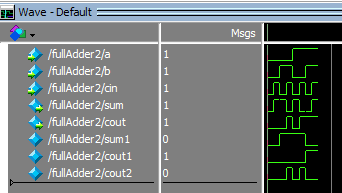


### Verilog code





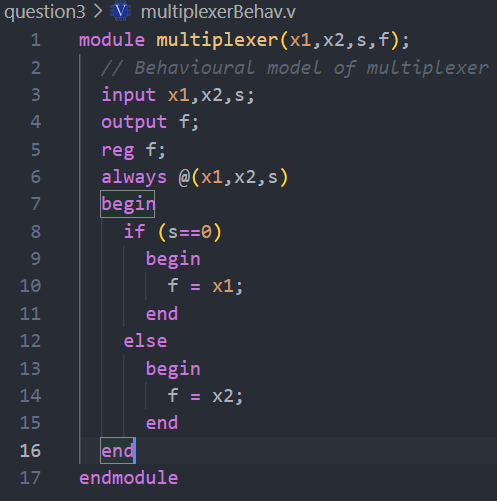
### Simulation waveform



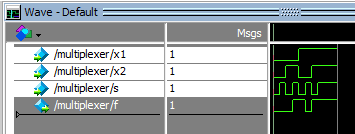
# Two to one multiplexer using behavioral model (using if statement).

## Solution:

### Verilog code



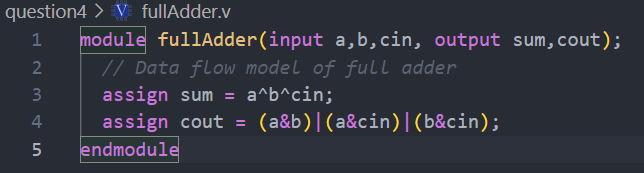
### Simulation waveform

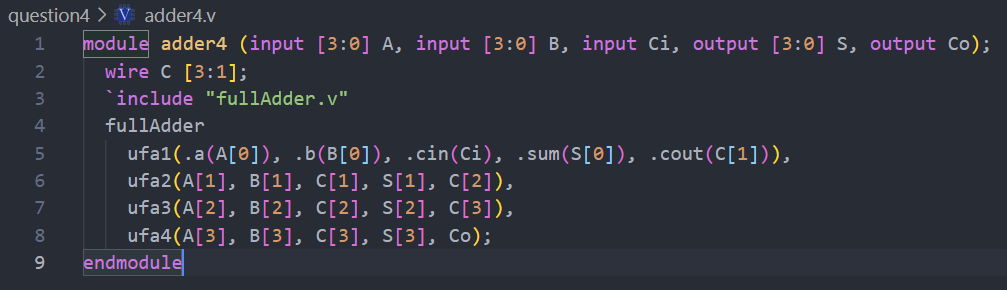


# 4-bit ripple carry adder using full adder modules.

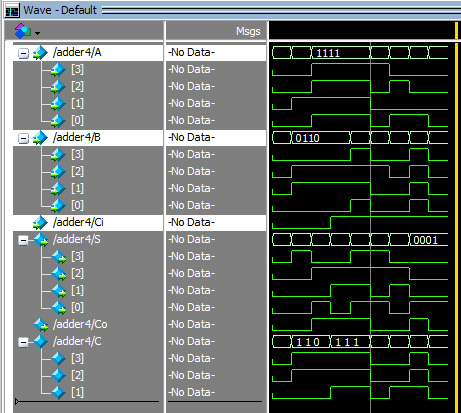
## Solution:

### Verilog code





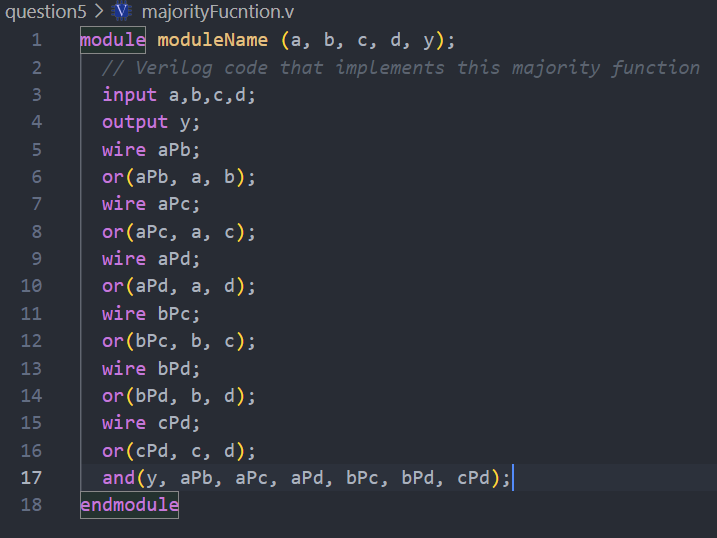
### Simulation waveform



# A four variable logic function that is equal to 1 if any three or all four of its variables are equal to 1 is called a majority function. Write a Verilog code that implements this majority function. Use the Boolean equation derived in Lab1 in assign statement.

## Solution:

### Verilog code



### Simulation waveform

