CHAPTER 2 CSL Register File

TABLE 2.1 Chapter Overview

2.1 CSL Register File Command Summary
2.2 CSL Register File Commands

2.1 CSL Register File Command Summary

2.1.1 Usage tables

CSL Register file
can be defined and instantiated
- can be defined in

TABLE 2.2 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 2.3 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	-

TABLE 2.3 CSL unit instantiation in other CSL classes

CSL class	Can be instantiated in
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	-

REGISTER FILE: Mandatory commands

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

REGISTER FILE: Signal

```
csl_register_file register_file_name;
add_logic(read_valid);
```

REGISTER FILE:

```
create_rtl_module();
```

REGISTER FILE: Custom port naming

```
set_reset_name(string);
set_clock_name(string);
set_wr_data_name(string);
set_rd_data_name(string);
set_wr_addr_name(string);
set_rd_addr_name(string);
set_wr_en_name(string);
set_rd_en_name(string);
set_valid_name(string);
```

2.2 CSL Register File Commands

```
set_width(numeric_expression);
```

DESCRIPTION :

Creates a Register file named *register_file_name.* The param width and depth there are numeric expresion and are mandatory for Register file.

[CSL Register File Command Summary]

EXAMPLE :

set_depth(numeric_expression);

DESCRIPTION :

Creates a Register file named *register_file_name.* The param width and depth there are numeric expresion and are mandatory for Register file.

[CSL Register File Command Summary]

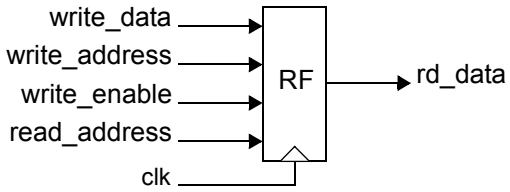
EXAMPLE :

csl_register_file *register_file_name*;

DESCRIPTION :

Declares a new register file with the name *register_file_name.*

FIGURE 2.1 Register file with no special options



[CSL Register File Command Summary]

EXAMPLE :

Creates a register file named *RF* and is set the width and depth.

CSL CODE

```
csl_register_file RF{
RF(){
set_width(32);
set_depth(4);
}
};
```

VERILOG CODE

```
module register_file(clk, reset, rf_wr_data_in, rf_wr_addr, rf_rd_en,
rf_wr_en, rf_rd_addr, rf_data_out);
parameter D_WIDTH=32;
parameter A_WIDTH=16;
input [D_WIDTH - 1: 0] rf_wr_data_in;
input [A_WIDTH - 1: 0] rf_wr_addr;
input rf_rd_en, rf_wr_en, clk, reset ;
output [D_WIDTH - 1: 0] rf_data_out;
input [A_WIDTH - 1: 0] rf_rd_addr;
reg [D_WIDTH - 1: 0] rf [A_WIDTH - 1: 0], rf_data_out;
integer i;
integer j;
   always @ (posedge clk) begin
      if(!reset) begin
         for(i = 0; i < A_WIDTH; i=i+1) begin
              rf[i] = {{D_WIDTH}, 1'b0};
           end
      end
   end
```

```
   always @ (posedge clk) begin
      if(rf_wr_en) begin
         rf[rf_wr_addr] <= rf_wr_data_in;
      end
      if(rf_rd_en) begin
       rf_data_out <= rf[rf_rd_addr];
      end
   end
endmodule
```

```
add_logic(read_valid);
   port: output - valid_<register_file>
```

DESCRIPTION :

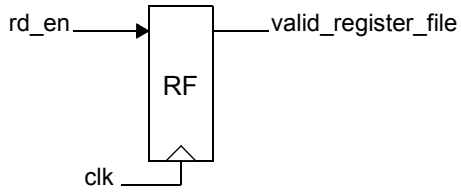
Create an output port for the register file's valid bit. The valid bit is the read enable delayed with 1 clock cycle and is used in pipelines.

[CSL Register File Command Summary]

EXAMPLE :

In this example it is created an output port for valid bit of a register file named RF.

