

3.4 Sequential logic

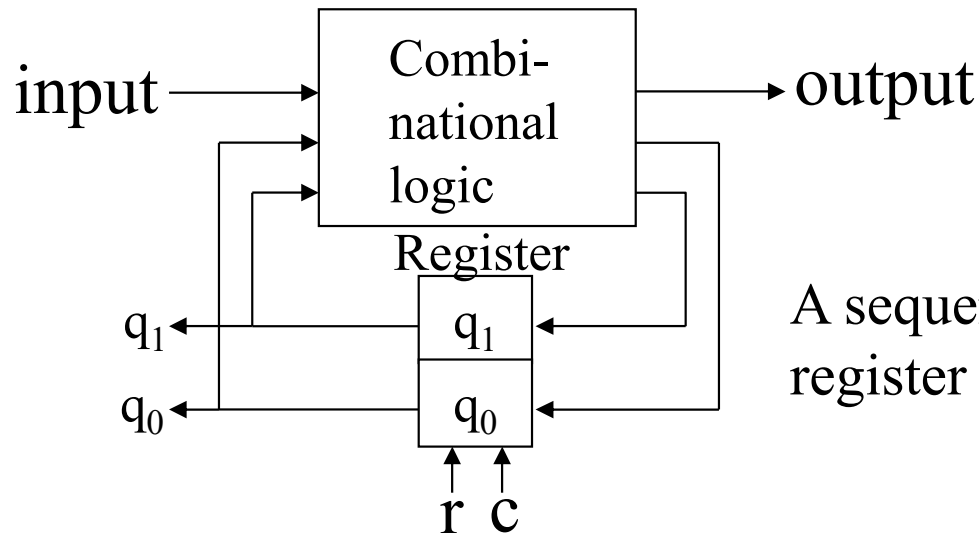
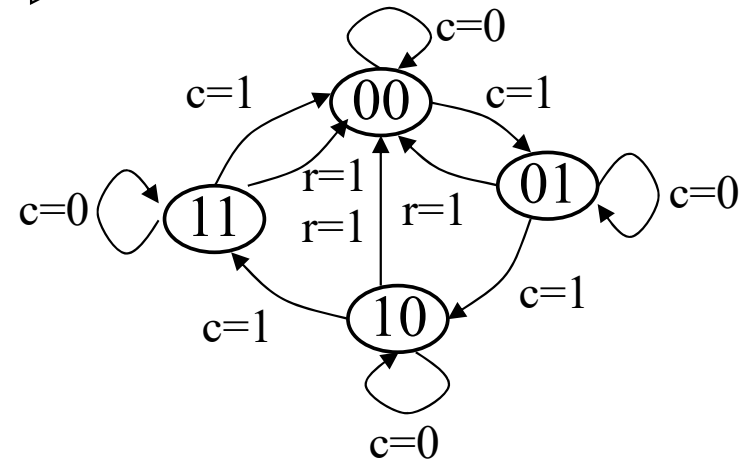
Fundamentals of control circuits

