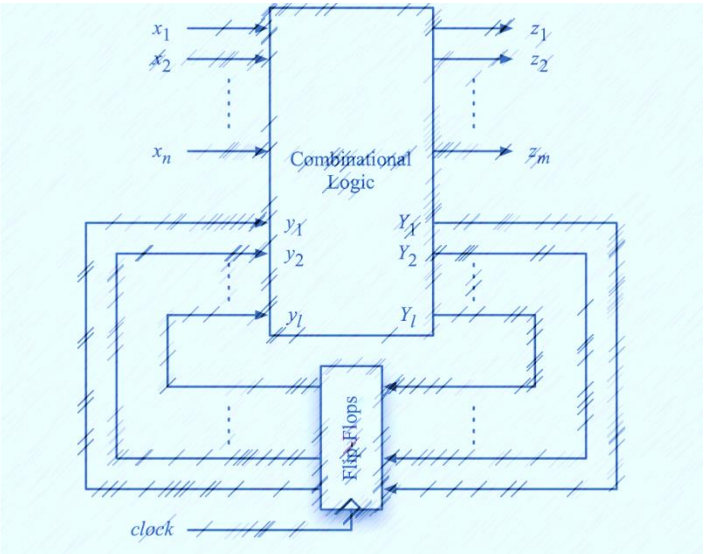
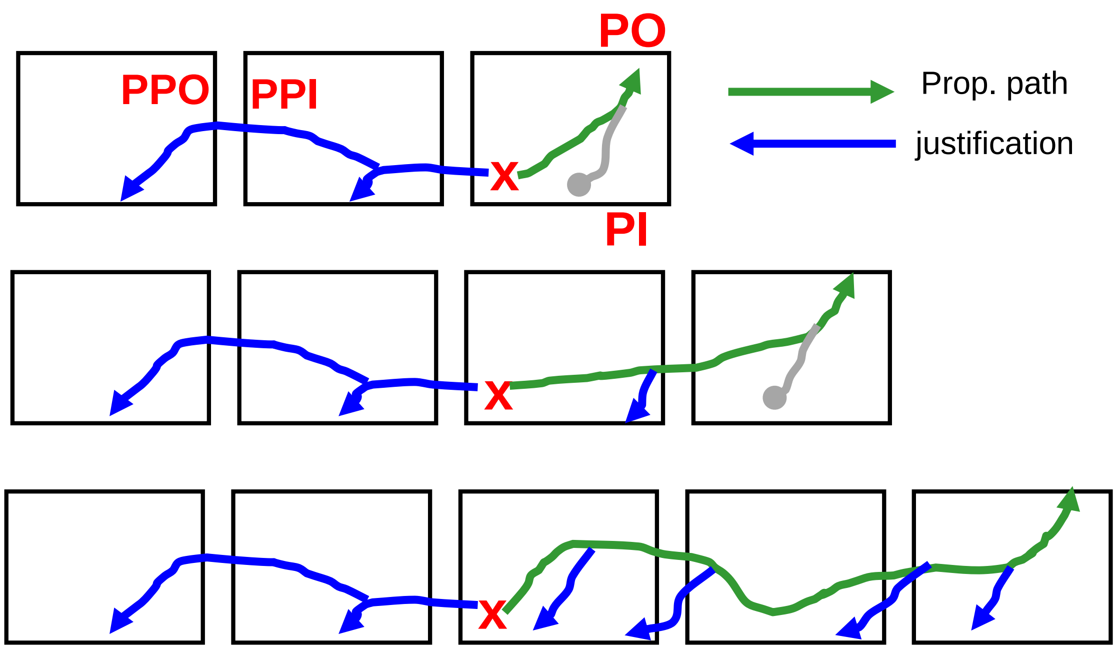
Sequential ATPG

- Introduction
- **Time-frame expansion methods**
  - The Extended D-algorithm [Kubo 68]
  - 9-valued D algorithm [Muth 76]
  - <u>**Backward Time Frame Processing* (not in exam)**</u>
    - <u>**EBT [Marlett 78]**</u>
    - **BACK [Cheng 88]**
  - Simulation-based methods*
- Issues of Sequential ATPG*
- Conclusions



Problems of Ext. D-Algorithm

- *Mixed Forward and Backward Time Frame Processing*
  - **Reuse existing D-algorithm**
    - **both forward fault propagation and backward justification**
  - **How many time frames ?**
    - **Memory requirement hard to predict**



Justification fails,

Backtrack!

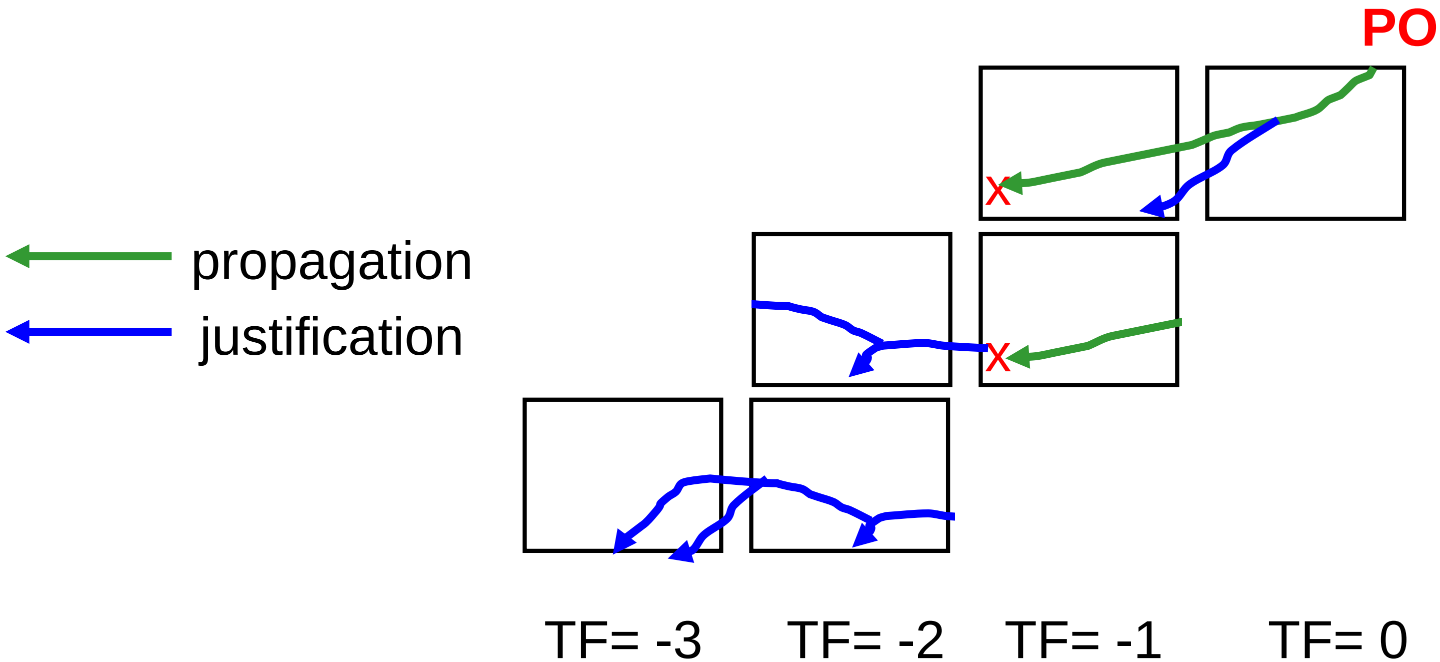
Justification fails,

Backtrack!

