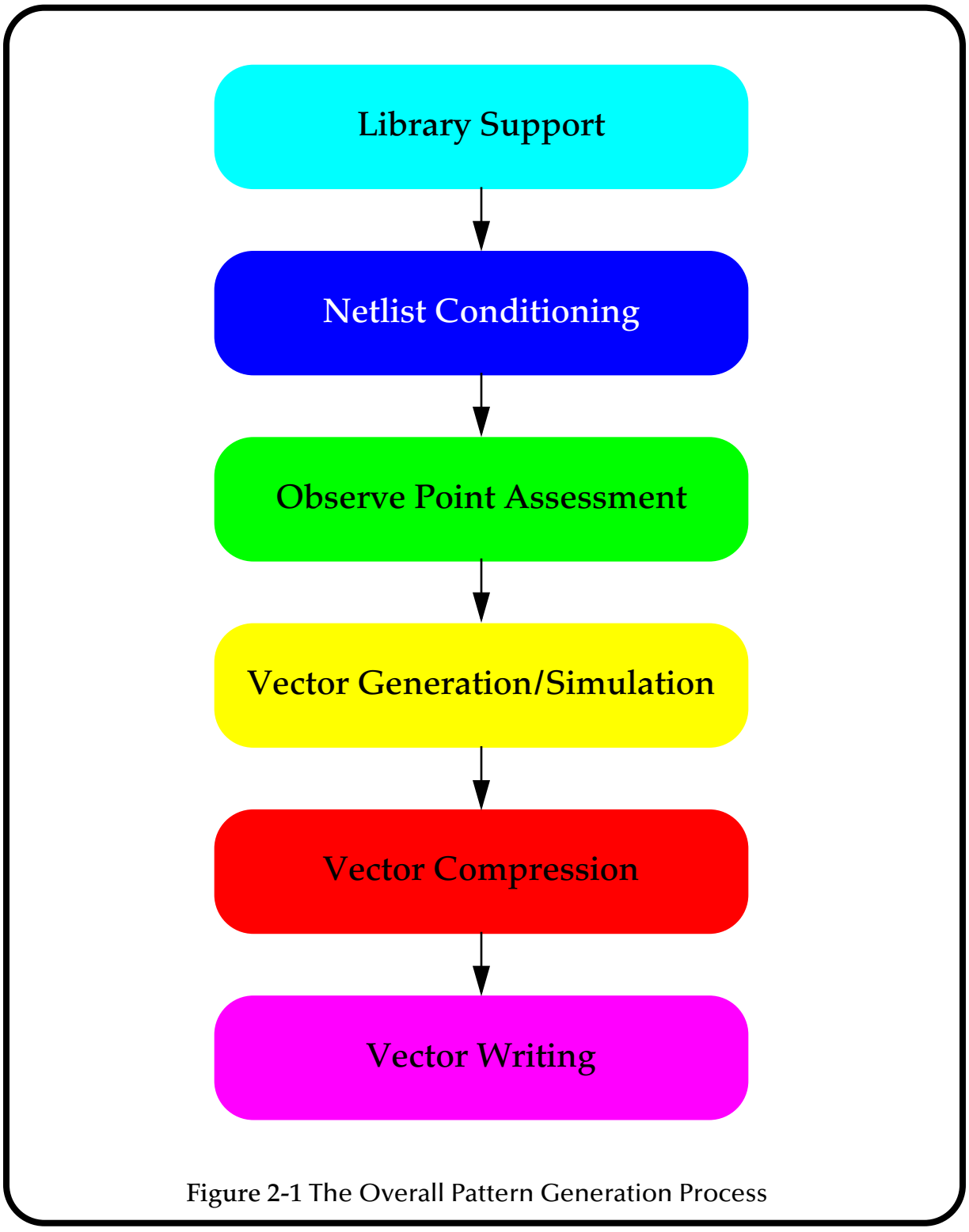


Figure 2-1 The Overall Pattern Generation Process

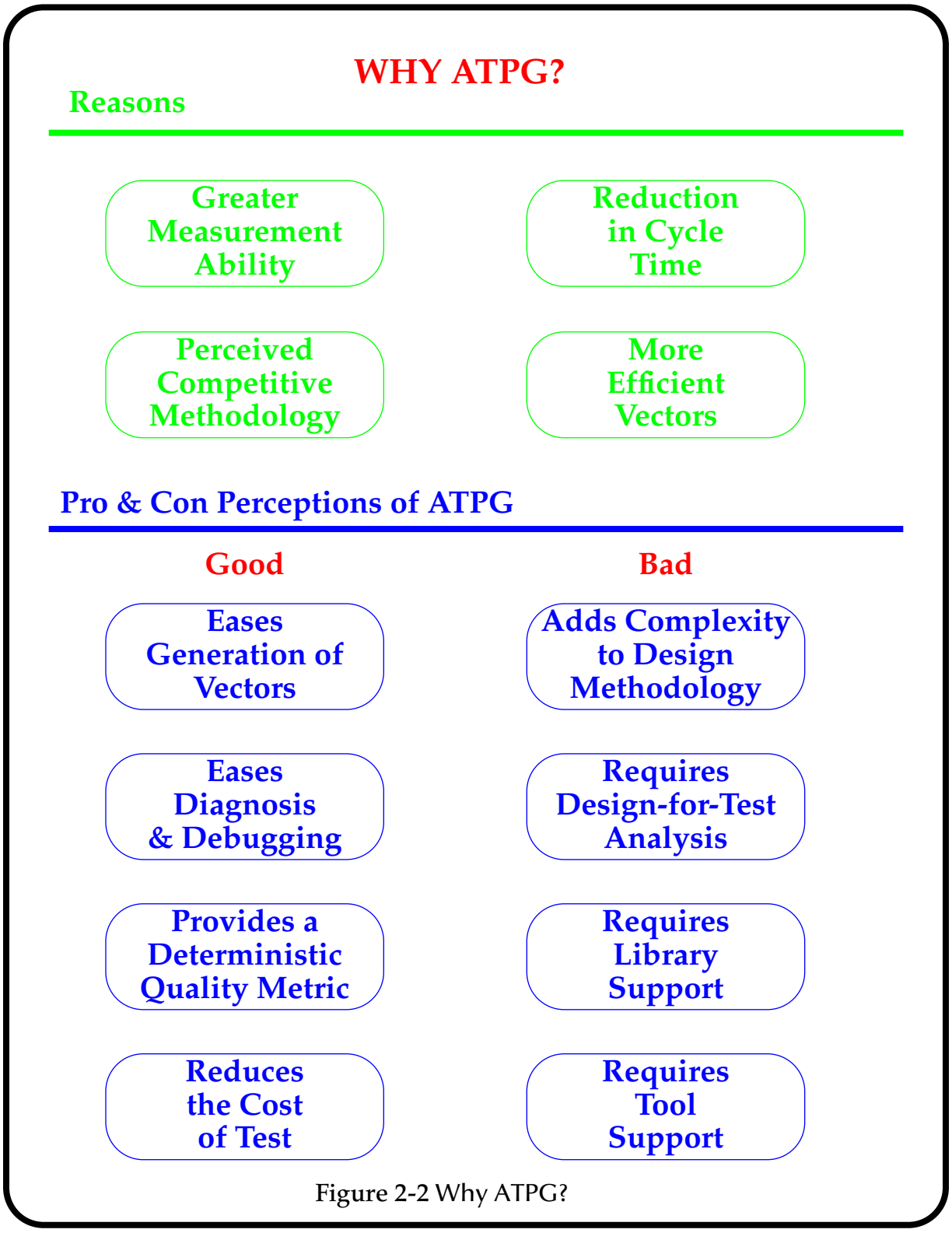


Figure 2-2 Why ATPG?