Implementation of sequential logic

State Transition table

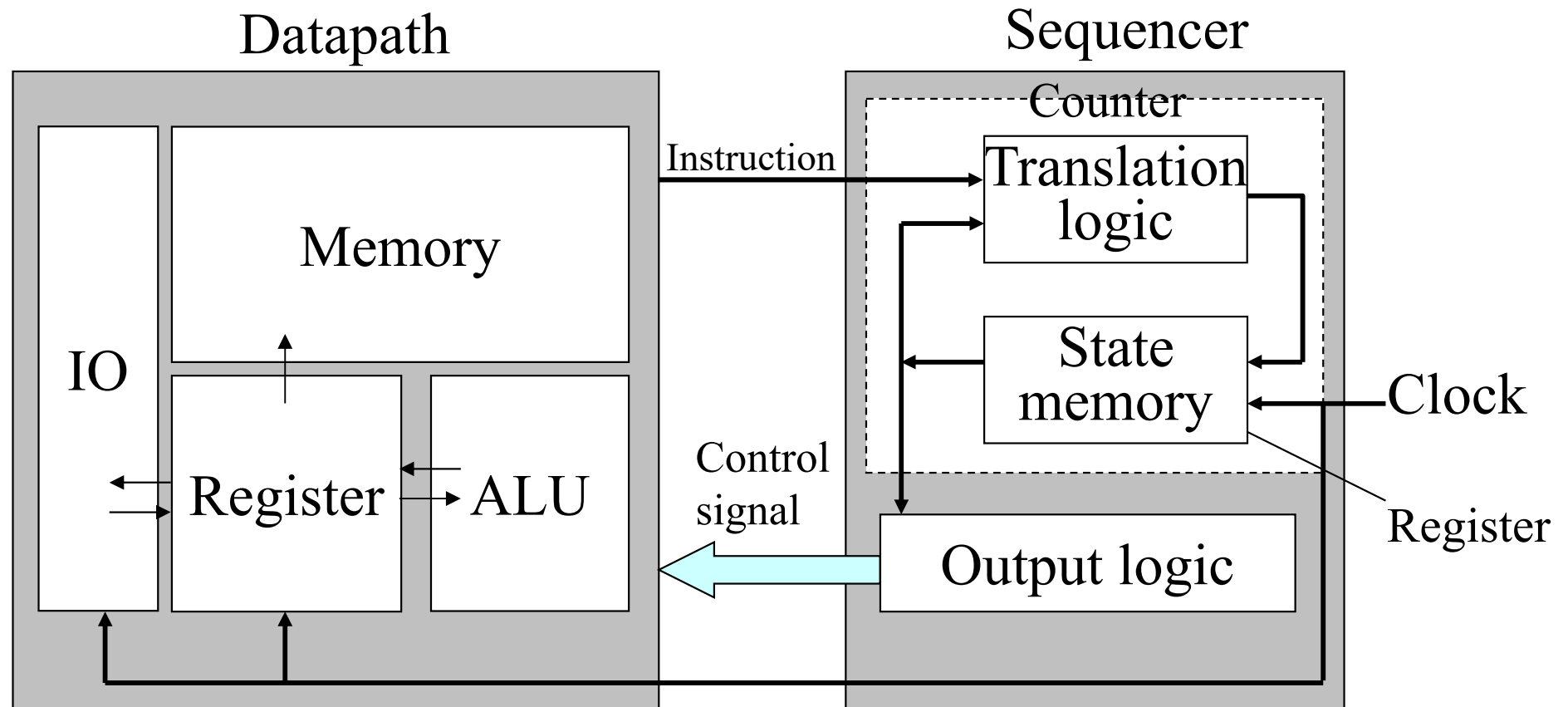
r	c	$q_1q_0(n+1)$
0	0	$q_1q_0(n)$
0	1	$q_1q_0(n) + "01"$
1	0	0 0
1	1	0 0

State Transition Diagram



A sequential logic consists of a register and a combinational logic.