Test generated

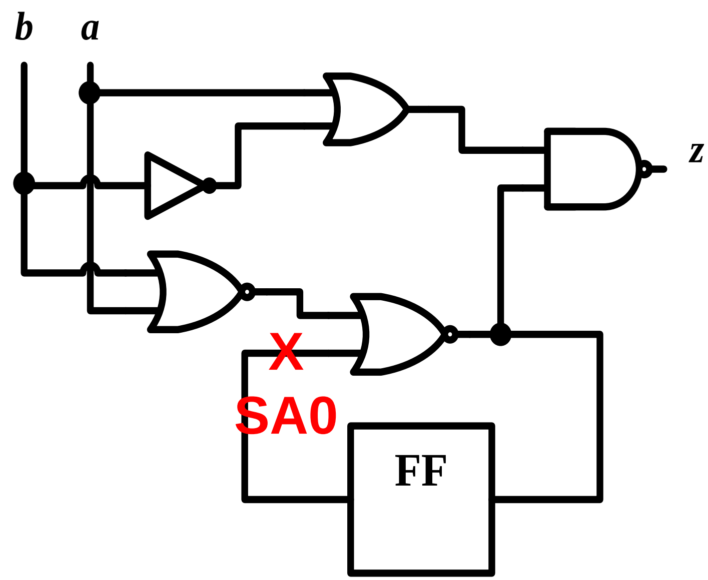
Extended Backtrace (EBT) [Marlett 78]

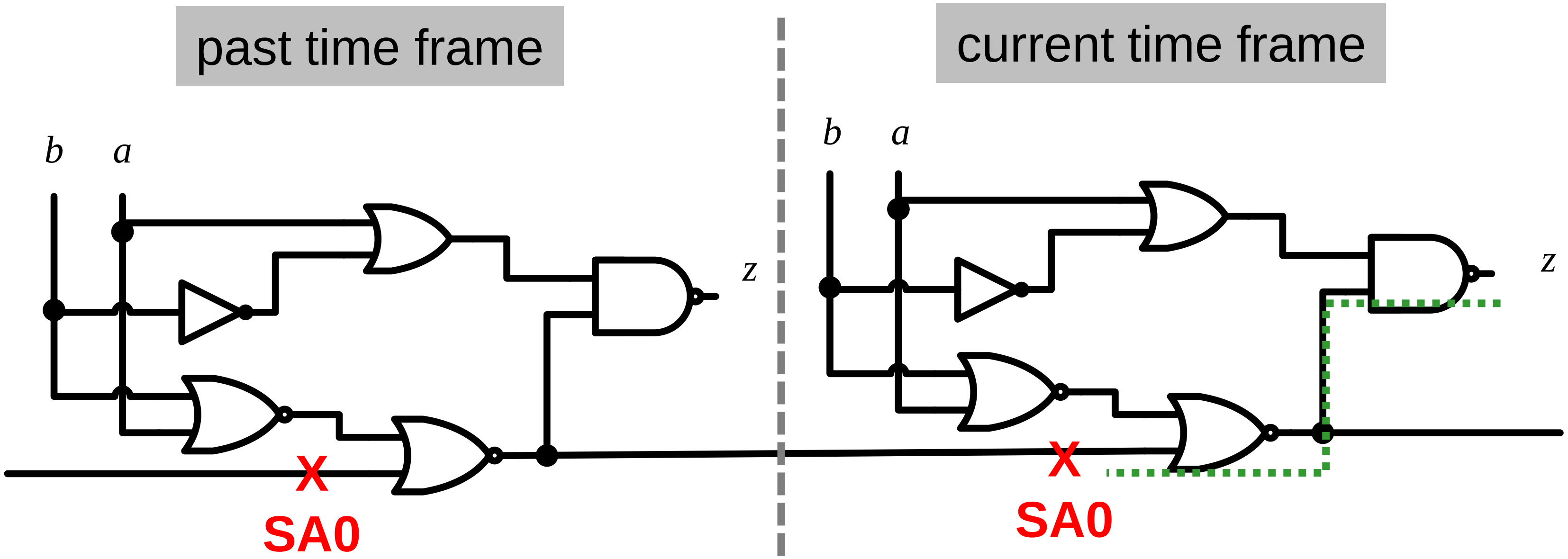
- *Backward Time Frame Processing* Only
  - 1. Select a path from fault site to PO
  - 2. Sensitize path backwards from the PO
  - 3. Justify required values backward
    - If justification fails, choose another path
- Advantage: only two time frames needed