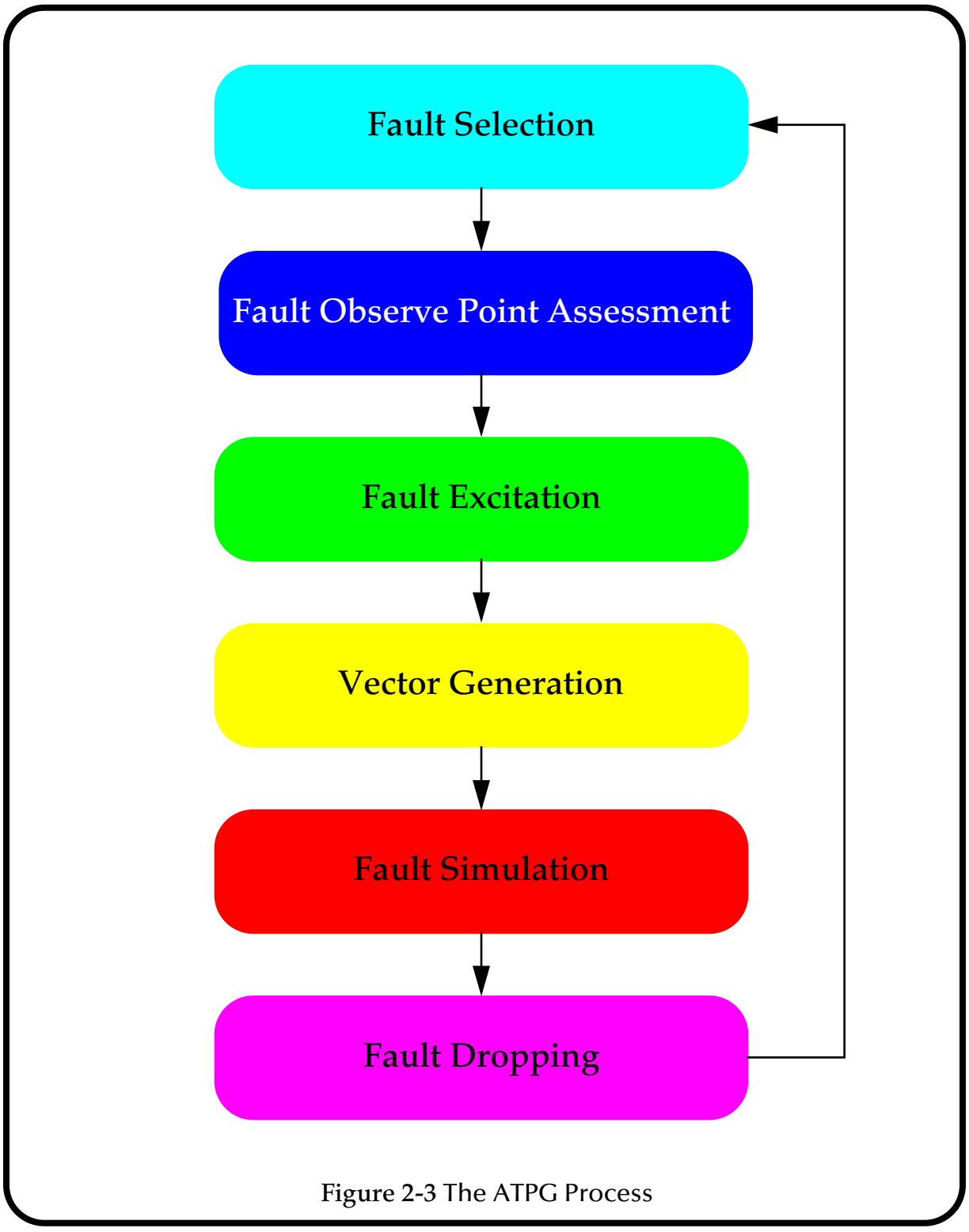


Figure 2-3 The ATPG Process

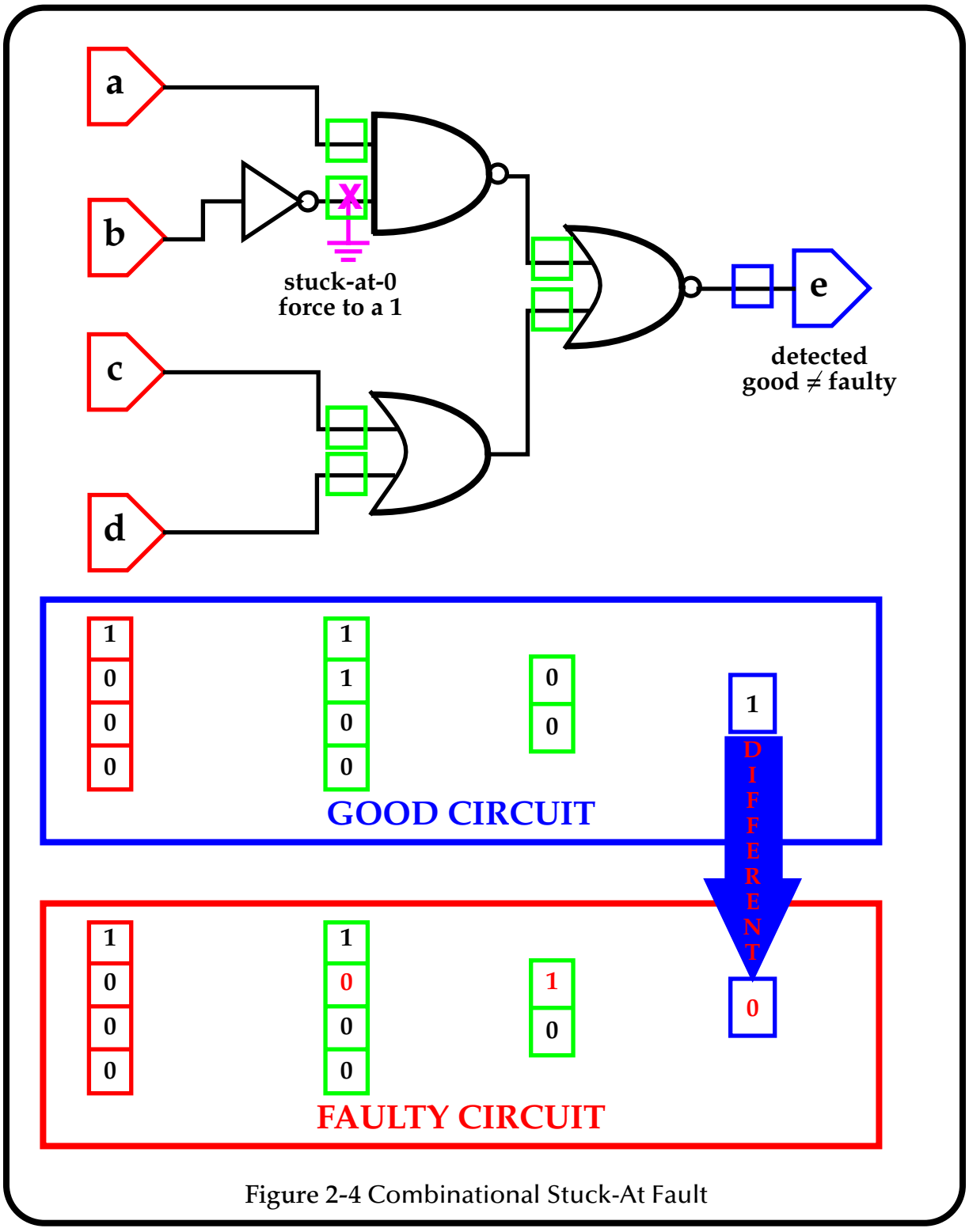


Figure 2-4 Combinational Stuck-At Fault

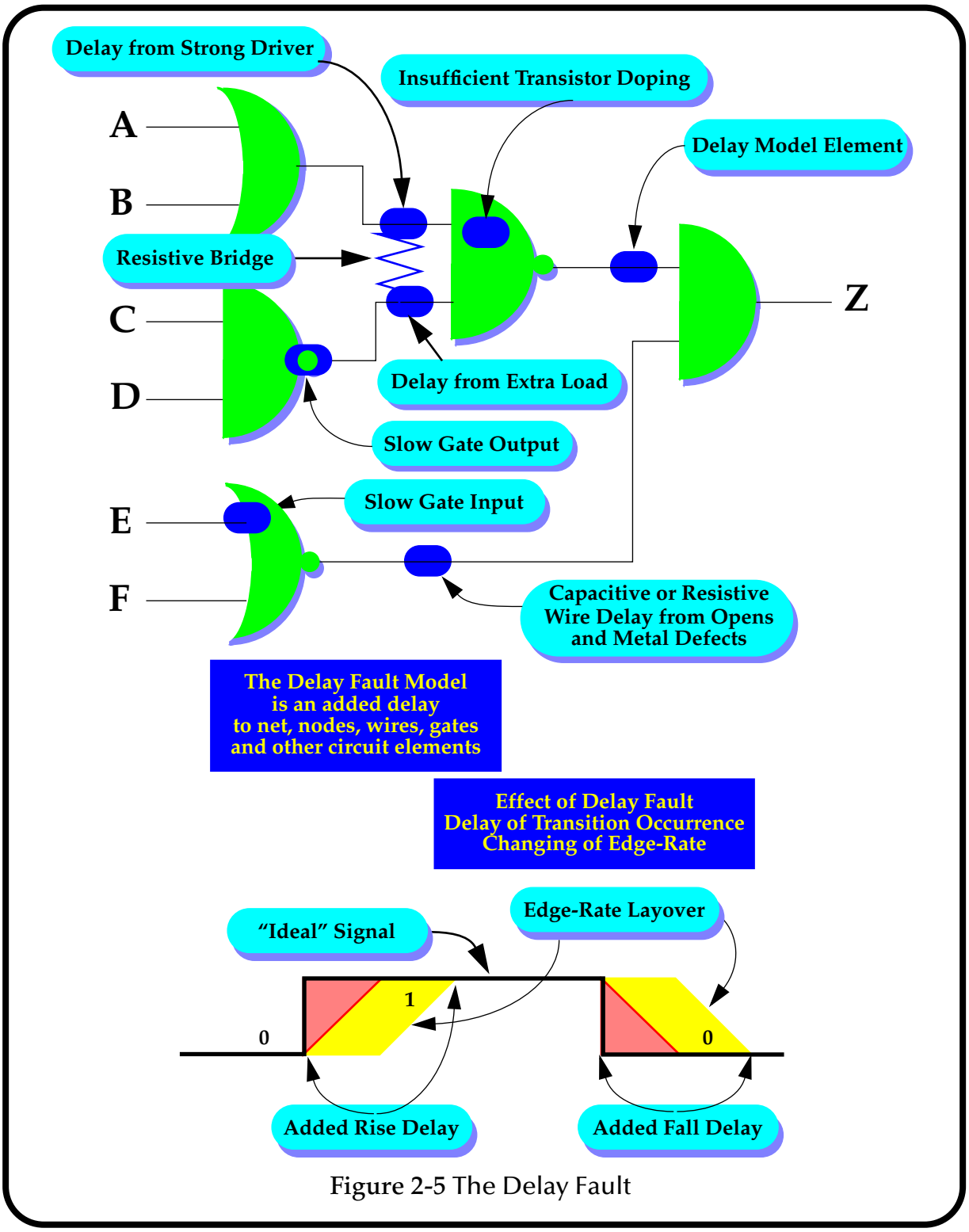


Figure 2-5 The Delay Fault

