**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **05-06-2020** | **Name:** | **Bhavith** |
| **Course:** | **Digital design using HDL** | **USN:** | **4AL17EC009** |
| **Topic:** | **Programming in verilog** | **Semester & Section:** | **6th,A** |
| **Github Repository:** | **Bhavith-Online-Courses** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **Image of session**  **PSX_20200606_083048** |
| **Report – Report can be typed or hand written for up to two pages.**   * **It is certainly possible to represent hardware semantics using traditional programming languages such as [C++](https://en.wikipedia.org/wiki/C++" \o "C++), which operate on [control flow](https://en.wikipedia.org/wiki/Control_flow" \o "Control flow) semantics as opposed to [data flow](https://en.wikipedia.org/wiki/Data_flow" \o "Data flow), although to function as such, programs must be augmented with extensive and unwieldy [class libraries](https://en.wikipedia.org/wiki/Class_library" \l "Object_and_class_libraries" \o "Class library).** * **Generally, however, software programming languages do not include any capability for explicitly expressing time, and thus cannot function as hardware description languages.** * **Before the introduction of [System Verilog](https://en.wikipedia.org/wiki/System_Verilog" \o "System Verilog) in 2002, [C++](https://en.wikipedia.org/wiki/C++" \o "C++) integration with a [logic simulator](https://en.wikipedia.org/wiki/Logic_simulation" \o "Logic simulation) was one of the few ways to use [object-oriented programming](https://en.wikipedia.org/wiki/Object-oriented_programming" \o "Object-oriented programming) in hardware verification.** * **System Verilog is the first major HDL to offer object orientation and garbage collection.** * **Using the proper subset of hardware description language, a program called a synthesizer, or [logic synthesis tool](https://en.wikipedia.org/wiki/Logic_synthesis" \o "Logic synthesis), can infer hardware logic operations from the language statements and produce an equivalent netlist of generic hardware primitives[*[jargon](https://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style" \l "Technical_language" \o "Wikipedia:Manual of Style)*] to implement the specified behaviour.** * **Synthesizers generally ignore the expression of any timing constructs in the text. Digital logic synthesizers, for example, generally use [clock edges](https://en.wikipedia.org/wiki/Clock_signal" \o "Clock signal) as the way to time the circuit, ignoring any timing constructs.** * **The ability to have a synthesizable subset of the language does not itself make a hardware description language.** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date:** | **05-06-2020** | **Name:** | **Bhavith** | |
| **Course:** | **Python** | **USN:** | **4Al17EC009** | |
| **Topic:** | **Imported Modules** | **Semester & Section:** | **6th,A** | |
| **AFTERNOON SESSION DETAILS** | | | |
| **Image of session**  **Screenshot (139)** | | | |
| **Report – Report can be typed or hand written for up to two pages.**   * **Python code in one [module](https://docs.python.org/3/glossary.html" \l "term-module) gains access to the code in another module by the process of [importing](https://docs.python.org/3/glossary.html" \l "term-importing) it.** * **The [import](https://docs.python.org/3/reference/simple_stmts.html" \l "import) statement is the most common way of invoking the import machinery, but it is not the only way. Functions such as [importlib.](https://docs.python.org/3/library/importlib.html" \l "importlib.import_module" \o "importlib.import_module)** * **[import\_module()](https://docs.python.org/3/library/importlib.html" \l "importlib.import_module" \o "importlib.import_module) and built-in [\_\_import\_\_()](https://docs.python.org/3/library/functions.html" \l "__import__" \o "__import__) can also be used to invoke the import machinery.** * **The [import](https://docs.python.org/3/reference/simple_stmts.html" \l "import) statement combines two operations; it searches for the named module, then it binds the results of that search to a name in the local scope.** * **The search operation of the import statement is defined as a call to the [\_\_import\_\_()](https://docs.python.org/3/library/functions.html" \l "__import__" \o "__import__) function, with the appropriate arguments.** * **The return value of [\_\_import\_\_()](https://docs.python.org/3/library/functions.html" \l "__import__" \o "__import__) is used to perform the name binding operation of the import statement.** * **See the import statement for the exact details of that name binding operation.** * **A direct call to [\_\_import\_\_()](https://docs.python.org/3/library/functions.html" \l "__import__" \o "__import__) performs only the module search and, if found, the module creation operation.** * **While certain side-effects may occur, such as the importing of parent packages, and the updating of various caches (including [sys.modules](https://docs.python.org/3/library/sys.html" \l "sys.modules" \o "sys.modules)), only the [import](https://docs.python.org/3/reference/simple_stmts.html" \l "import) statement performs a name binding operation.** * **When an [import](https://docs.python.org/3/reference/simple_stmts.html" \l "import) statement is executed, the standard builtin [\_\_import\_\_()](https://docs.python.org/3/library/functions.html" \l "__import__" \o "__import__) function is called. Other mechanisms for invoking the import system (such as [importlib.import\_module()](https://docs.python.org/3/library/importlib.html" \l "importlib.import_module" \o "importlib.import_module)) may choose to bypass [\_\_import\_\_()](https://docs.python.org/3/library/functions.html" \l "__import__" \o "__import__) and use their own solutions to implement import semantics.** | | | |