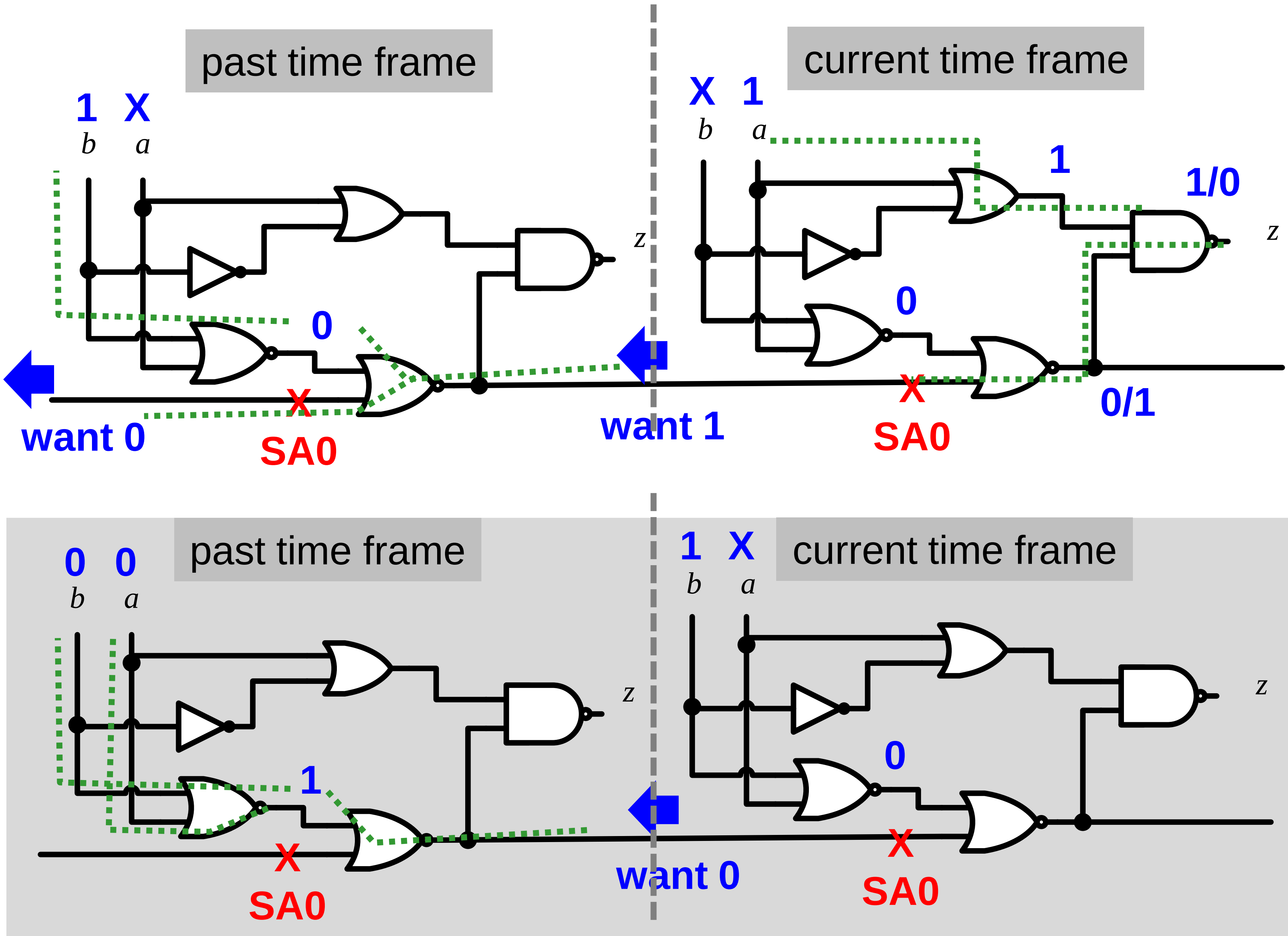


EBT Example

1. Create two time frames
2. Choose a path



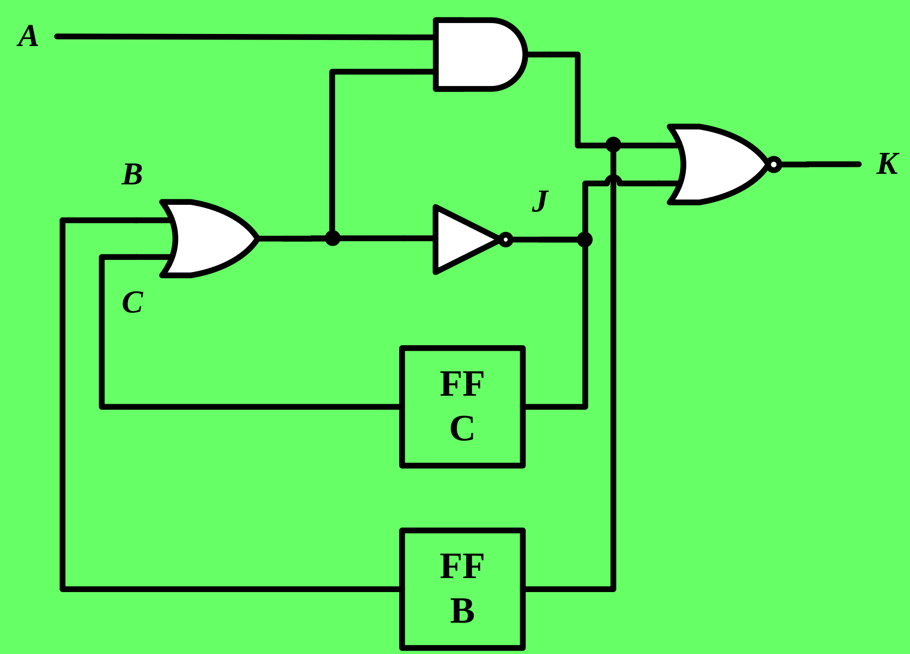



past time frame
1 X
b a
z
0
X
want 0
SA0
want 1
current time frame
X 1
b a
1
1/0
z
0
X
0/1
SA0
past time frame
0 0
b a
z
1
X
SA0
want 0
1 X
b a
current time frame
z
0
X
SA0