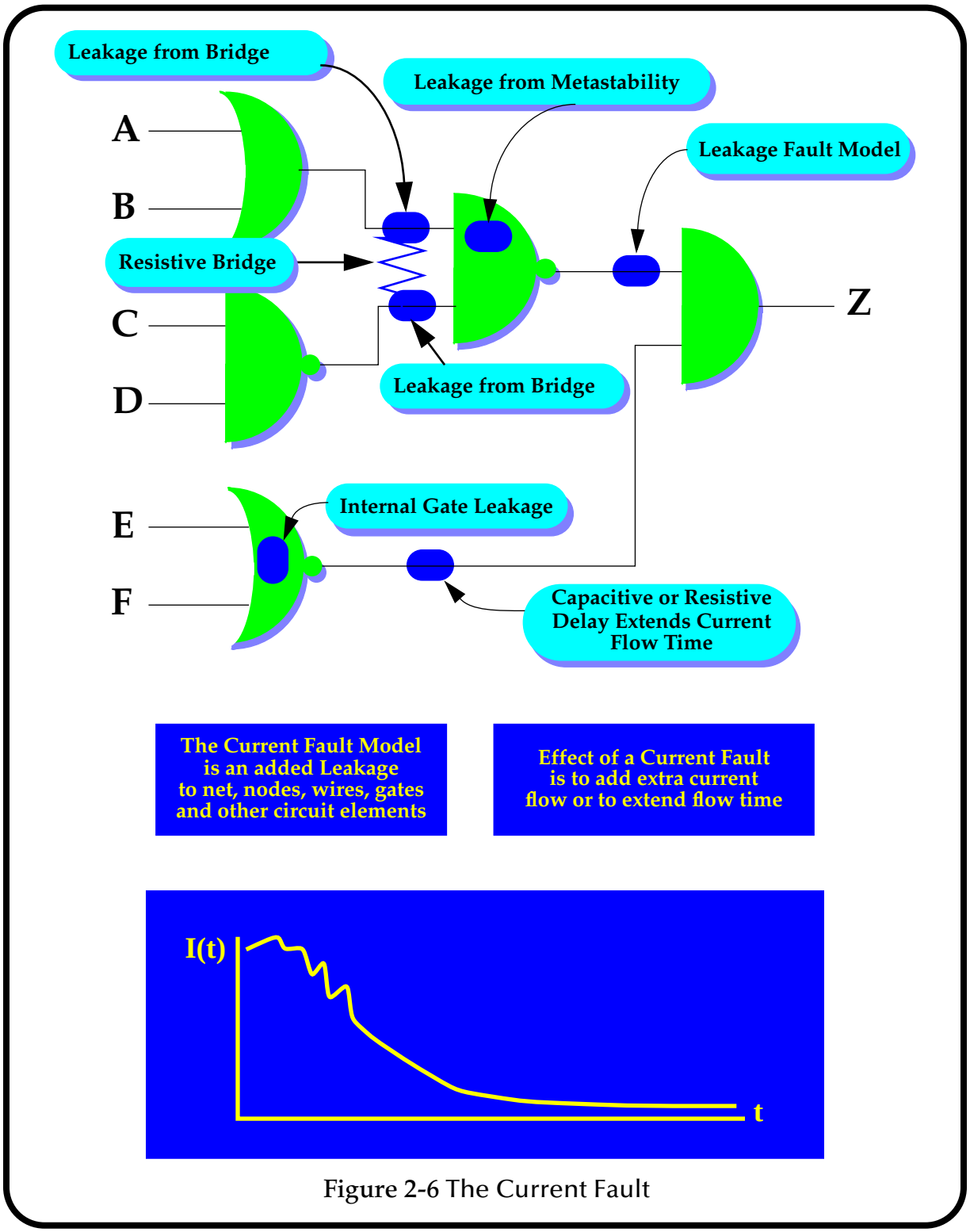


Figure 2-6 The Current Fault

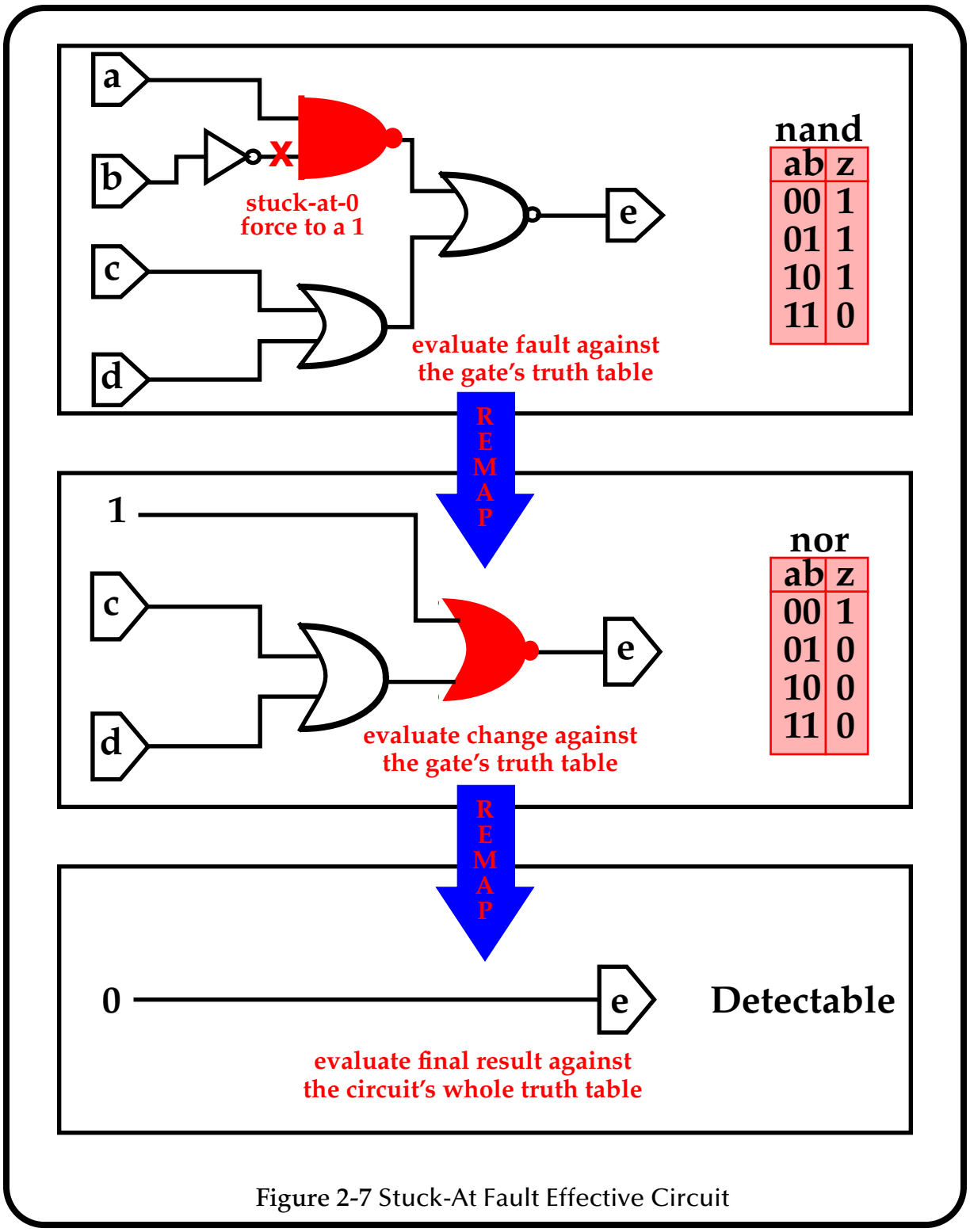


Figure 2-7 Stuck-At Fault Effective Circuit

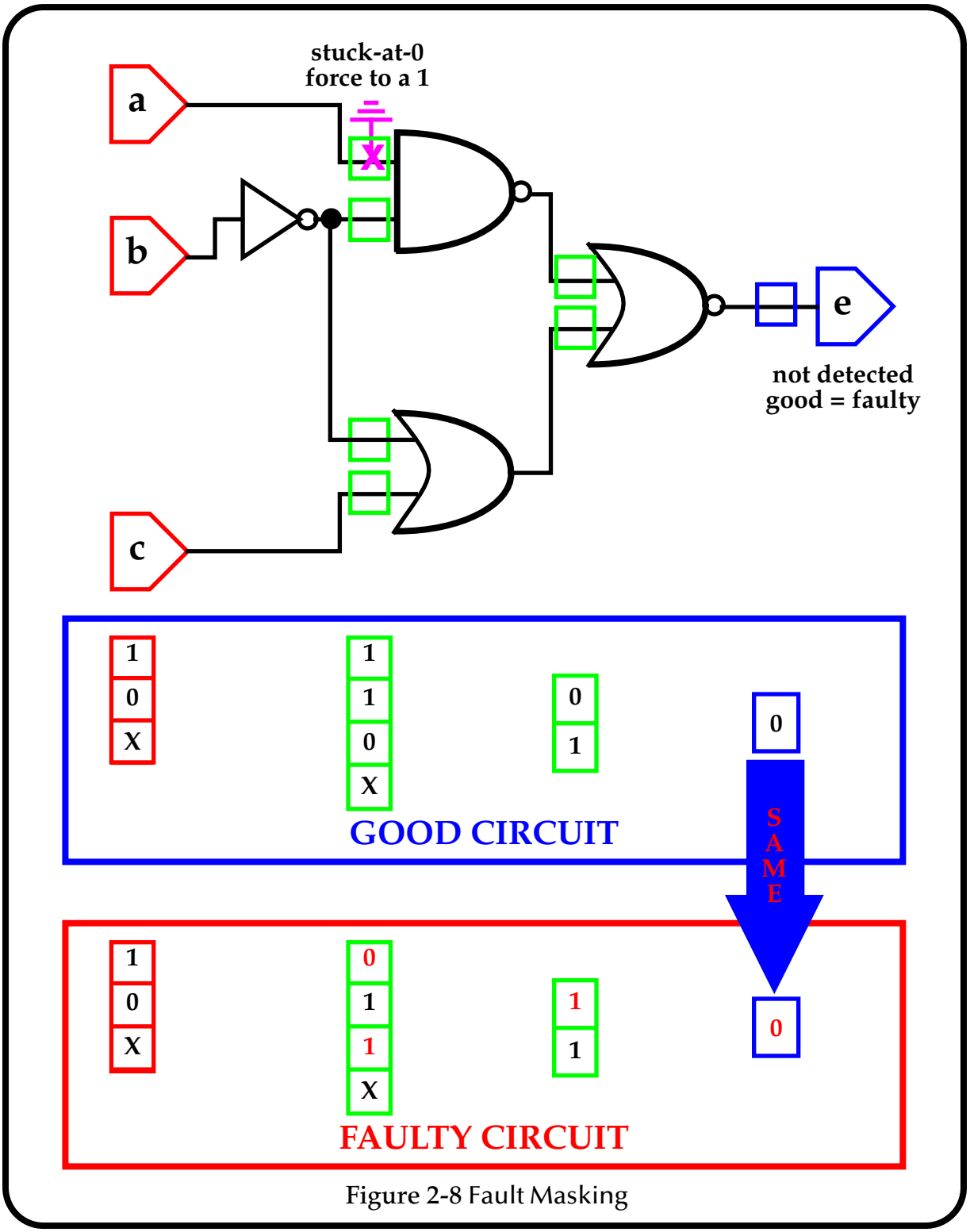
