

```

1  module moor_with_default (input clk, rst, j, output w);
2      reg [2:0] ns, ps;
3
4      always @(ps,j) begin
5          ns = 3'b000;
6          case(ps)
7              3'b000: ns = j ? 3'b001 : 3'b000;
8              3'b001: ns = j ? 3'b001 : 3'b010;
9              3'b010: ns = j ? 3'b011 : 3'b000;
10             3'b011: ns = j ? 3'b100 : 3'b010;
11             3'b100: ns = j ? 3'b001 : 3'b010;
12             default: ns = 3'b000;
13         endcase
14     end
15
16     assign w = (ps == 3'b100) ? 1'b1 : 1'b0;
17
18     always @(posedge clk, posedge rst) begin
19         if (rst)
20             ps <= 3'b000;
21         else
22             ps <= ns;
23     end
24 endmodule

```

```

591  === moor_with_default ===
592
593      Number of wires:                16
594      Number of wire bits:            20
595      Number of public wires:         5
596      Number of public wire bits:     9
597      Number of memories:             0
598      Number of memory bits:          0
599      Number of processes:            0
600      Number of cells:                16
601      $ _ANDNOT_                      1
602      $ _AND_                          2
603      $ _AOI3_                         2
604      $ _DFF_PP0_                     4
605      $ _DFF_PP1_                     1
606      $ _NOR_                          3
607      $ _NOT_                          3

```

```

1  module moor_without_default (input clk, rst, j, output w);
2      reg [2:0] ns, ps;
3
4      always @(ps,j) begin
5          ns = 3'b000;
6          case(ps)
7              3'b000: ns = j ? 3'b001 : 3'b000;
8              3'b001: ns = j ? 3'b001 : 3'b010;
9              3'b010: ns = j ? 3'b011 : 3'b000;
10             3'b011: ns = j ? 3'b100 : 3'b010;
11             3'b100: ns = j ? 3'b001 : 3'b010;
12         endcase
13     end
14
15     assign w = (ps == 3'b100) ? 1'b1 : 1'b0;
16
17     always @(posedge clk, posedge rst) begin
18         if (rst)
19             ps <= 3'b000;
20         else
21             ps <= ns;
22     end
23 endmodule

```

```

591  === moor_without_default ===
592
593      Number of wires:                16
594      Number of wire bits:            20
595      Number of public wires:         5
596      Number of public wire bits:     9
597      Number of memories:             0
598      Number of memory bits:         0
599      Number of processes:            0
600      Number of cells:                16
601      $ _ANDNOT_                      1
602      $ _AND_                          2
603      $ _A0I3_                         2
604      $ _DFF_PP0_                     4
605      $ _DFF_PP1_                     1
606      $ _NOR_                          3
607      $ _NOT_                          3

```

```
591    === moor_without_default ===
```

```
591    === moor_with_default ===
```

592			16
593	Number of wires:	16	20
594	Number of wire bits:	20	5
595	Number of public wires:	5	9
596	Number of public wire bits:	9	0
597	Number of memories:	0	0
598	Number of memory bits:	0	0
599	Number of processes:	0	16
600	Number of cells:	16	1
601	\$_ANDNOT_	1	2
602	\$_AND_	2	2
603	\$_A0I3_	2	4
604	\$_DFF_PP0_	4	1
605	\$_DFF_PP1_	1	3
606	\$_NOR_	3	3
607	\$_NOT_	3	

```
1 /* Generated by Yosys 0.9 (git sha1 1979e0b) */
2
3 (* top = 1 *)
4 (* src = "moore_with_default.v:1" *)
5 module moor_with_default(clk, rst, j, w);
6
7   wire _00;
8   wire _01;
9   wire _02;
10  wire _03;
11  wire _04;
12  wire _05;
13  wire _06;
14  wire _07;
15  wire _08;
16  wire _09;
17  wire _10;
18  wire _11;
19  wire _12;
20  wire _13;
21
22  (* src = "moore_with_default.v:1" *)
23  input clk;
24  input rst;
25  output w;
26  NOT _14 (
27    .A(j),
28    .Y(_05)
29  );
30
31  NOT _15 (
32    .A(j),
33    .Y(_05)
34  );
35
36  assign w = _14;
37 endmodule
```

```
synth_moore_with_default.v .. synth_moore_without_default.v X
moore_machine_yosys > synth_moore_without_default.v
1 /* Generated by Yosys 0.9 (git sha1 1979e0b) */
2
3 (* top = 1 *)
4 (* src = "moore_with_default.v:1" *)
5 module moor_with_default(clk, rst, j, w);
6
7   wire _00;
8   wire _01;
9   wire _02;
10  wire _03;
11  wire _04;
12  wire _05;
13  wire _06;
14  wire _07;
15  wire _08;
16  wire _09;
17  wire _10;
18  wire _11;
19  wire _12;
20  wire _13;
21
22  (* src = "moore_with_default.v:1" *)
23  input clk;
24  input rst;
25  output w;
26  NOT _14 (
27    .A(j),
28    .Y(_05)
29  );
30
31  NOT _15 (
32    .A(j),
33    .Y(_05)
34  );
35
36  assign w = _14;
37 endmodule
```

```
1 /* Generated by Yosys 0.9 (git sha1 1979e0b) */
2
3 (* top = 1 *)
4 (* src = "moore_without_default.v:1" *)
5 module moor_without_default(clk, rst, j, w);
6
7   wire _00;
8   wire _01;
9   wire _02;
10  wire _03;
11  wire _04;
12  wire _05;
13  wire _06;
14  wire _07;
15  wire _08;
16  wire _09;
17  wire _10;
18  wire _11;
19  wire _12;
20  wire _13;
21
22  (* src = "moore_without_default.v:1" *)
23  input clk;
24  input j;
25  wire [4:0] ps;
26  input rst;
27
28  (* src = "moore_without_default.v:1" *)
29  output w;
30  NOT _14 (
31    .A(j),
32    .Y(_05)
33  );
34
35  NOT _15 (
36    .A(j),
37    .Y(_05)
38  );
39
40  assign w = _14;
41 endmodule
```

```

1  module moor_with_default_v2 (input clk, rst, j, output w);
2      reg [2:0] ns, ps;
3
4      always @(ps,j) begin
5          ns = 3'b100;
6          case(ps)
7              3'b100: ns = j ? 3'b101 : 3'b100;
8              3'b101: ns = j ? 3'b101 : 3'b110;
9              3'b110: ns = j ? 3'b111 : 3'b100;
10             3'b111: ns = j ? 3'b000 : 3'b110;
11             3'b000: ns = j ? 3'b101 : 3'b110;
12             default: ns = 3'b100;
13         endcase
14     end
15
16     assign w = (ps == 3'b000) ? 1'b1 : 1'b0;
17
18     always @(posedge clk, posedge rst) begin
19         if (rst)
20             ps <= 3'b100;
21         else
22             ps <= ns;
23     end
24 endmodule