ab = 00, 1x, x1

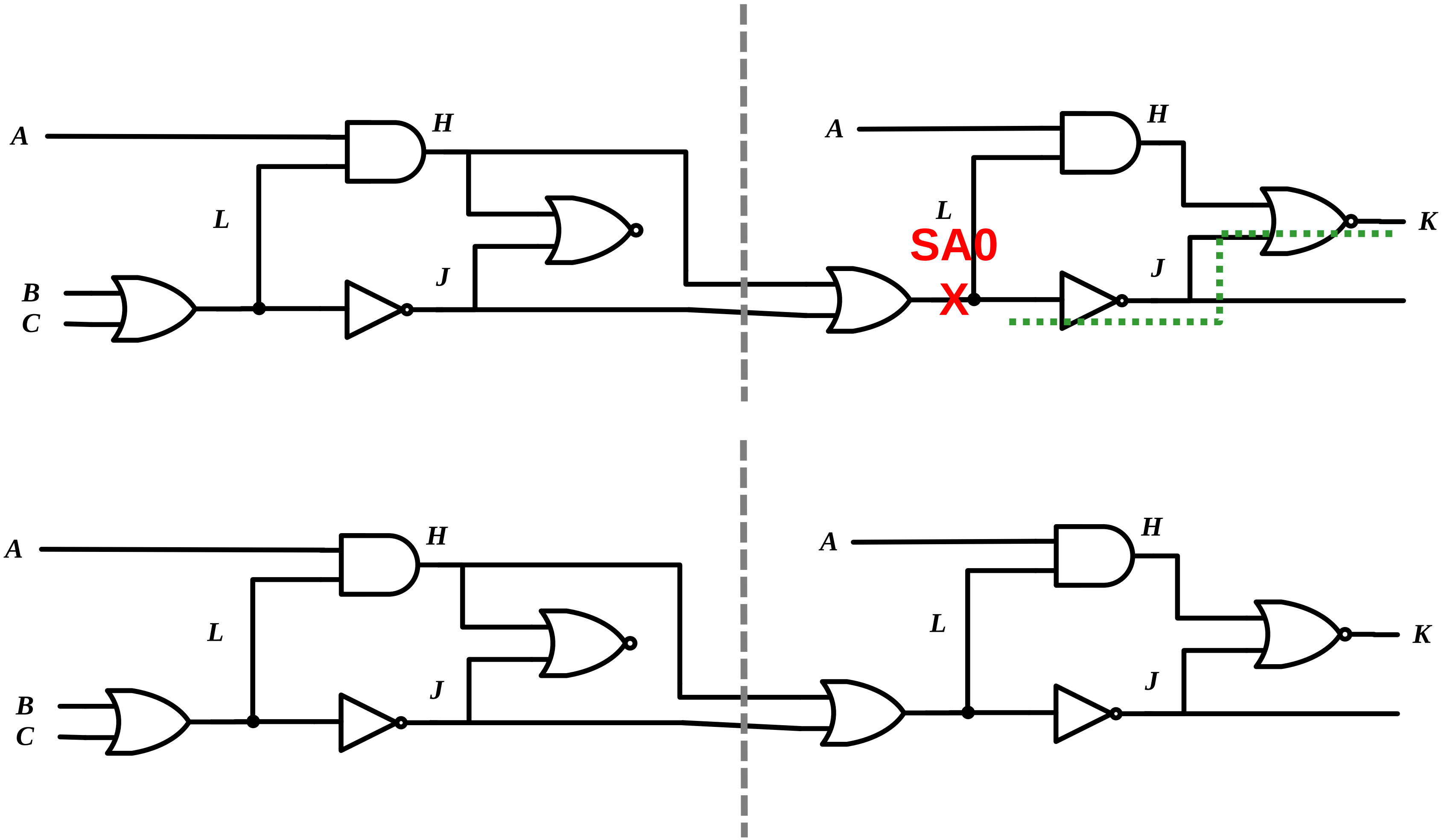
Quiz

Q: Please redraw this into two time frames

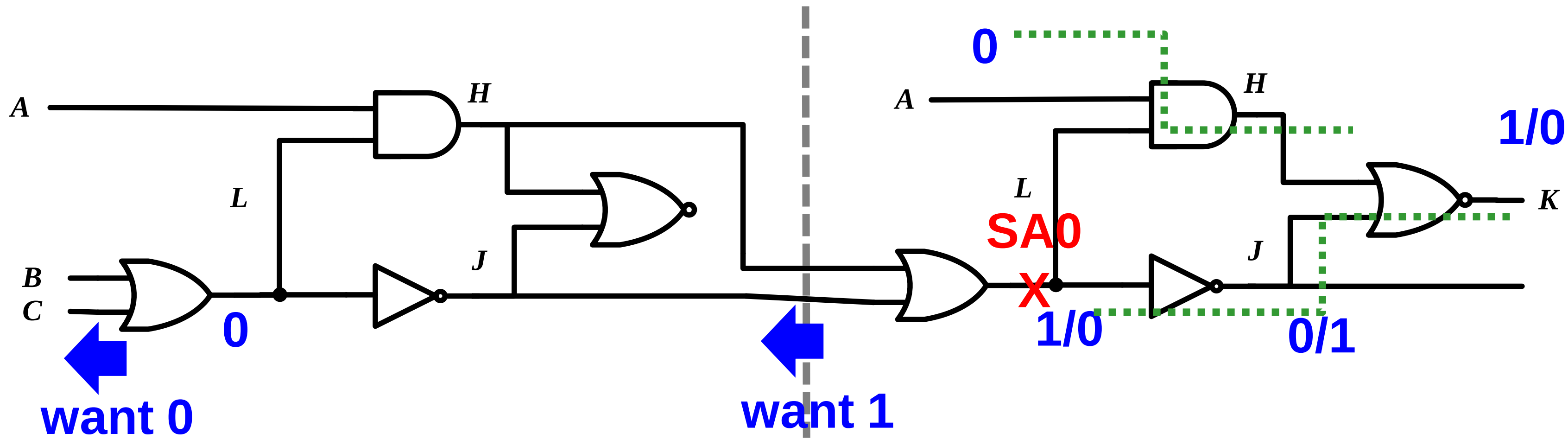


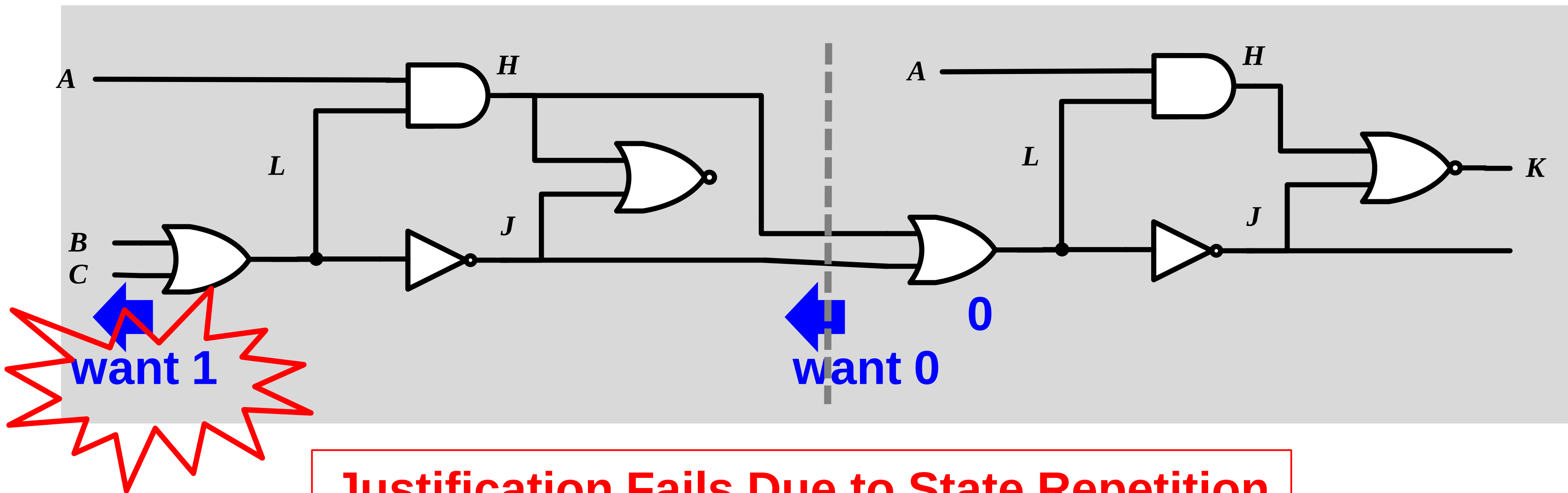
Quiz (cont'd)

Q: Use EBT to generate test patterns for SA0 fault



State Repetition





Justification Fails Due to State Repetition

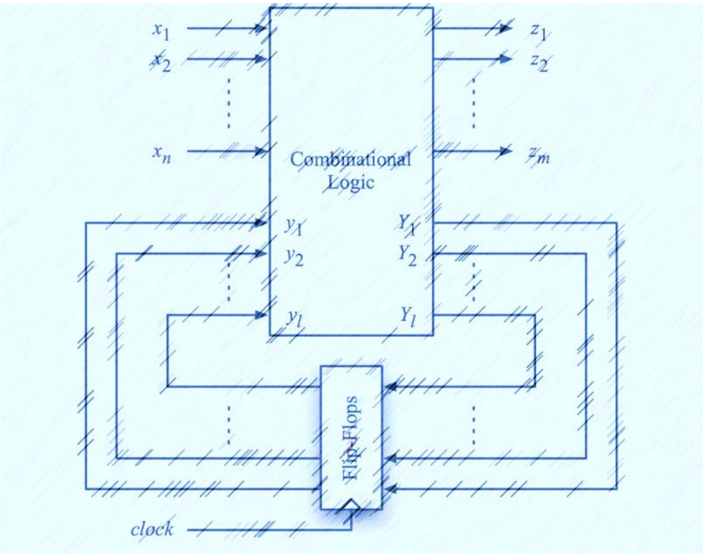
Comparison

	Advantages	Disadvantages	Example
Mixed F/B Time Frame	☺Reuse existing D algorithm	☹Too many time frames	Extended D
Backward TF Only	☺Fixed time frames ☺State repetition recognized	☹ Too many paths! How to choose path?	EBT

