Change as shown in red Section 7.10.1 – In Built-in methods section

7.10.1 Built-in ~~namespace~~ package

SystemVerilog provides a built-in ~~namespace~~ package that contains system types (e.g., classes), variables, tasks, and functions. ~~The builtin namespace resides at the top of the hierarchy.~~ Users may not insert additional declarations into the built-in ~~namespace~~ package. The built-in package is implicitly wildcard imported into the compilation-unit scope of every compilation unit (see Section 18.3). ~~The~~ Thus, declarations in the built-in ~~namespace~~ package are directly available in any scope~~,~~ (like system tasks and functions~~,~~) ~~and can be~~ unless they are redefined by user code. ~~in any other scope. However, unlike system tasks and functions, the tasks and functions in the built-in namespace may not be redefined by PLI functions.~~

built_in_data_type ::= [**std ::**] data_type_identifier *// Not in Annex A*

built_in_function_call ::= [**std ::**] built_in_identifier [**(** list_of_arguments **)]**

The package name **std** followed by the scope resolution operator `::` ~~with no identifier on the left~~ can be used to unambiguously access names in the built-in ~~namespace~~ package. For example:

```
    std::sys_task();        // unambiguously call the system provided sys_task
```

Unlike system tasks and functions, tasks and functions in the built-in ~~namespace~~ package need not be prefixed with a **$** to avoid collisions with user-defined identifiers. This mechanism allows functional extensions to the language in a backward compatible manner, without the addition of new keywords or polluting local name spaces.