**Stimulus Generator:**  
In order to test the model of some design, a verification engineer must apply test patterns to the input ports and observe the output ports over time to decide whether the inputs were transformed to the expected outputs. The generator component generates input vectors. For simple memory stimulus generator generates read, write operations, address and data to be stored in the address if its write operation. Modern generators generate random, biased, and valid stimuli. In verilog $random does this job. The randomness is important to achieve a high distribution over the huge space of the available input stimuli. To this end, users of these generators intentionally under-specify the requirements for the generated tests. It is the role of the generator to randomly fill this gap. This mechanism allows the generator to create inputs that reveal bugs not being searched for directly by the user. Generators also bias the stimuli toward design corner cases to further stress the logic. Biasing and randomness serve different goals and there are tradeoffs between them, hence different generators have a different mix of these characteristics. Since the input for the design must be valid and many targets should be maintained, many generators use the Constraint Satisfaction Problem technique to solve the complex testing requirements. SystemVerilog, Vera, SystemC and Specman have " constraints " to specify The legality of the design inputs. In verilog ,to constrain the memory address to be between 0 to 63, {$random} % 64 is used. The model-based generators use this model to produce the correct stimuli for the target design. The stimulus generator should be intelligent and easily controllable.  
  
  
  
**Bus Functional Models :**  
  
The Bus Functional Model (BFM) for a device interacts with the DUT by both driving and sampling the DUT signals. A bus functional model is a model that provides a task or procedural interface to specify certain bus operations for a defined bus protocol. For a memory DUT, transactions usually take the form of read and write operations. Bus functional models are easy to use and provide good performance. It has to follow the timing protocol of the DUT interface. BFM describes the functionality and provides a cycle accurate interface to DUT. It models external behavior of the device. For re usability, the implementation of the BFM functionality should be kept as independent of the communication to the BFM as it can be.  
  
  
**Driver :**  
  
Driver is a types of BFM. The drivers translate the stimuli produced by the generator into the actual inputs for the design under verification. Generators create inputs at a high level of abstraction; namely, as transactions like read write operation. The drivers convert this input into actual design inputs which is at a low level like bits ,as defined in the specification of the designs interface. If the generator generates read operation, then read task is called, in that, the DUT input pin "read\_write" is asserted.

Graphical user interface, text, application

Description automatically generated