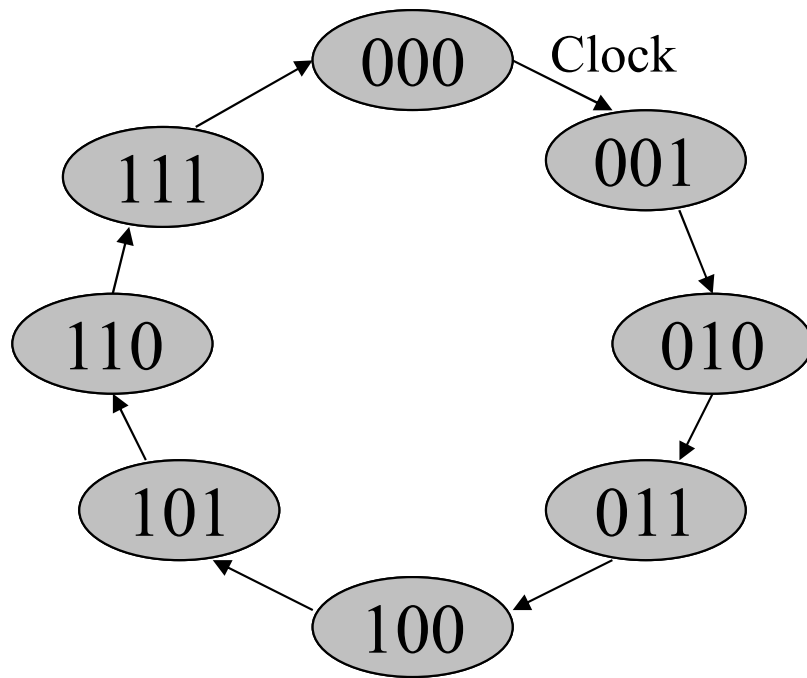
Datapath and sequencer



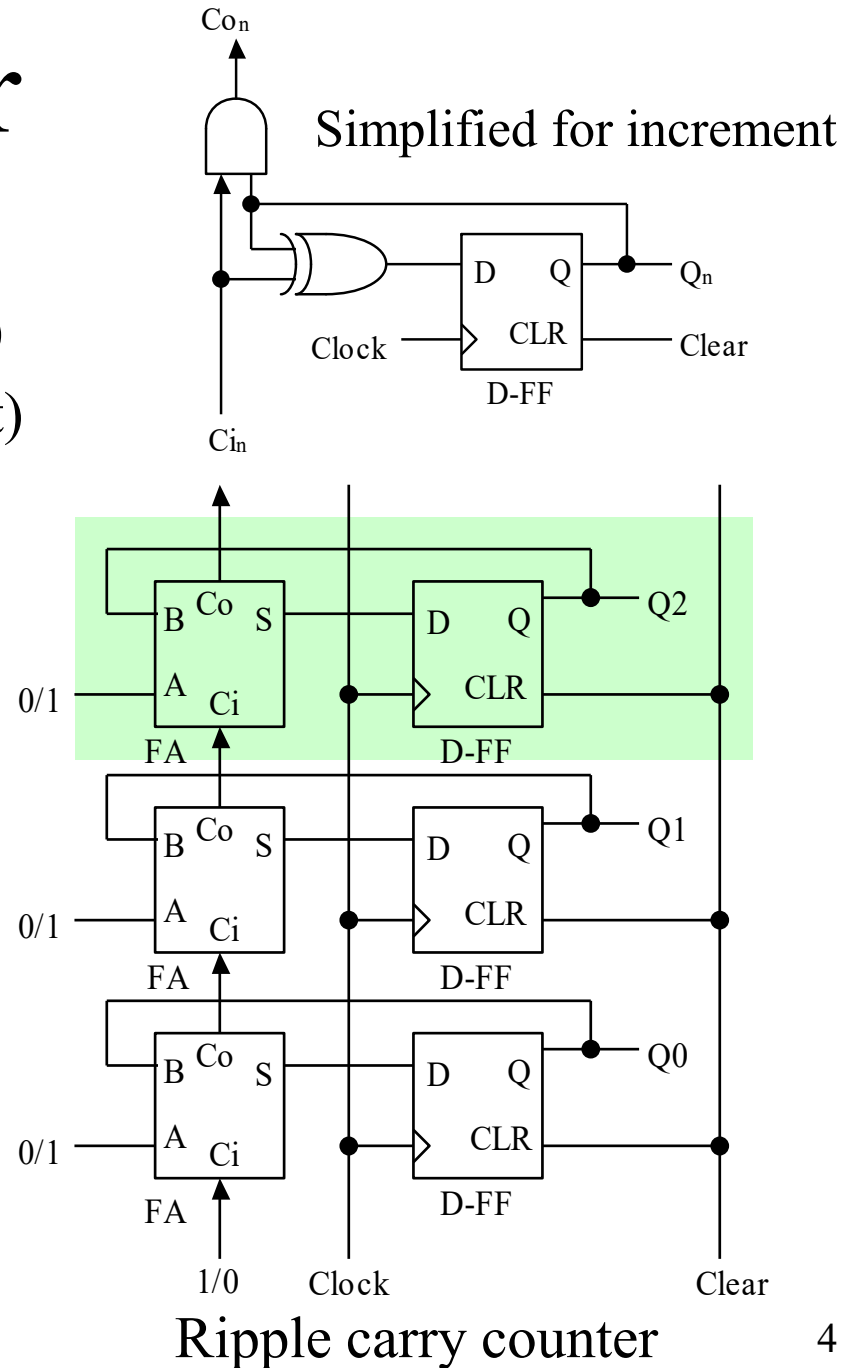
Synchronous state machine