Need Help to Make Smart Decision

Sequential ATPG

- Introduction
- Time-frame expansion methods
  - The Extended D-algorithm [Kubo 68]
  - 9-valued D algorithm [Muth 76]
  - Backward Time Frame Processing* (not in exam)
    - EBT [Marlett 78]
    - BACK [Cheng 88]
  - Simulation-based methods*
- Issues of Sequential ATPG*
- Conclusions



BACK Algorithm [Cheng 88]

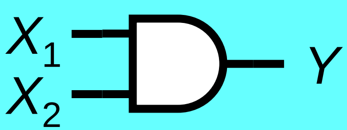
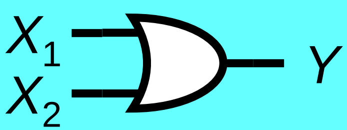
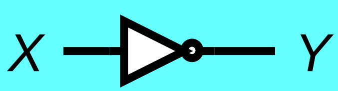
- BACK selects a *PO* for fault detection,
  - Do not explicitly select a *path*
- PO selection based on a testability measures
  - *Drivability*

- Sensitized path will be created implicitly when drivability calculation

BACK Chooses PO Instead of Path

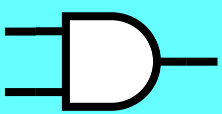
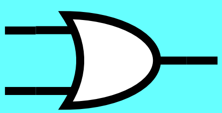
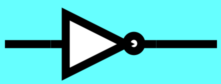
Review: SCOAP (CH 6)

- *$CC^0(N)$, $CC^1(N)$*
  - **Minimum number of combinational PI assignments and logic levels required to control a 0 or a 1 on node *N***
  - **Estimates effort to control signal to zero or one**
    - ***Smaller* number, *easier* to control**

	$CC^0(Y)$	$CC^1(Y)$
	$\min[CC^0(X_1), CC^0(X_2)] + 1$	$CC^1(X_1) + CC^1(X_2) + 1$
	$CC^0(X_1) + CC^0(X_2) + 1$	$\min[CC^1(X_1), CC^1(X_2)] + 1$
	$CC^1(X) + 1$	$CC^0(X) + 1$
Primary inputs	1	1

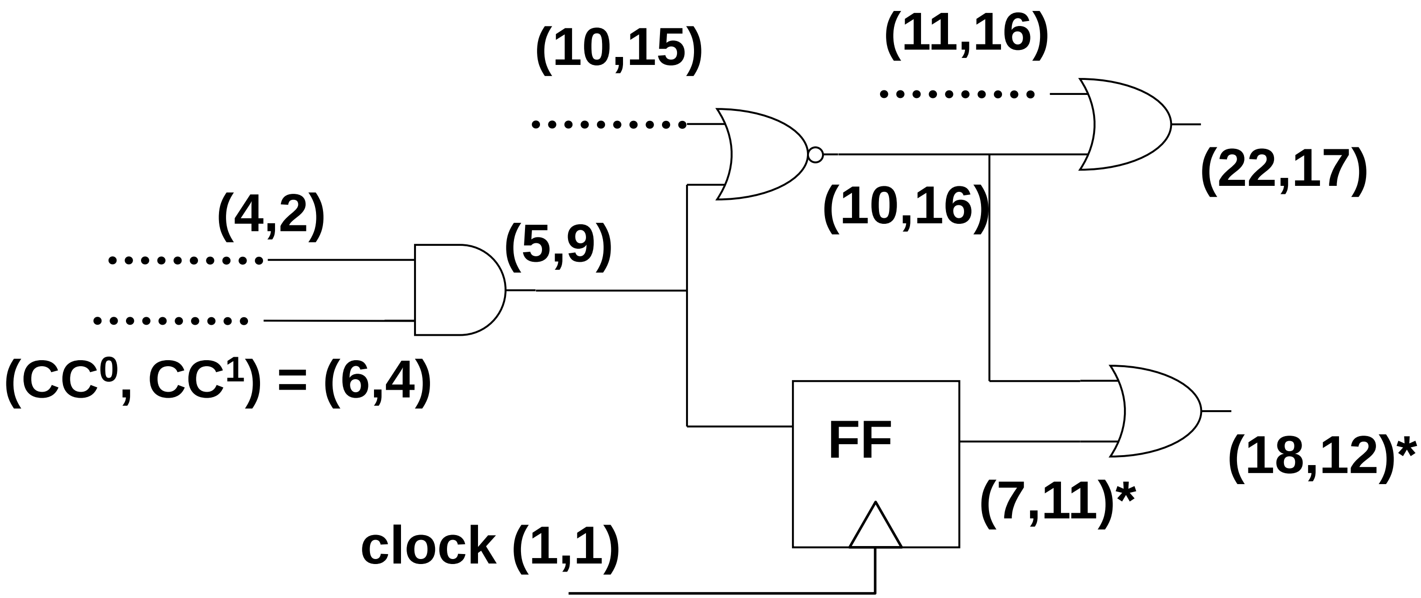
Drivability: $d^{0/1}$, $d^{1/0}$

- Estimates **effort** to propagate a **D or D'** from fault site to Y
  - $d^{0/1}(Y)$ = *drivability of D'* to node Y
  - $d^{1/0}(Y)$ = *drivability of D* to node Y

	$d^{0/1}(Y)$	$d^{1/0}(Y)$
X_1, X_2  Y	$\min\{CC^1(X_1) + d^{0/1}(X_2),$ $d^{0/1}(X_1)+CC^1(X_2),$ $d^{0/1}(X_1)+d^{0/1}(X_2)\}+1$	$\min\{CC^1(X_1)+d^{1/0}(X_2),$ $d^{1/0}(X_1)+CC^1(X_2),$ $d^{1/0}(X_1)+d^{1/0}(X_2)\}+1$
X_1, X_2  Y	$\min\{CC^0(X_1) + d^{0/1}(X_2),$ $d^{0/1}(X_1)+CC^0(X_2),$ $d^{0/1}(X_1)+d^{0/1}(X_2)\}+1$	$\min\{CC^0(X_1)+d^{1/0}(X_2),$ $d^{1/0}(X_1)+CC^0(X_2),$ $d^{1/0}(X_1)+d^{1/0}(X_2)\}+1$
X  Y	$d^{1/0}(X) + 1$	$d^{0/1}(X) + 1$
Y = FF out X = FF in	$d^{0/1}(X)$ + K (constant)	$d^{1/0}(X)$ + K (constant)

