\[ \text{always @ (posedge clk) begin} \]
\[ \text{if (reset == 1'61) begin} \]
\[ \text{out <= #1 1'60;} \]
\[ \text{else end} \]
\[ \text{else begin} \]
\[ \text{out <= #1 D;} \]
\[ \text{end} \]
\[ \text{end} \]

**FF with reset and En**

<table>
<thead>
<tr>
<th>reset</th>
<th>En</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>disabled</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>OFF FF</td>
</tr>
<tr>
<td>*</td>
<td>1</td>
<td>reset? disabled?</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>reset, Q=0</td>
</tr>
</tbody>
</table>
always@prelude clk begin

if (reset == '1'51) --

end
end

CLK
a
b
c
d
101
101
X
Y
X
Y

reset only these & multi-cycle reset
Asynchronous reset

Rule 9: Use my synchronous resets
Design Compiler (DC) - Synopsys

HW
verilog
* .v

D.C.
.synopsys -dc .setup

gate
netlist
*.vg

reports

dc_compile

std
cell
library

\[
\frac{1}{\text{setup}} = 10 \text{ns} - 5.07 \text{ns}
\]

Clock

G.L.

clock

\[ t_{\text{setup}} \]
<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>cin</th>
<th>c0</th>
<th>c1</th>
<th>s</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

\[ 0, 1, 4, 5 \]