Binary counter

A	C _i	
0	1	→ (Increment)
1	0	→ (Decrement)

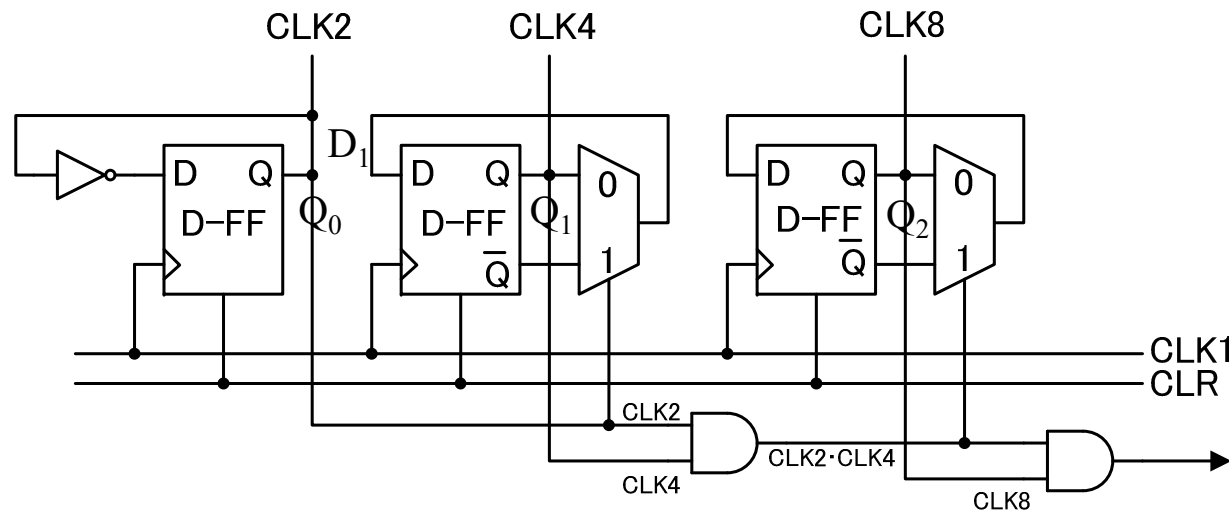
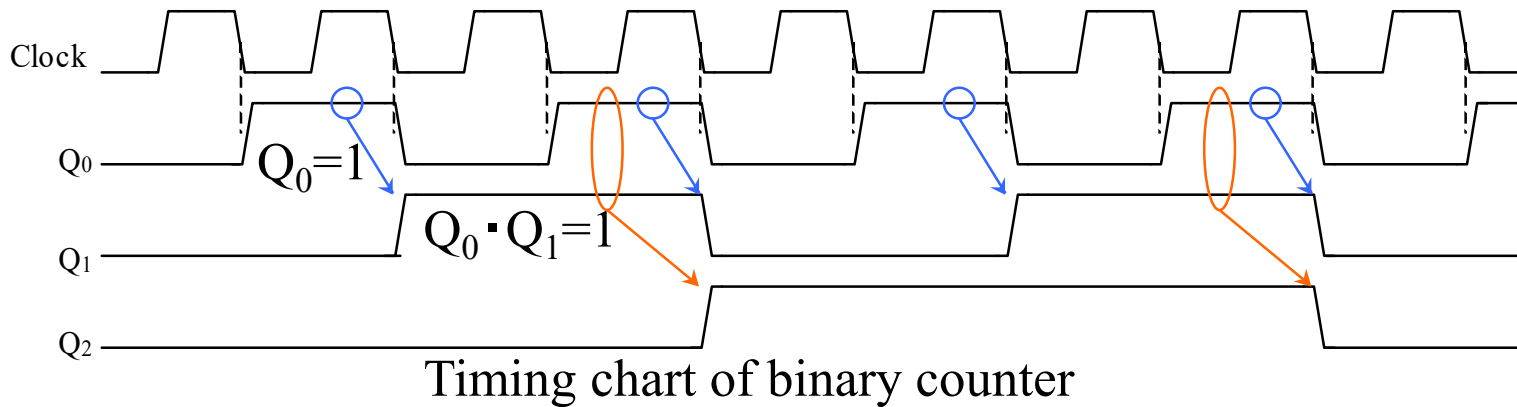