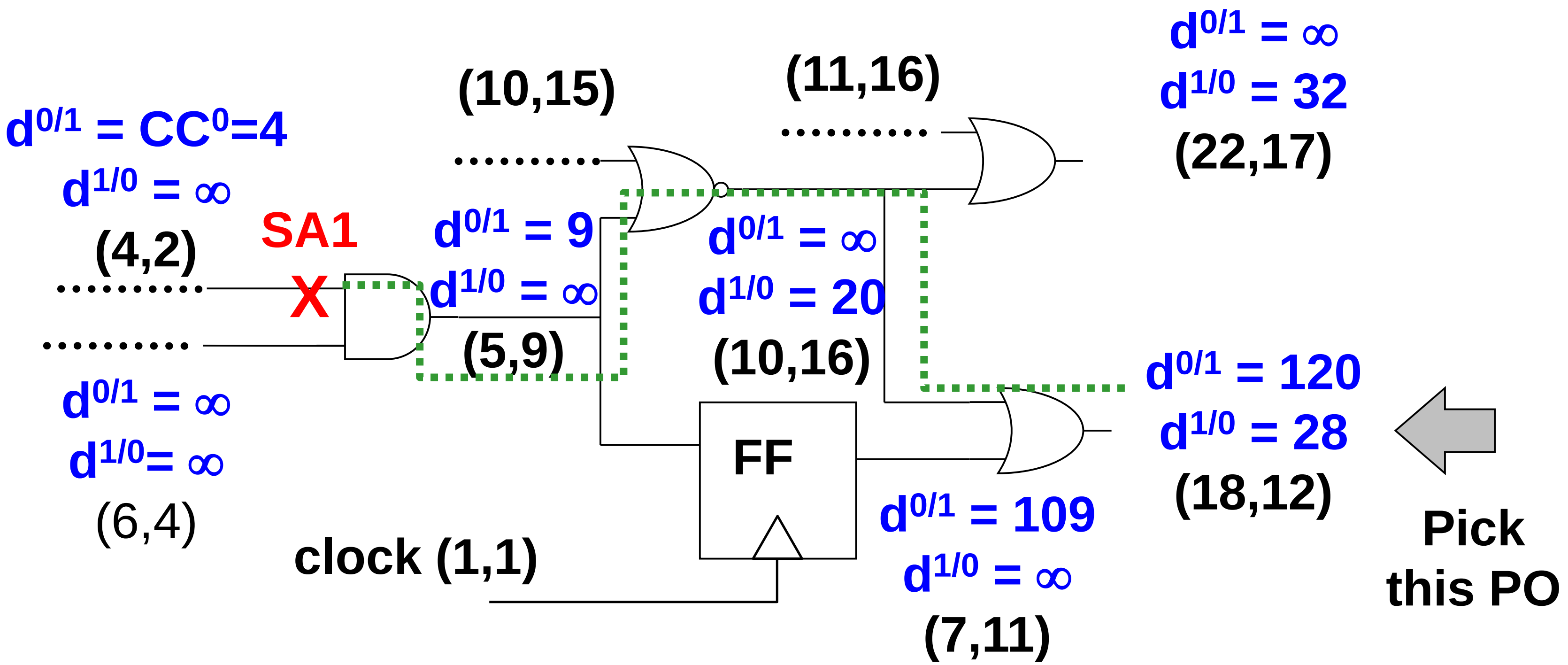
Example 1/2: SCOAP (BA Fig. 8.7)

- **Fault-free circuit**



* different from textbook

Example 2/2: Drivability (BA Fig. 8.7)