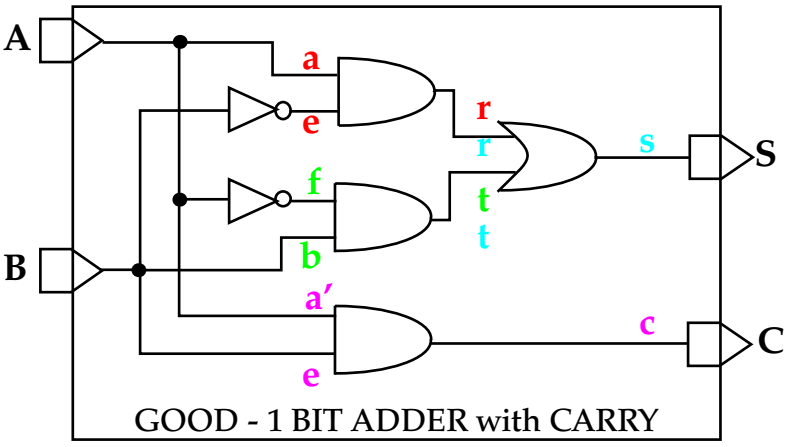
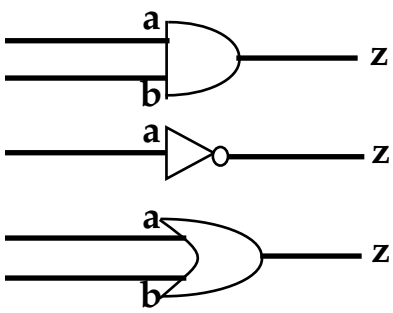


Figure 2-8 Fault Masking



faultlist	
a @ 0	a @ 1
b @ 0	b @ 1
e @ 0	e @ 1
f @ 0	f @ 1
r @ 0	r @ 1
t @ 0	t @ 1
s @ 0	s @ 1
c @ 0	c @ 1
16 faults	