FIGURE 2.2 Register file with valid bit



CSL CODE:

```
csl_register_file RF{
RF(){
set_width(32);
set_depth(4);
add_logic(read_valid);
}
};
```

VERILOG CODE:

```
//to be changed
module
RF(wr_data,wr_address,rd_data,rd_address,clk,wr_en,validation_bits);
input wr_en;
input [7:0] wr_data;
input clk;
input [1:0] wr_address;
input [1:0] rd_address;
output [7:0] rd_data;
output [3:0]validation_bits;
reg [7:0] rd_data;
reg [3:0] validation_bits;
reg [7:0]reg_file[3:0];
integer i;
integer j;
```

```
//initilize register file
initial begin
for(j=0;j<5;j=j+1)
begin
    reg_file[j]=8'b0;
end
end
always@(posedge clk)
begin
if(wr_en) begin
   reg_file[wr_address]=wr_data;
end
end
always@(posedge clk)
begin
    rd_data=reg_file[rd_address];
end
always@(posedge clk)
begin

    if(wr_en)
     begin
         validation_bits[wr_address]=1'b0;
        for(i=0;i<5;i=i+1)
        begin
        if(i!=wr_address)
        begin
        validation_bits[i]=1'b1;
    end
    end
end
    else
    begin
        for(i=0;i<5;i=i+1)
        begin
        validation_bits[i]=1'b1;
    end
    end
end
endmodule
```

```
create_rtl_module();
```

DESCRIPTION :

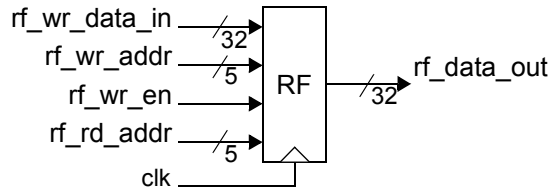
Creates a RTL module from a previous declared register file with the name *module_name* and put it in a file named *rtl_language_extension*.

[CSL Register File Command Summary]

EXAMPLE :

Creates a RTL module for the register file RF.

FIGURE 2.3



CSL CODE

```
csl_register_file RF{
RF(){
set_width(32);
set_depth(4);
create_rtl_module();
}
};
```

VERILOG CODE

`set_reset_name(string);`

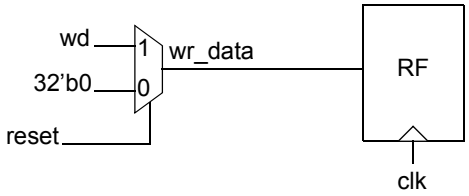
DESCRIPTION :

It sets the *name* for the reset port of register_file.

EXAMPLE :

Sets the name “rst” for reset.

FIGURE 2.4



CSL CODE

```
csl_register_file RF{
RF(){
set_width(32);
set_depth(4);
set_reset_name("rst");
}
};
```

VERILOG CODE

set_clock_name(string);

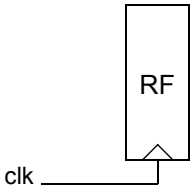
DESCRIPTION :

It sets the *name* for the clock port of register_file.

EXAMPLE :

Sets the name “clk” for the clock.

FIGURE 2.5



CSL CODE

```
csl_register_file RF{
RF(){
set_width(32);
set_depth(4);
set_clock_name("clk");
}
};
```

VERILOG CODE

set_wr_data_name(string);

DESCRIPTION :

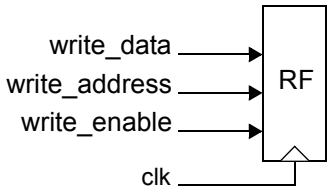
It sets the *name* for the wr_data port of register_file.

[CSL Register File Command Summary]

EXAMPLE :

Sets the name “write_data” for wr_data port of register file RF.

FIGURE 2.6



CSL CODE

```
csl_register_file RF{
RF(){
set_width(32);
set_depth(4);
set_wr_data_name("write_data");
}
};
```

VERILOG CODE

set_rd_data_name(string);

DESCRIPTION :

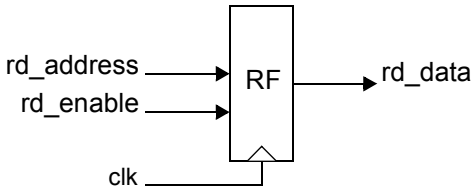
It sets the *name* for the rd_data port of register_file.