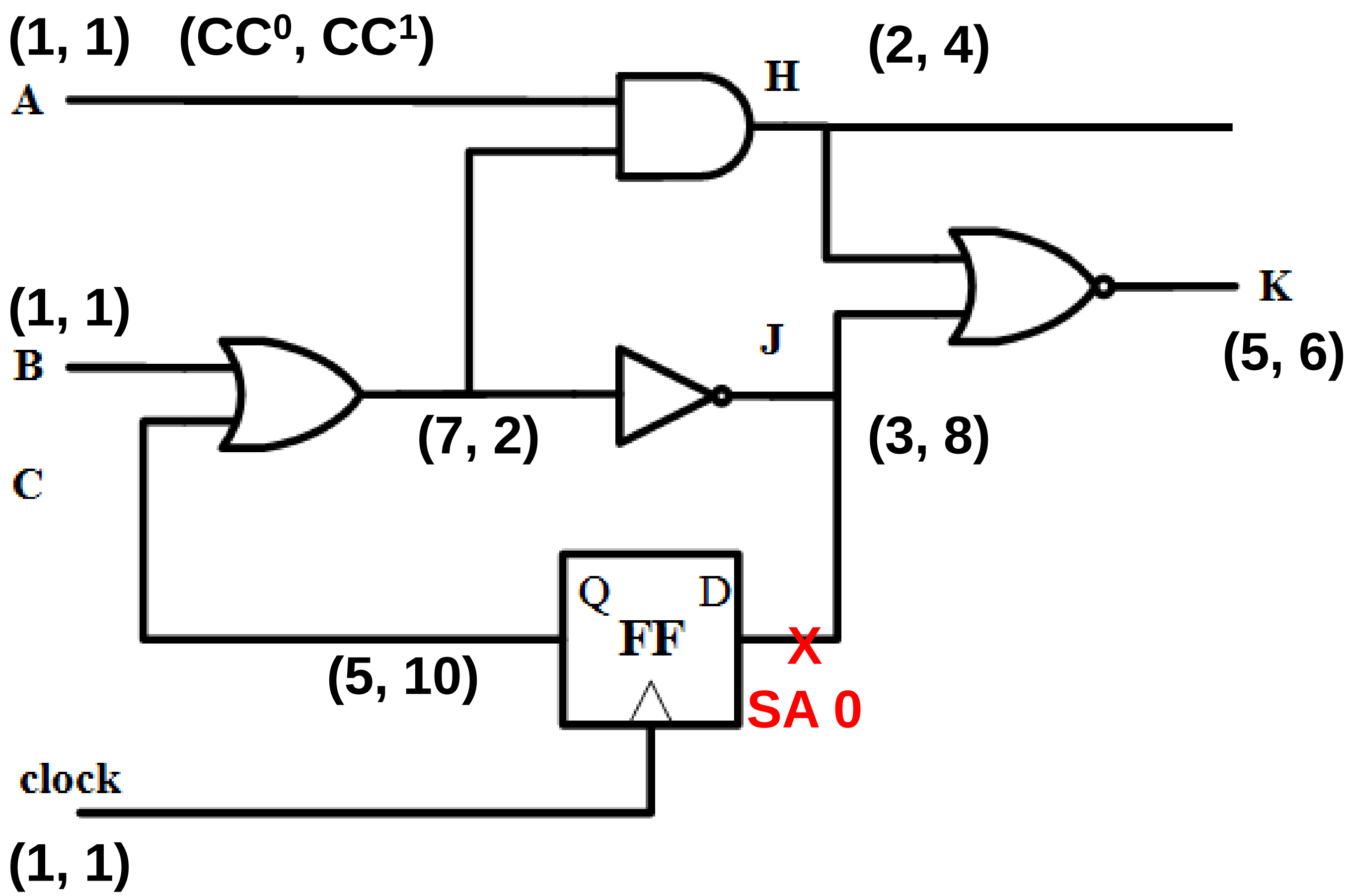


K=100

When D impossible, $d^{1/0} = \infty$

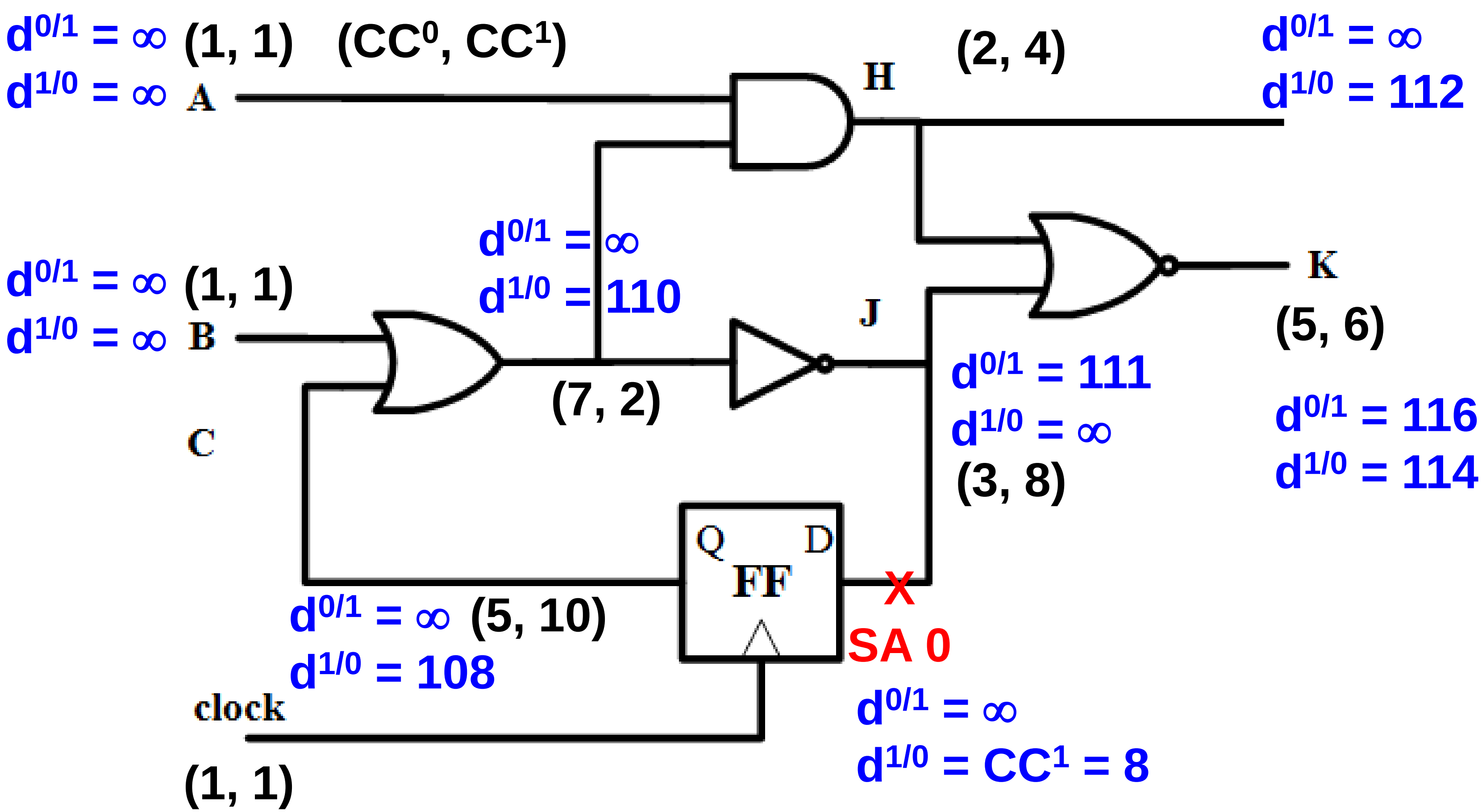
Quiz

Q: Based on SCOAP, please calculate drivability of SA0 fault. (Assume K=100)



Quiz

Q: Based on SCOAP, please calculate drivability of SA0 fault.
(Assume K=100)



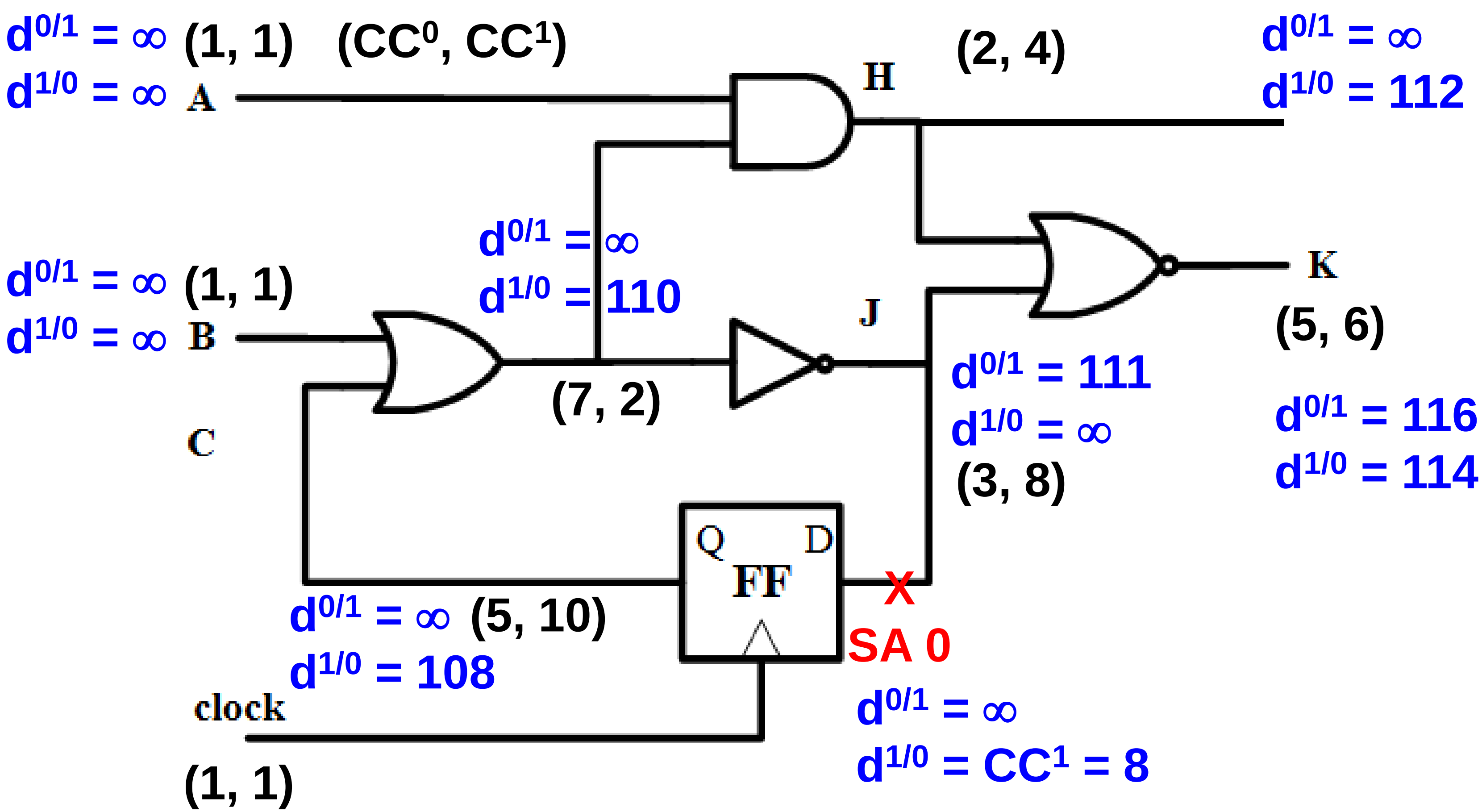
Summary

- *Backward time frame processing* has advantages:
  - Fixed time frame
  - State repetition recognized
- EBT selects one path at a time
- BACK proposes drivability to select best PO

	Advantages	Disadvantages	Example
Mixed F/B Time Frame	☺Reuse existing alg.	☹ Too many time frames	Extended D
Backward TF Only	☺Fixed time frames ☺State repetition recognized	☹ Too many paths!	EBT BACK
Forward TF Only	☺ No need to justify	☹ Hard to decide time frames	FASTEST (PODEM-like)

FFT1

- Do we need feedback to iteratively calculate Drivability?