State transition of binary counter

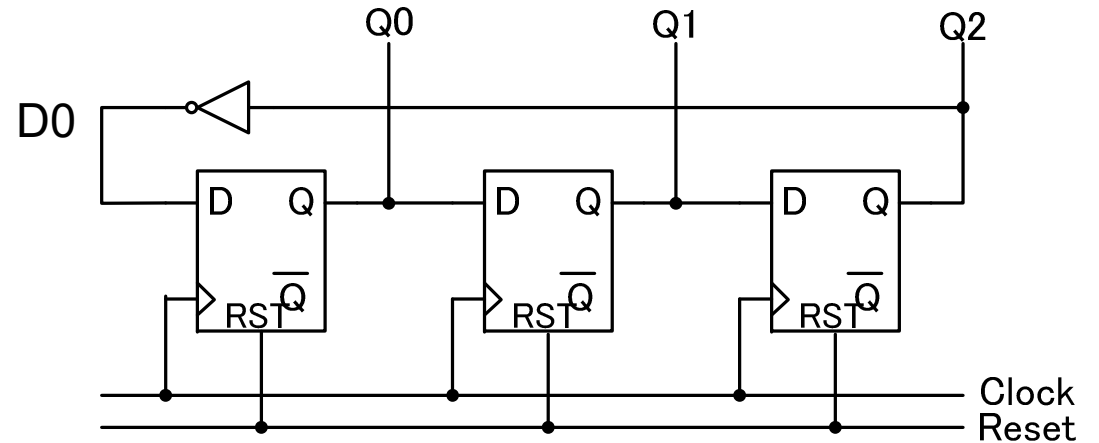
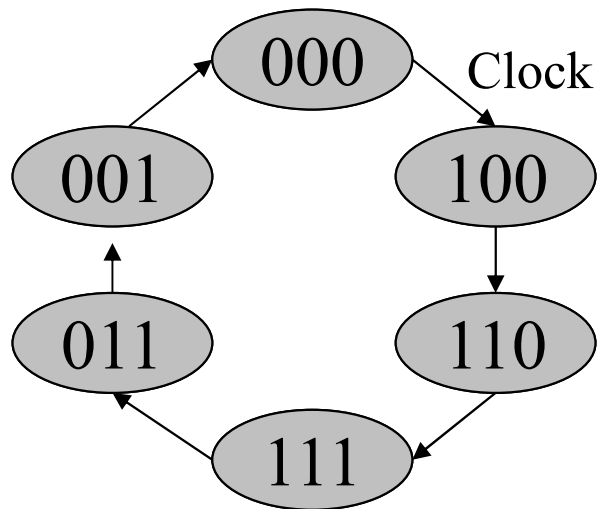


Binary counter with carry lookahead mechanism

A value of D_1 is determined according to the result of $Q_0 \cdot Q_1$ (In general, $\prod_{i=0}^{n-1} Q_i$).
 The maximum clock frequency is higher than that of the ripple carry counter.