[CSL Register File Command Summary]

EXAMPLE :

Sets the name “read_data” for rd_data port of register file RF.

FIGURE 2.7



CSL CODE

```
csl_register_file RF{
RF(){
set_width(32);
set_depth(4);
set_rd_data_name("read_data");
}
};
```

VERILOG CODE

set_wr_addr_name(string);

DESCRIPTION :

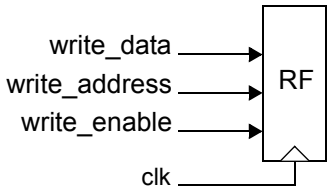
It sets the *name* for the wr_addr port of register_file.

[CSL Register File Command Summary]

EXAMPLE :

Sets the name “write_address” for wr_addr port of register file RF.

FIGURE 2.8



CSL CODE

```
csl_register_file RF{
RF(){
set_width(32);
set_depth(4);
set_wr_addr_name("write_address");
}
};
```

VERILOG CODE

set_rd_addr_name(string);

DESCRIPTION :

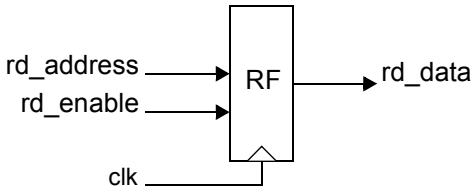
It sets the *name* for the rd_addr port of register_file.

[CSL Register File Command Summary]

EXAMPLE :

Sets the name “read_address” for rd_addr port of register file RF.

FIGURE 2.9



CSL CODE

```
csl_register_file RF{
RF(){
set_width(32);
set_depth(4);
set_rd_addr_name("read_address");
}
};
```

VERILOG COD

```
set_wr_en_name(string);
```

DESCRIPTION :

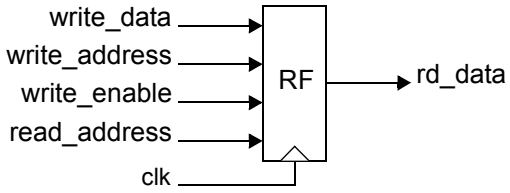
It sets the *name* for the wr_en port of register_file.

[CSL Register File Command Summary]

EXAMPLE :

Sets the name “write_enable” for wr_en port of register file RF.

FIGURE 2.10



CSL CODE

```
csl_register_file RF{
RF(){
set_width(32);
set_depth(4);
set_wr_en_name("write_enable");
}
};
```

VERILOG COD

```
set_rd_en_name(string);
```

DESCRIPTION :

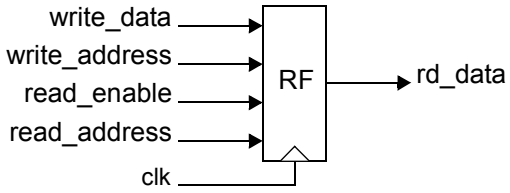
It sets the *name* for the rd_en port of register_file.

[CSL Register File Command Summary]

EXAMPLE :

Sets the name “read_enable” for rd_en port of register file RF.

FIGURE 2.11



CSL CODE

```
csl_register_file RF{
RF(){
set_width(32);
set_depth(4);
set_rd_en_name("read_enable");
 }
};
```

VERILOG CODE

```
set_valid_name(string);
```

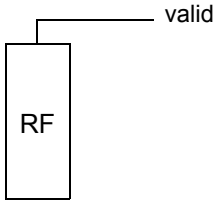
DESCRIPTION :

It sets the *name* for the valid port of register_file.

EXAMPLE :

Sets the name “vld” for valid port of register file RF.

FIGURE 2.12



CSL CODE

```
csl_register_file RF{
RF(){
set_width(32);
set_depth(4);
set_valid_name("vld");
}
};
```

VERILOG CODE