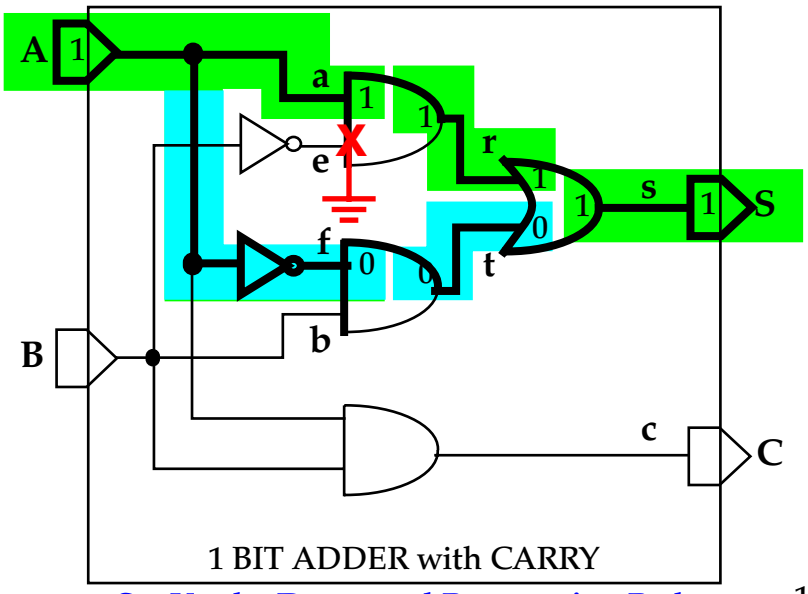


Fault Equivalence Table

AND	**a@0 = b@0 = z@0**
INV	**a@1 = z@0 : a@0 = z@1**
OR	**a@1 = b@1 = z@1**

1. Any fault that requires a logic 1 on the output of an AND-gate will also place 1's on inputs
2. Similar analysis exists for all other gate-level elements
3. If one fault is detected, all equivalent faults are detected
4. Fault selection only needs to target one of the equivalent faults

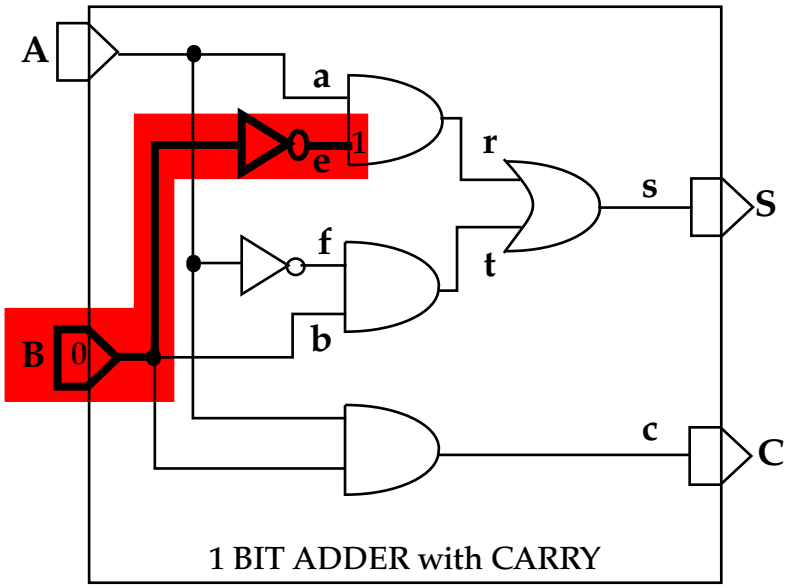
Figure 2-9 Fault Equivalence Example



Set Up the Detect and Propagation Path

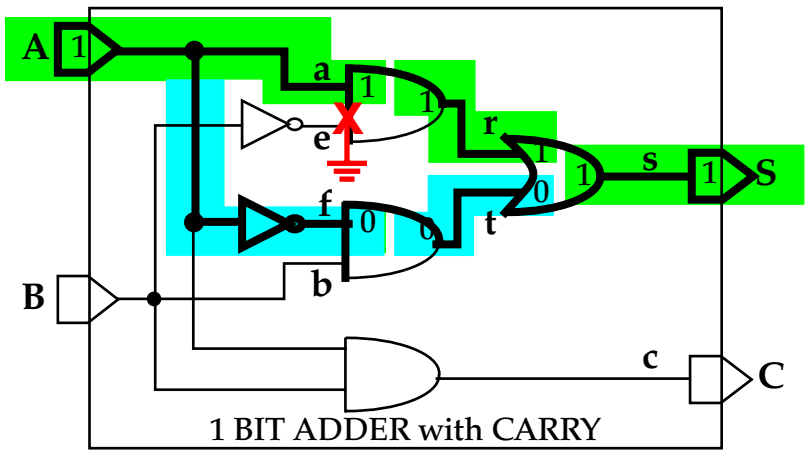
faultlist	
a@ 0	a@ 1
b@ 0	b@ 1
e@ 0	e@ 1
f @ 0	f @ 1
r @ 0	r @ 1
t @ 0	t @ 1
s @ 0	s @ 1
c @ 0	c @ 1
16 faults	

1. Set up the path to pass the opposite of e S @ 0, which is e = 1
2. Exercise by setting e equal to1
3. Detect by observing S for wrong value during fault simulation



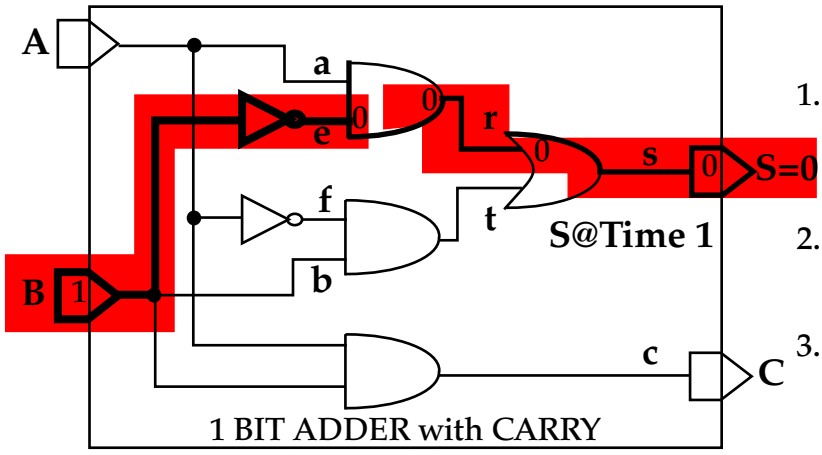
Exercise the Fault

Figure 2-10 Stuck-At Fault ATPG

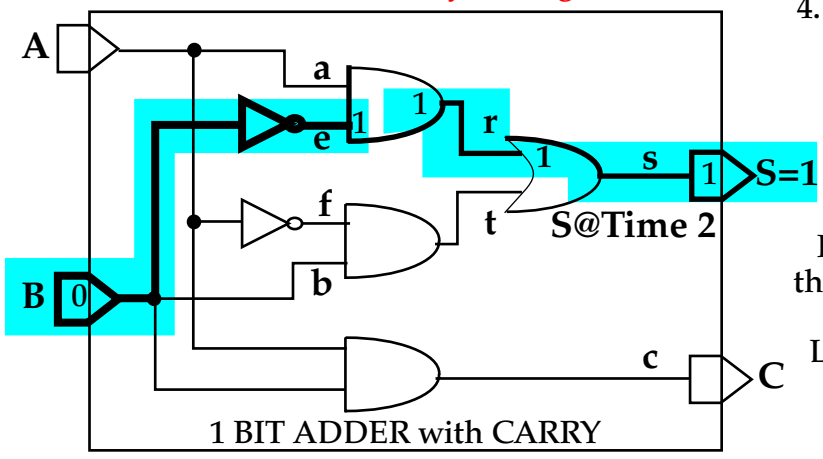


Set Up the Detect Path to Pass a 1

faultlist	
a@ 0	a@ 1
b@ 0	b@ 1
e@ 0	e@ 1
f @ 0	f @ 1
r @ 0	r @ 1
t @ 0	t @ 1
s @ 0	s @ 1
c@ 0	c@ 1
16 faults	