Johnson counter



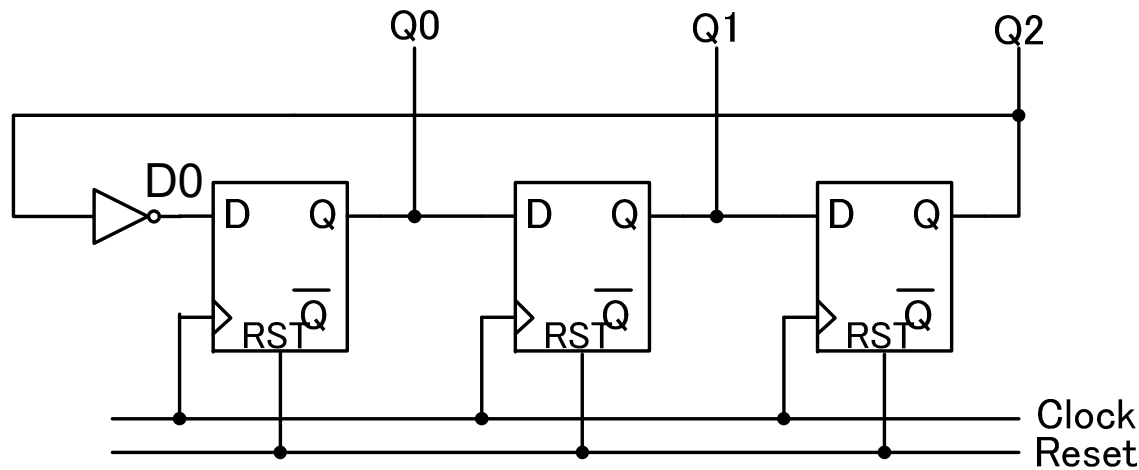
State transition of Johnson counter

There is no hazard during the state transition, because the 1-bit of the state variable is inverted for every transition.

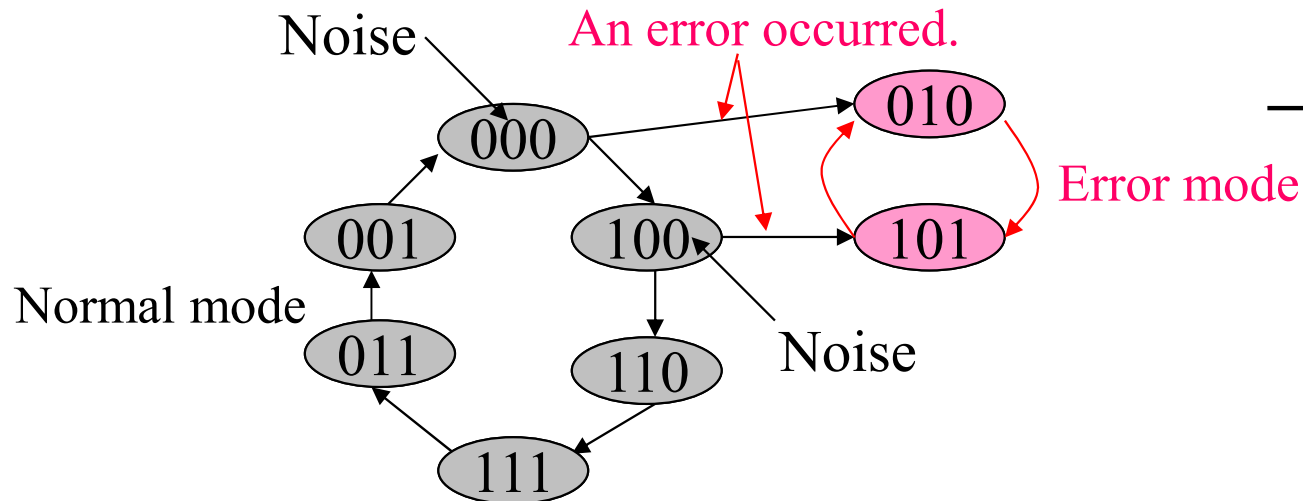
	Q0	Q1	Q2	D0
	0	0	0	1
	1	0	0	1
	1	1	0	1
	1	1	1	0
	0	1	1	0
	0	0	1	0
	0	0	0	1