```

```

=== moor_with_default_v2 ===

```

Number of wires:	16
Number of wire bits:	20
Number of public wires:	5
Number of public wire bits:	9
Number of memories:	0
Number of memory bits:	0
Number of processes:	0
Number of cells:	16
\$_ANDNOT_	1
\$_AND_	2
\$_AOI3_	2
\$_DFF_PP0_	4
\$_DFF_PP1_	1
\$_NOR_	3
\$_NOT_	3


```

1  module moor_without_default_v2 (input clk, rst, j, output w);
2      reg [2:0] ns, ps;
3
4      always @(ps,j) begin
5          ns = 3'b100;
6          case(ps)
7              3'b100: ns = j ? 3'b101 : 3'b100;
8              3'b101: ns = j ? 3'b101 : 3'b110;
9              3'b110: ns = j ? 3'b111 : 3'b100;
10             3'b111: ns = j ? 3'b000 : 3'b110;
11             3'b000: ns = j ? 3'b101 : 3'b110;
12         endcase
13     end
14
15     assign w = (ps == 3'b000) ? 1'b1 : 1'b0;
16
17     always @(posedge clk, posedge rst) begin
18         if (rst)
19             ps <= 3'b100;
20         else
21             ps <= ns;
22     end
23 endmodule

```

```

=== moor_without_default_v2 ===

```

Number of wires:	16
Number of wire bits:	20
Number of public wires:	5
Number of public wire bits:	9
Number of memories:	0
Number of memory bits:	0
Number of processes:	0
Number of cells:	16
\$_ANDNOT_	1
\$_AND_	2
\$_AOI3_	2
\$_DFF_PP0_	4
\$_DFF_PP1_	1
\$_NOR_	3
\$_NOT_	3

```
=== moor with default v2 ===
```

\$ NOT 3

The image displays two versions of a Verilog Moore machine implementation, side-by-side, illustrating the effect of removing a default state.

Left Pane (moore_with_default_v2.v): This code includes a default state for the Moore machine. Key features include:

- A `case` statement for the `next_state` logic that includes a default entry: `default: s0;` (line 19).
- A `case` statement for the `output` logic that includes a default entry: `default: 4'b0000;` (line 29).
- The initial state is set to `s0` in the `initial` block (line 34).

Right Pane (moore_without_default_v2.v): This code removes the default state, relying on the initial state assignment. Key changes include:

- The `case` statement for `next_state` no longer has a default entry (line 19).
- The `case` statement for `output` no longer has a default entry (line 29).
- The initial state is still set to `s0` in the `initial` block (line 34).

Annotations:

- A red arrow points from the `default: s0;` line in the left pane to the `case` statement in the right pane.
- Another red arrow points from the `default: 4'b0000;` line in the left pane to the `case` statement in the right pane.
- A third red arrow points from the `initial: s0;` line in the left pane to the `initial: s0;` line in the right pane.