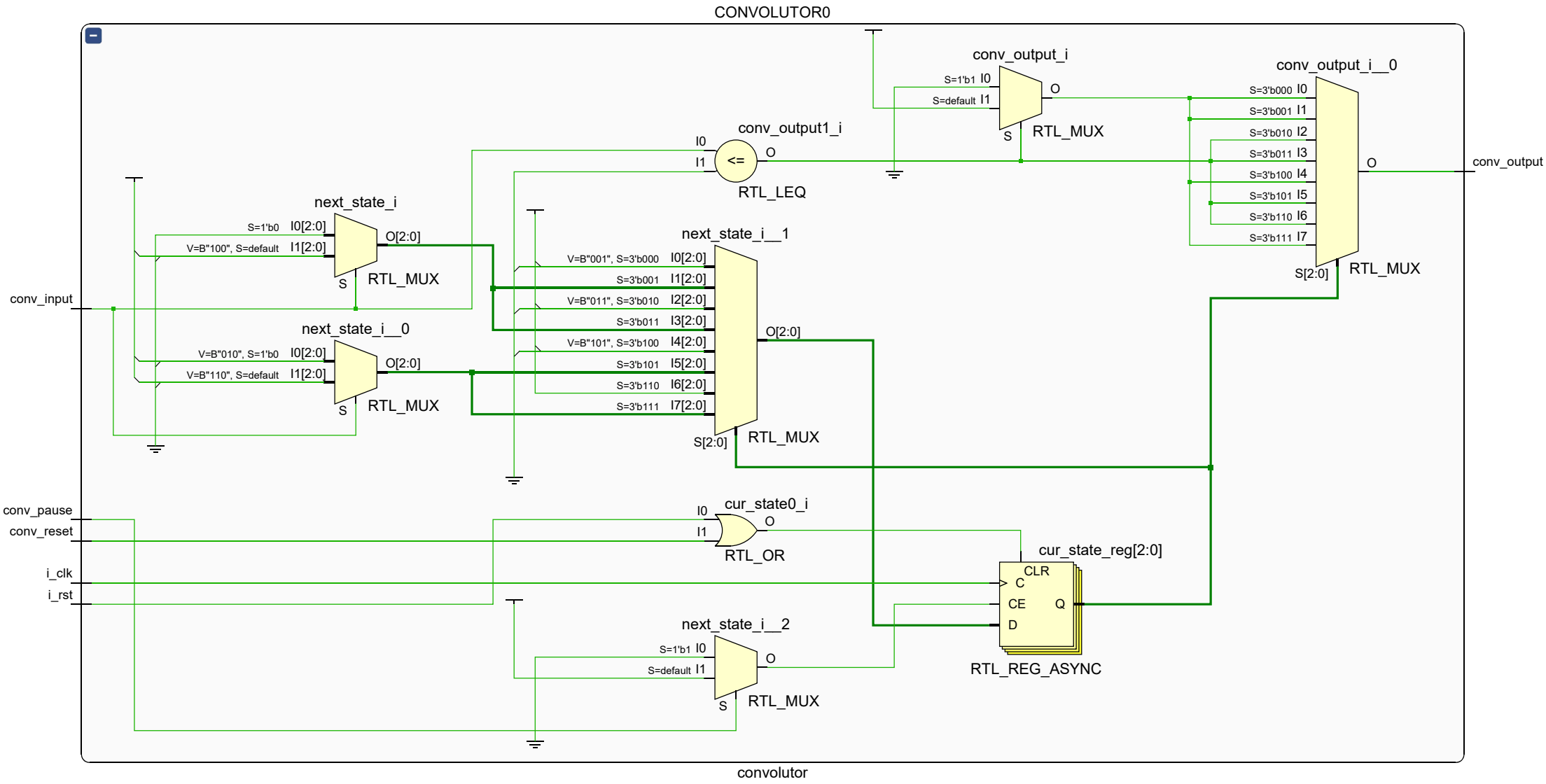
CONVOLUTOR0
conv_output_i
S=1'b1 I0
S=default I1
O
S
RTL_MUX
conv_output_i__0
S=3'b000 I0
S=3'b001 I1
S=3'b010 I2
S=3'b011 I3
S=3'b100 I4
S=3'b101 I5
S=3'b110 I6
S=3'b111 I7
O
S[2:0]
RTL_MUX
conv_output
conv_output1_i
I0
I1
<=
O
RTL_LEQ
next_state_i
S=1'b0 I0[2:0]
V=B"100", S=default I1[2:0]
O[2:0]
S
RTL_MUX
conv_input
next_state_i__1
V=B"001", S=3'b000 I0[2:0]
S=3'b001 I1[2:0]
V=B"011", S=3'b010 I2[2:0]
S=3'b011 I3[2:0]
V=B"101", S=3'b100 I4[2:0]
S=3'b101 I5[2:0]
S=3'b110 I6[2:0]
S=3'b111 I7[2:0]
O[2:0]
S[2:0]
RTL_MUX
next_state_i__0
V=B"010", S=1'b0 I0[2:0]
V=B"110", S=default I1[2:0]
O[2:0]
S
RTL_MUX
conv_pause
conv_reset
i_clk
i_rst
cur_state0_i
I0
I1
O
RTL_OR
cur_state_reg[2:0]
CLR
C
CE
D
Q
RTL_REG_ASYNC
next_state_i__2
S=1'b1 I0
S=default I1
O
S
RTL_MUX
convolutor