NOT

Transition to undefined state

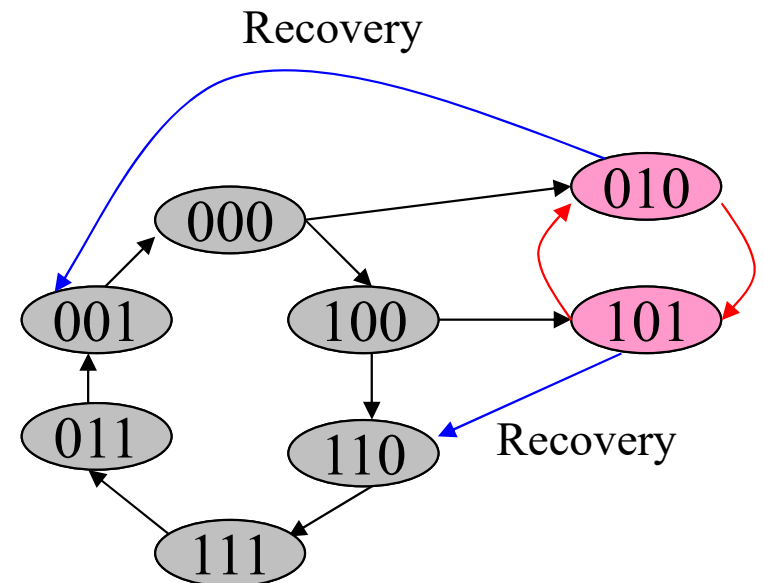


Q0	Q1	Q2	D0
0	0	0	1
1	0	0	1
1	1	0	1
1	1	1	0
0	1	1	0
0	0	1	0
<hr/>			
0	1	0	Undefined
1	0	1	Undefined



Recovery by error handling

	Q0	Q1	Q2	D0	
	0	0	0	1	} Normal mode
	1	0	0	1	
Recovery	1	1	0	1	
	1	1	1	0	
	0	1	1	0	} Error mode
Recovery	0	0	1	0	
	0	1	0	0	
	1	0	1	1	



Note: A condition branching that has an undefined transition may cause an unexpected error in the hardware and software design.

Design example of error handling

Truth table of translation logic

Q ₀	Q ₁	Q ₂	D ₀
0	0	0	1
1	0	0	1
1	1	0	1
1	1	1	0
0	1	1	0
0	0	1	0
<hr/>			
0	1	0	0
1	0	1	1

Error detection

$$E = Q_0 \cdot \overline{Q_1} \cdot Q_2 + \overline{Q_0} \cdot Q_1 \cdot \overline{Q_2} = \overline{\overline{Q_0 \cdot \overline{Q_1} \cdot Q_2 + \overline{Q_0} \cdot Q_1 \cdot \overline{Q_2}}}$$

Next state of D-FF₀

$$D_0 = E \cdot Q_2 + \overline{E} \cdot \overline{Q_2} = \overline{E \oplus Q_2}$$