Pre-Fail the Fault by Passing a 0

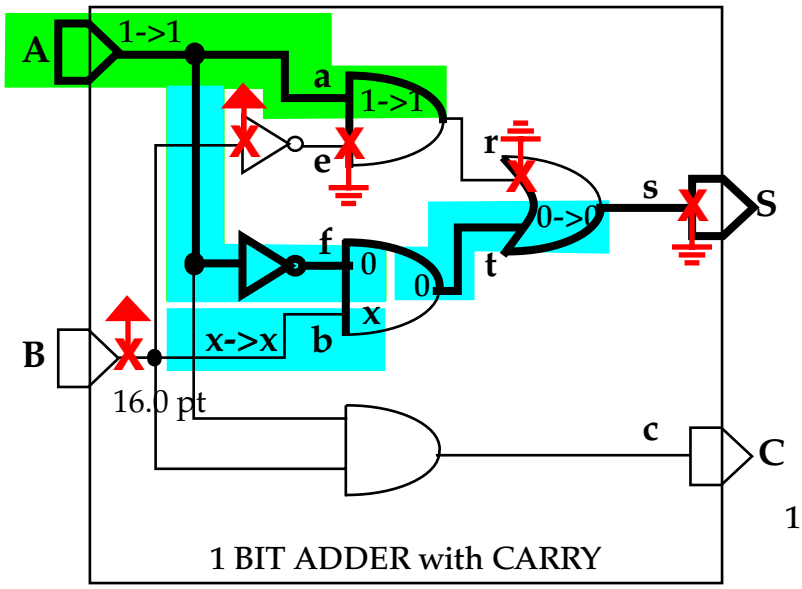


Exercise the Fault to Pass a 1

1. Set up the path to pass the opposite of e S @ 0, which is e = 1
2. Pre-fail by setting e equal to 0
3. Exercise by setting e equal to 1 some time period later
4. Detect by observing S for wrong value during timing simulation

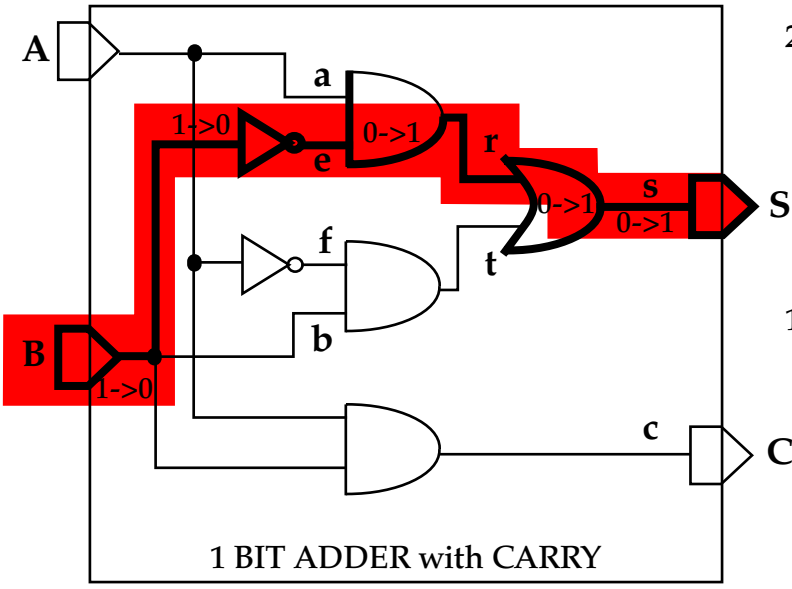
The Transition Delay Faultlist is identical to the Stuck-At Faultlist but the goal is to detect a Logic Transition within a given time period

Figure 2-11 Transition Delay Fault



faultlist	
a@ 0	a@ 1
b@ 0	b@ 1
e@ 0	e@ 1
f @ 0	f @ 1
r @ 0	r @ 1
t @ 0	t @ 1
s @ 0	s @ 1
c @ 0	c @ 1
16 faults	

1. Set up the path to pass a transition on B-to-S through e, r, and s by setting the off-path values to be stable for 2 time periods



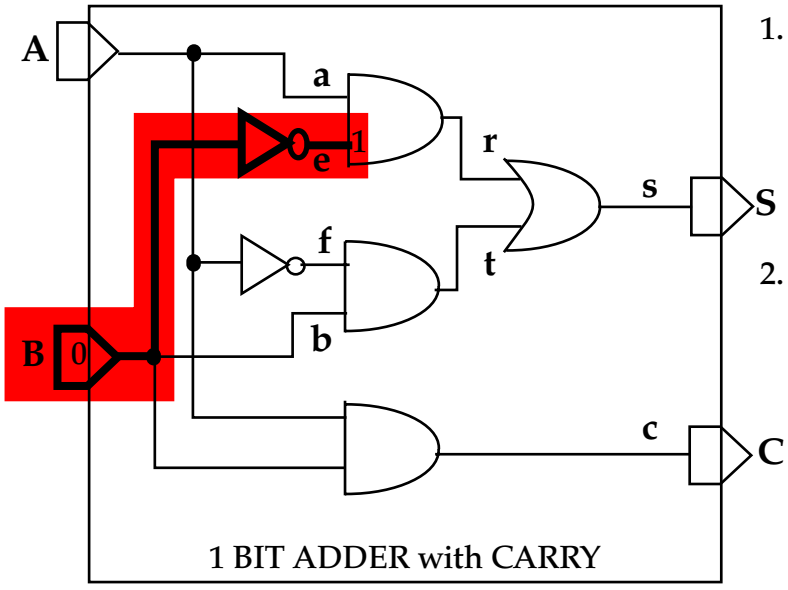
2. Exercise by first setting B equal to 1 and then to 0. This is known as a vector-pair

S@Time1 -> S@Time2

1. Detect by observing S for wrong value during fault simulation with respect to a time standard

Figure 2-12 Path Delay Fault

faultlist	
a@ 0	a@ 1
b@ 0	b@ 1
e@ 0	e@ 1
f @ 0	f @ 1
r @ 0	r @ 1
t @ 0	t @ 1
s @ 0	s @ 1
c@ 0	c@ 1
16 faults	



