

## Verilog Code Format

- 1-) Top module name and the file name should be the same.
- 2-) Testbench of a module should be named as moduleName\_tb (extended with \_tb).
- 3-) Inside begin/end, module/endmodule, case/endcase parts should be indented.

```
always @(posedge clk)
begin
    if (!rst)
    begin
        valid1 <= 1'b0;
    end
    else
    begin
        valid1 <= valid0;
    end
end
```

- 4-) Module declarations should be Ansi-C style Verilog-2001.

```
module module_name #(
    parameter SIZE = 8
)
(
    input  clk,
    input  rst,
    input  [SIZE-1:0] rdata0,
    input  [SIZE-1:0] rdata1,
    output [SIZE-1:0] wdata0,
    output [SIZE-1:0] wdata1
);
...
endmodule
```

- 5-) Parameter names should be uppercase and signal names should be lowercase letters.

- 6-) Instantiation of a module should be done in multiple lines not a single line, only one signal assignment should be done in each line.

```
module_name #(
    .SIZE(16)
) instance_name
(
    .clk (clk),
    .rst (1'b0),      // connect unnecessary inputs to the ground (logic-0)
    .rdata0 (rdata),
    .rdata1 (16'b0),
    .wdata0 (wdata),
    .wdata1 ()        // leave empty unconnected outputs
);
```

- 7-) For instantiations, use connecting by name method instead of connecting by order method mostly.

- 8-) Module names, signal names and parameter names should be understandable. Understandable long names should be preferred to short names.

- 9-) Comments should be used to make the code more understandable.

## Verilog Kod Formatı

1-) Üst Modül ismi ile dosya ismi aynı olmalıdır.

2-) Bir modülün test kodu modül\_ismi\_tb olarak isimlendirilmelidir (sona \_tb eklenmiş).

3-) begin/end, module/endmodule, case/endcase içinde kalan bölümler için girinti eklenmelidir.

```
always @(posedge clk)
begin
    if (!rst)
    begin
        valid1 <= 1'b0;
    end
    else
    begin
        valid1 <= valid0;
    end
end
```

4-) Modül tanımlamaları Ansi-C style Verilog-2001 şeklinde olmalıdır.

```
module module_name #(
    parameter SIZE = 8
)
(
    input  clk,
    input  rst,
    input  [SIZE-1:0] rdata0,
    input  [SIZE-1:0] rdata1,
    output [SIZE-1:0] wdata0,
    output [SIZE-1:0] wdata1
);
...
endmodule
```

5-) Parametre isimleri büyük harf, sinyal isimleri küçük harf ile ifade edilmelidir.

6-) Modül örneklemesi tek satırda değil her satırda bir sinyal ataması olacak şekilde çoklu satırda yapılmalıdır.

```
module_name #(
    .SIZE(16)
) instance_name
(
    .clk (clk),
    .rst (1'b0),           // connect unnecessary inputs to the ground (logic-0)
    .rdata0 (rdata),
    .rdata1 (16'b0),
    .wdata0 (wdata),
    .wdata1 ()             // leave empty unconnected outputs
);
```

7-) Modül örneklemesi için isimle çağırma (connecting by name) yöntemi, sırayla çağırma (connecting by order) yöntemine göre daha sık tercih edilmelidir.

8-) Modül, sinyal, parametre isimleri anlaşılır olmalıdır. Uzun ve anlaşılır isimler, kısa isimlere göre tercih edilmelidir.

9-) Yorum satırları kodu daha anlaşılır hale getirmek için kullanılabilir.