CHAPTER 1 CSL Memory

TABLE 1.1 Chapter Outline

CSL Memory

can be defined and instantiated

- can be defined in

TABLE 1.2 CSL unit definition in other CSL classes

CSL class	Can be defined in
global scope	YES
CSL Unit	-
CSL Signal Group	-
CSL Interface	-
CSL Testbench	-
CSL Vector	-
CSL State Data	-
CSL Register	-
CSL Register File	-
CSL Fifo	-
CSL Memory Map	-
CSL Memory Map Page	-
CSL Isa Element	-
CSL Isa Field	-
CSL Field	-

- can be instantiated in

TABLE 1.3 CSL unit instantiation in other CSL classes

CSL class	Can be instantiated in
global scope	-
CSL Unit	YES
CSL Signal Group	-
CSL Interface	-
CSL Testbench	YES
CSL Vector	-
CSL State Data	-
CSL Register	-
CSL Register File	-
CSL Fifo	-

TABLE 1.3 CSL unit instantiation in other CSL classes

CSL class	Can be instantiated in
CSL Memory Map	-
CSL Memory Map Page	-
CSL Isa Element	-
CSL Isa Field	-
CSL Field	-

1.1 CSL Memory Command Summary

CSL Memory: Mandatory commands

```
set_width(numeric_expression);
set_depth(numeric_expression);
```

CSL Memory

```
csl_memory memory_name;
initialize_random_values(seed_value);
initialize_random(value);
set_num_rd_ifc(numeric expression);
set_num_wr_ifc(numeric expression);
add_logic(reset);
add_logic(clock);
```

1.2 CSL Memory Commands

CSL Memory : mandatory commands

```
set_width(numeric_expression);
```

DESCRIPTION :

Creates a Memory named *rmemory_name.* The param width and depth there are numeric expresion and are mandatory for Memory.

[CSL Register File Command Summary]

EXAMPLE :

```
set_depth(numeric_expression);
```

DESCRIPTION :

Creates a Memory named memory_*name.* The param width and depth there are numeric expresion and are mandatory for Memory.

EXAMPLE :

CSL Memory

```
csl_memory memory_name;
```

DESCRIPTION :

Creates a memory named *memory_name* .

[CSL Memory Command Summary]

EXAMPLE :

//

CSL CODE

```
csl_memory mem {
mem(){
set_width(8);
set_depth(128);
}
```

VERILOG CODE

```
//
```

```
initialize(filename);
```

DESCRIPTION :

Use the values in the file to initialize a memory in the C++ simulator or a DUT memory instance in the test bench.

[CSL Memory Command Summary]

EXAMPLE :

//

CSL CODE

```
    csl_memory mem {
      mem(){
        set_width(8);
        set_depth(64);
        initialize("mem_file");
      }
    };
```

VERILOG CODE

```
    //
```

initialize_random_values(seed_value);

DESCRIPTION :

Initialize each memory location in a memory with random values in the C++ simulator or a DUT memory instance in the test bench. Use the *seed_value* to initialize the random number generator.

[CSL Memory Command Summary]

EXAMPLE :

//

CSL CODE

```
csl_memory mem{
  mem(){
    set_width(8);
    set_depth(64);
    initialize_random_values(8);
  }
};
```

VERILOG CODE

```
//
```

```
initialize_random(value);
```

DESCRIPTION :

Use the same value to initialize each memory location in the memory in the C++ simulator or a DUT memory instance in the test bench.

[CSL Memory Command Summary]

EXAMPLE :

//

CSL CODE

```
    csl_memory mem{
      mem(){
        set_width(8);
        set_depth(64);
        initialize_random(16);
      }
    };
```

VERILOG CODE

```
    //
```

set_num_rd_ifc(numeric expression);

DESCRIPTION :

Sets the number of the read interfaces generated in the memory unit.

[CSL Memory Command Summary]

EXAMPLE :

//

CSL CODE

```
    csl_memory mem{
      mem(){
        set_width(8);
        set_depth(128);
        set_num_rd_ifc(2);
      }
    };
```

VERILOG CODE

```
    //
```

set_num_wr_ifc(numeric expression);

DESCRIPTION :

Sets the number of the write interfaces generated in the memory unit.

[CSL Memory Command Summary]

EXAMPLE :

//

CSL CODE

```
    csl_memory mem {
      mem(){
        set_width(8);
        set_depth(64);
        set_num_wr_ifc(2);
      }
    };
```

VERILOG CODE

```
    //
```

`add_logic(reset);`

DESCRIPTION :

This is an reset command. When the reset signal is on the low level (logic “0“) the memory is filled up with 0 values, the clock signal edge doesn’t matter.

[CSL Memory Command Summary]

EXAMPLE :

//

CSL CODE

```
csl_memory mem {
  mem(){
    set_width(8);
    set_depth(64);
    add_logic(reset);
  }
};
```

VERILOG CODE

```
//
```

`add_logic(clock);`

DESCRIPTION :

Adds a clock signal to a memory.

[CSL Memory Command Summary]

EXAMPLE :

//

CSL CODE

```
csl_memory mem {
  mem(){
    set_width(8);
    set_depth(64);
    add_logic(clock);
  }
};
```

VERILOG CODE

```
//
```