1. Exercise by first setting e equal to1
2. Detect by measuring current and accept vector by quietness

Exercise the Fault

Figure 2-13 Current Fault

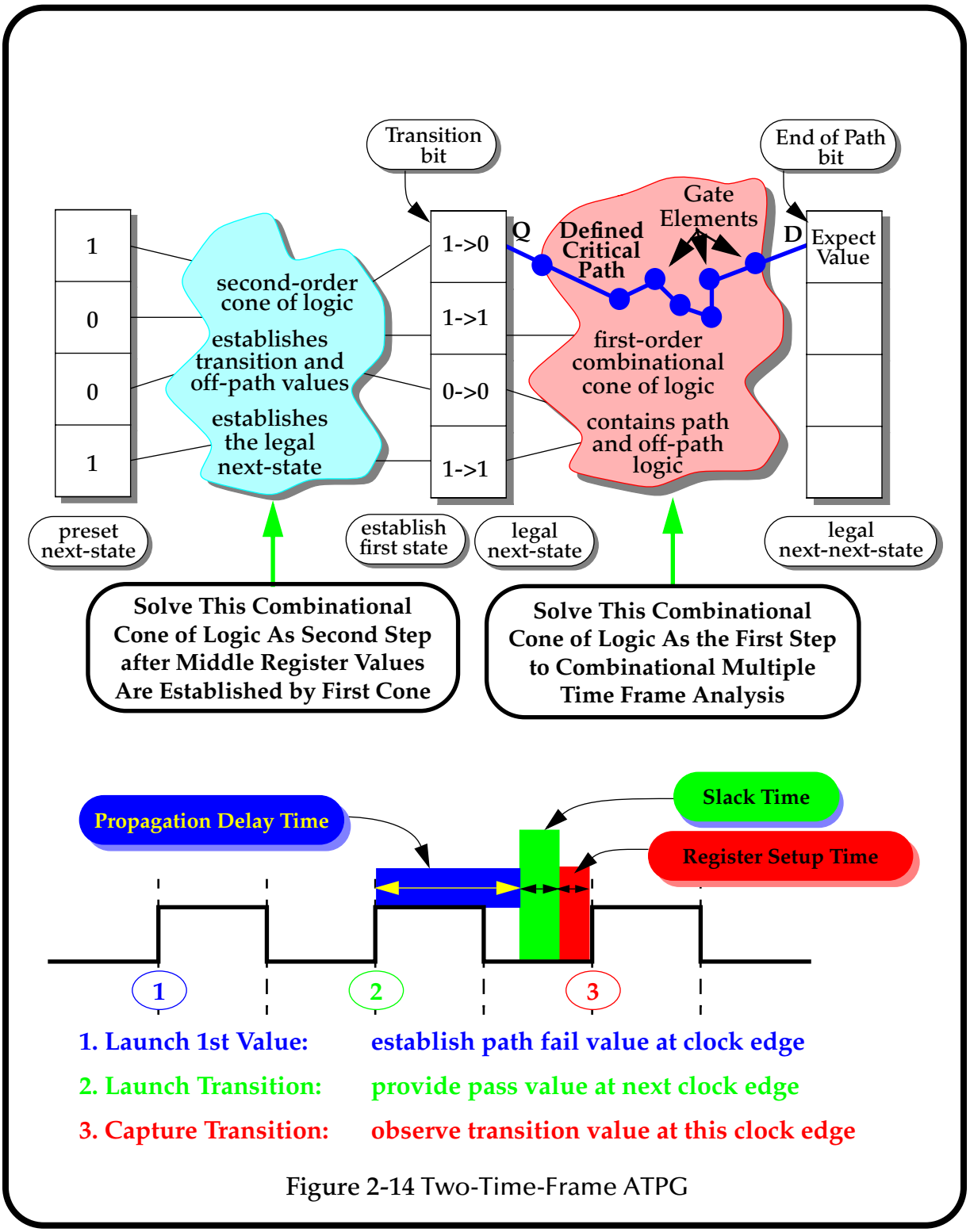
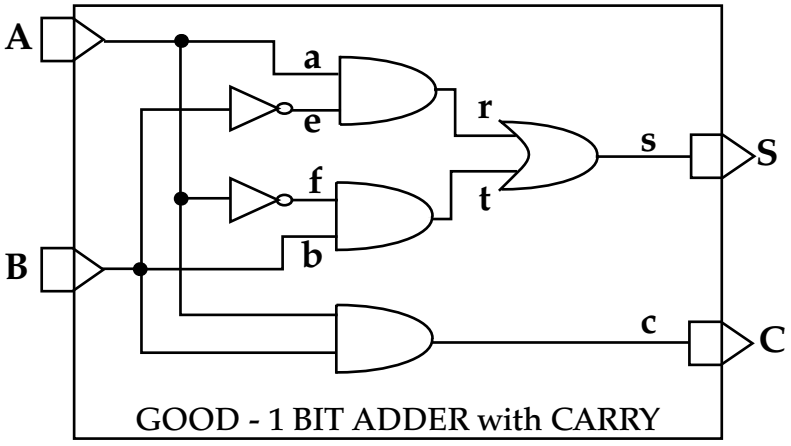
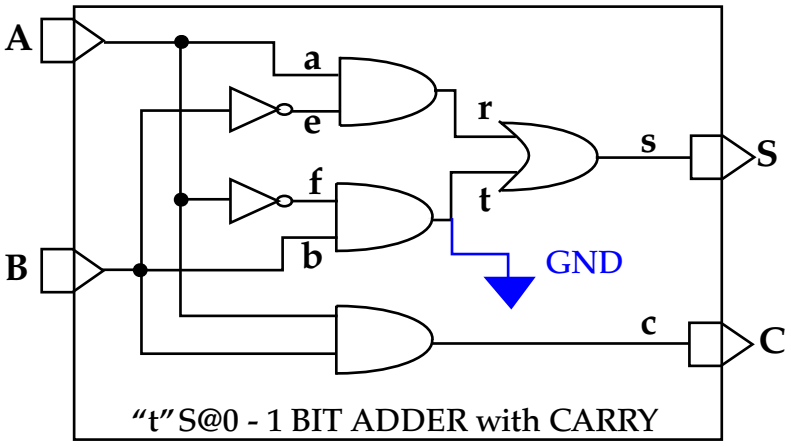
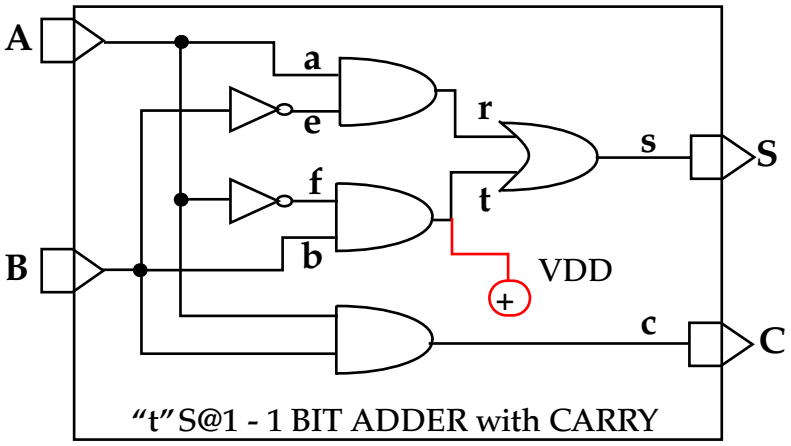


Figure 2-14 Two-Time-Frame ATPG



faultlist	
a @ 0	a @ 1
b @ 0	b @ 1
e @ 0	e @ 1
f @ 0	f @ 1
r @ 0	r @ 1
t @ 0	t @ 1
s @ 0	s @ 1
c @ 0	c @ 1
16 faults	





1. Create multiple copies of the netlist for each fault.
2. Apply same vectors to each copy.
3. Compare each copy to good simulation (expected response).
4. Fault is detected if bad circuit and good circuit differ at a detect point.
5. Measurement is faults detected divided by total number of faults (8/16 = 50%).

Figure 2-15 Fault Simulation example

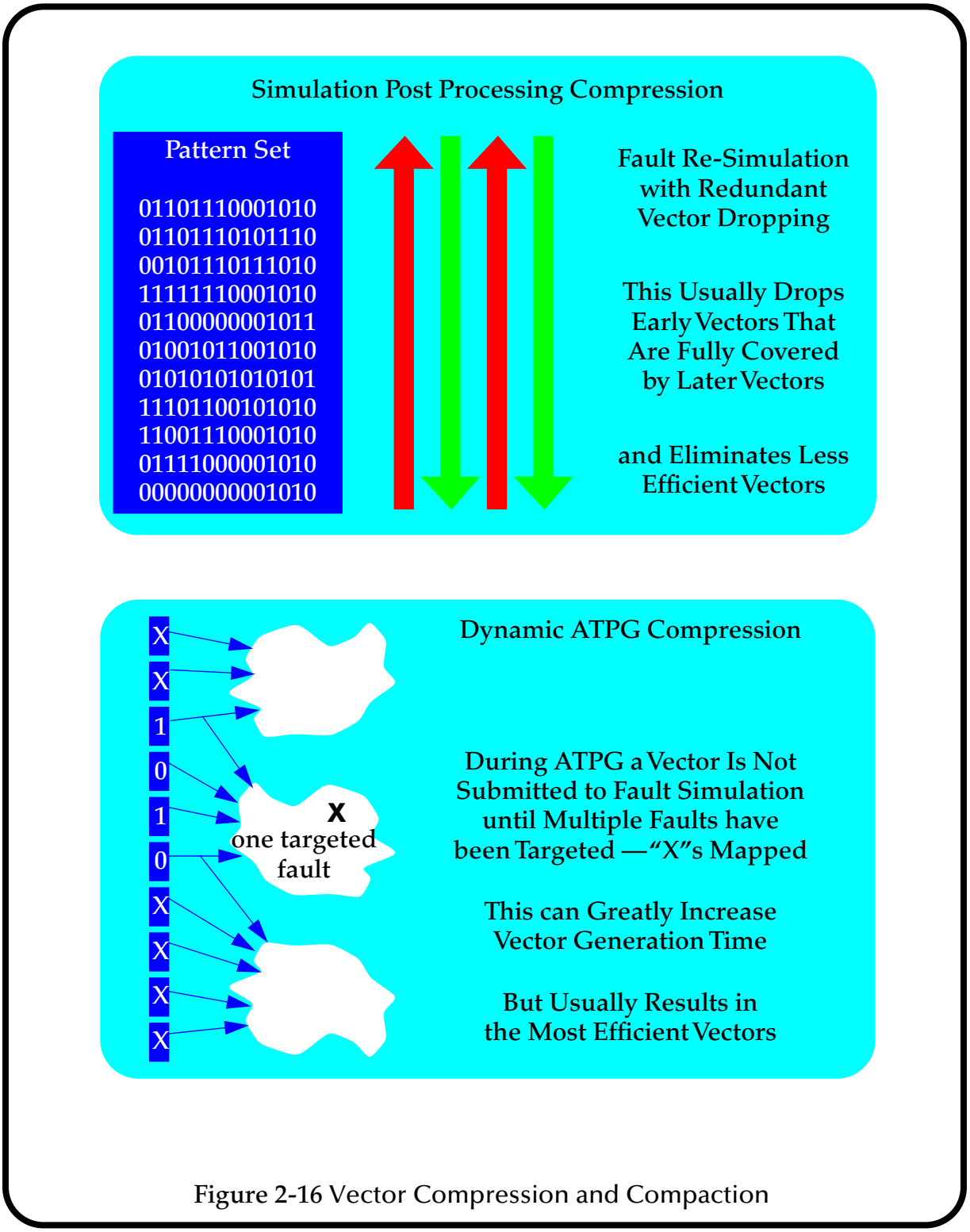


Figure 2-16 Vector Compression and Compaction

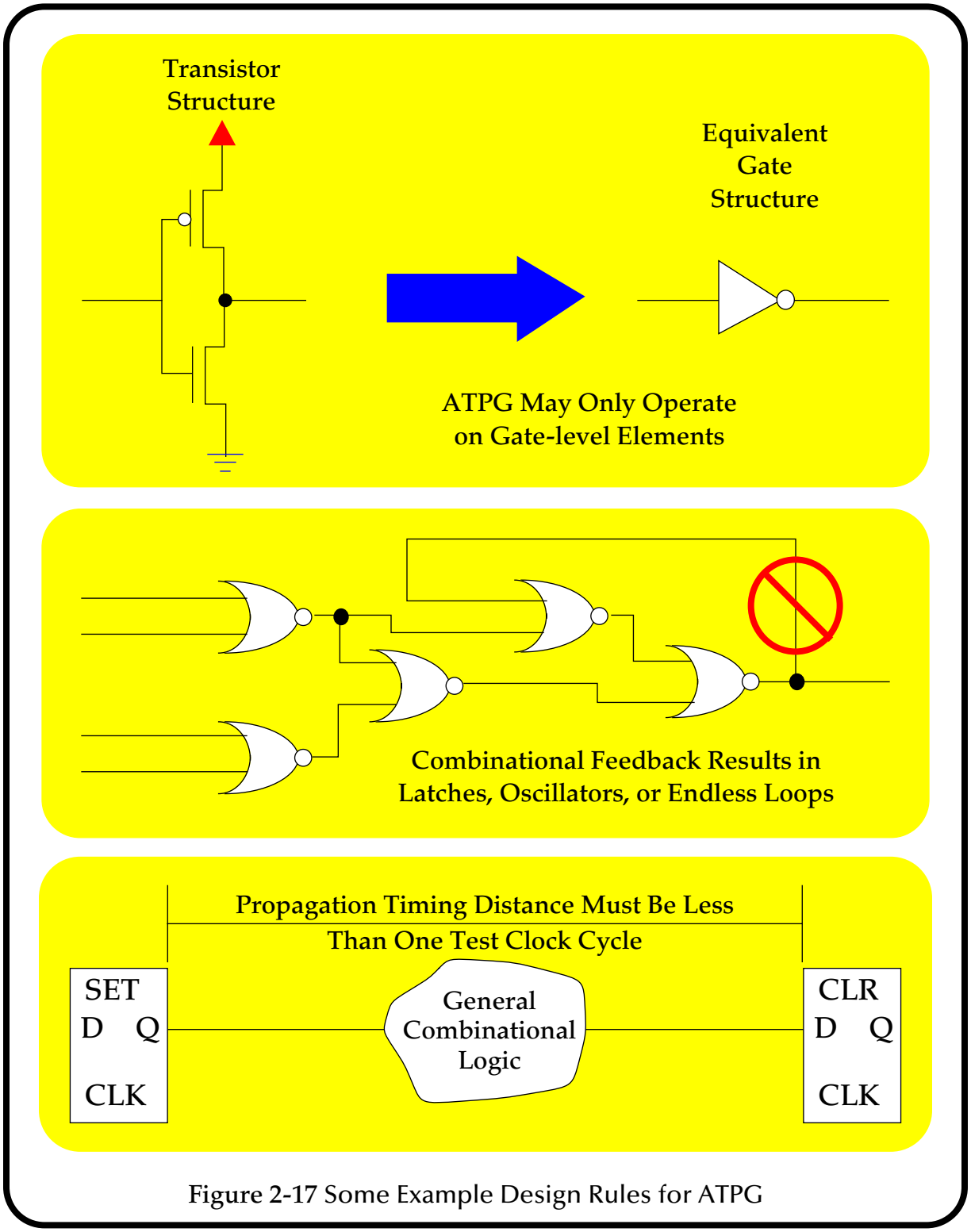


Figure 2-17 Some Example Design Rules for ATPG

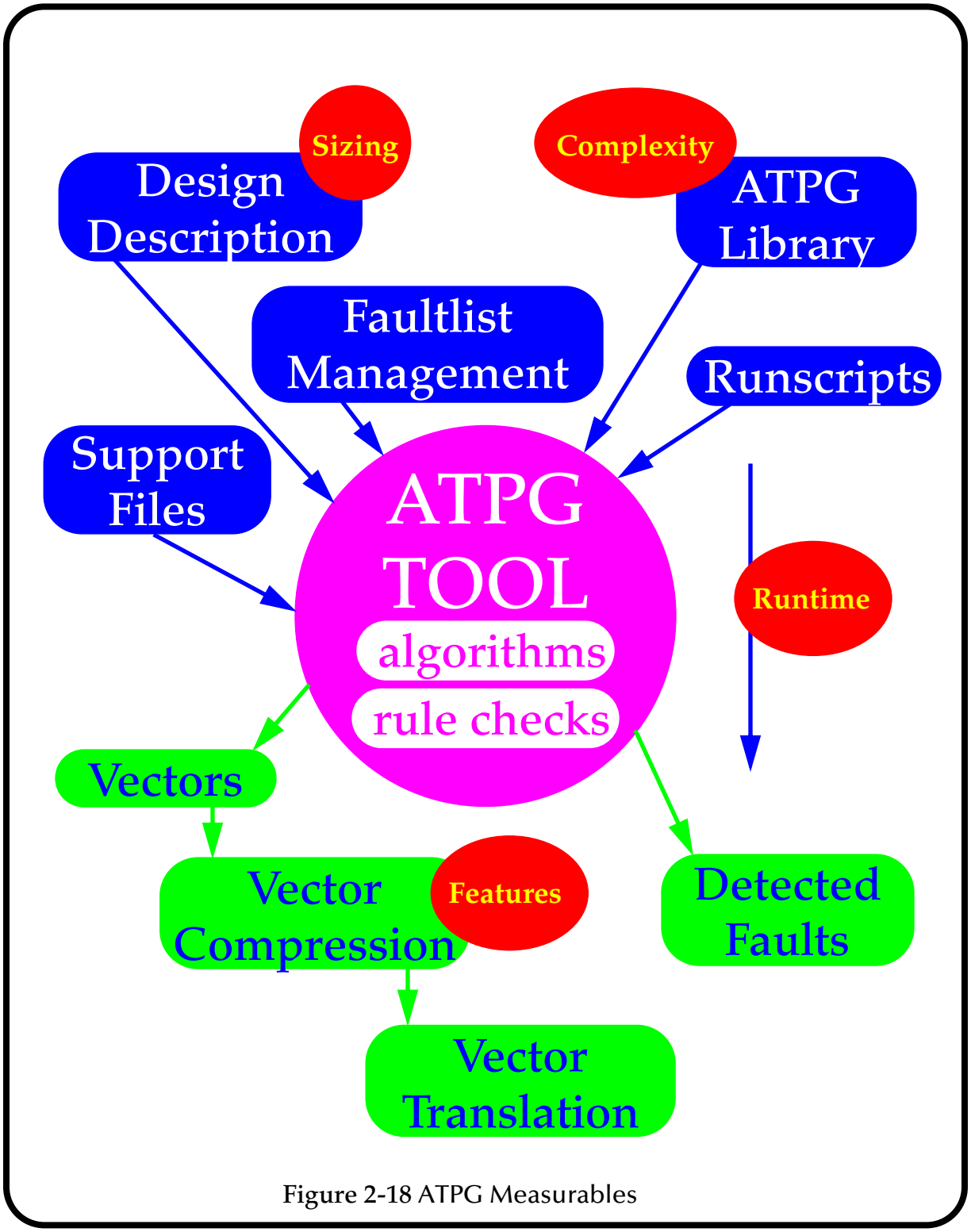


Figure 2-18 ATPG Measurables