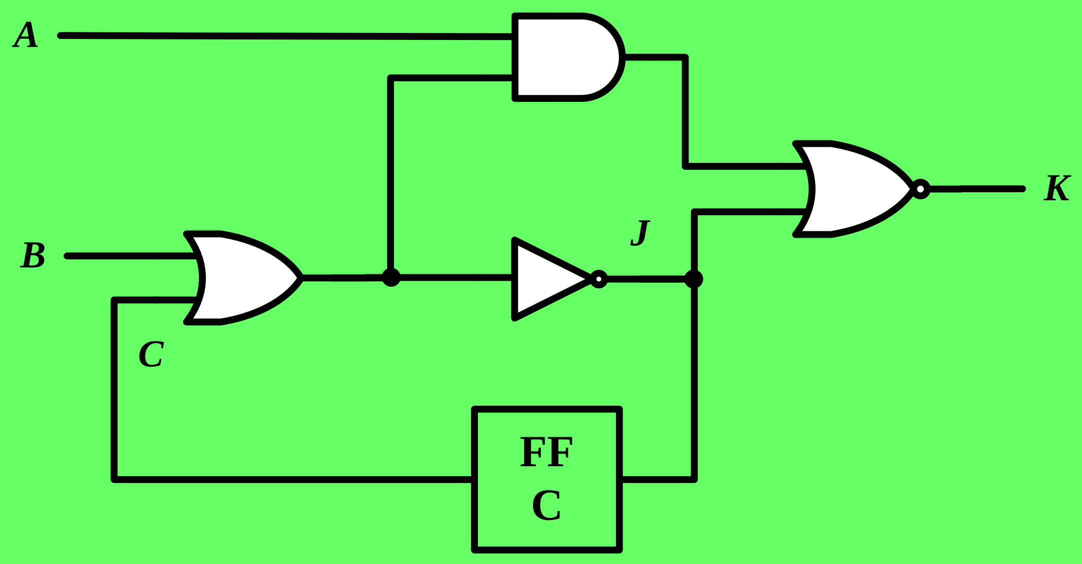
FFT2

- Q: For forward time frame processing, can we fix time frames, like EBT?

	Advantages	Disadvantages	Example
Mixed F/B Time Frame	☺Reuse existing alg.	☹ Too many time frames	Extended D
Backward TF Only	☺Fixed time frames ☺State repetition recognized	☹ Too many paths!	EBT BACK
Forward TF Only	☺ No need to justify	☹ Hard to decide time frames	PODEM-like

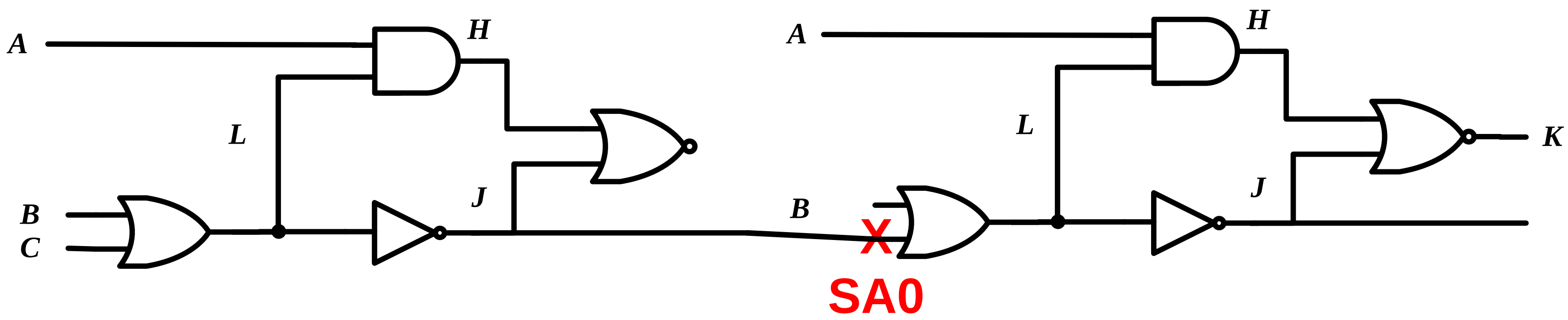
Quiz

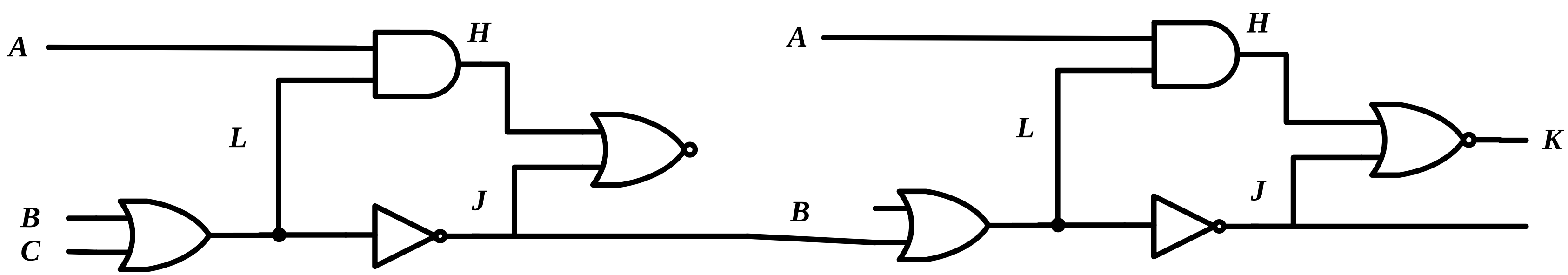
Q: Please redraw this into two time frames



Quiz (cont'd)

Q: Use EBT to generate test patterns for SA0 fault





Drivability

- *Drivability* estimates effort of propagating a D or D' from fault site to that signal
  - Similar to *SCOAP*
- Review of SCOAP

	$CC^0(Y)$	$CC^1(Y)$
$Y = X_1$ AND X_2	$\min[CC^0(X_1), CC^0(X_2)] + 1$	$CC^1(X_1) + CC^1(X_2) + 1$
$Y = X_1$ OR X_2	$CC^0(X_1) + CC^0(X_2) + 1$	$\min[CC^1(X_1), CC^1(X_2)] + 1$
$Y = X'$	$CC^1(X) + 1$	$CC^0(X) + 1$
Primary inputs	1	1

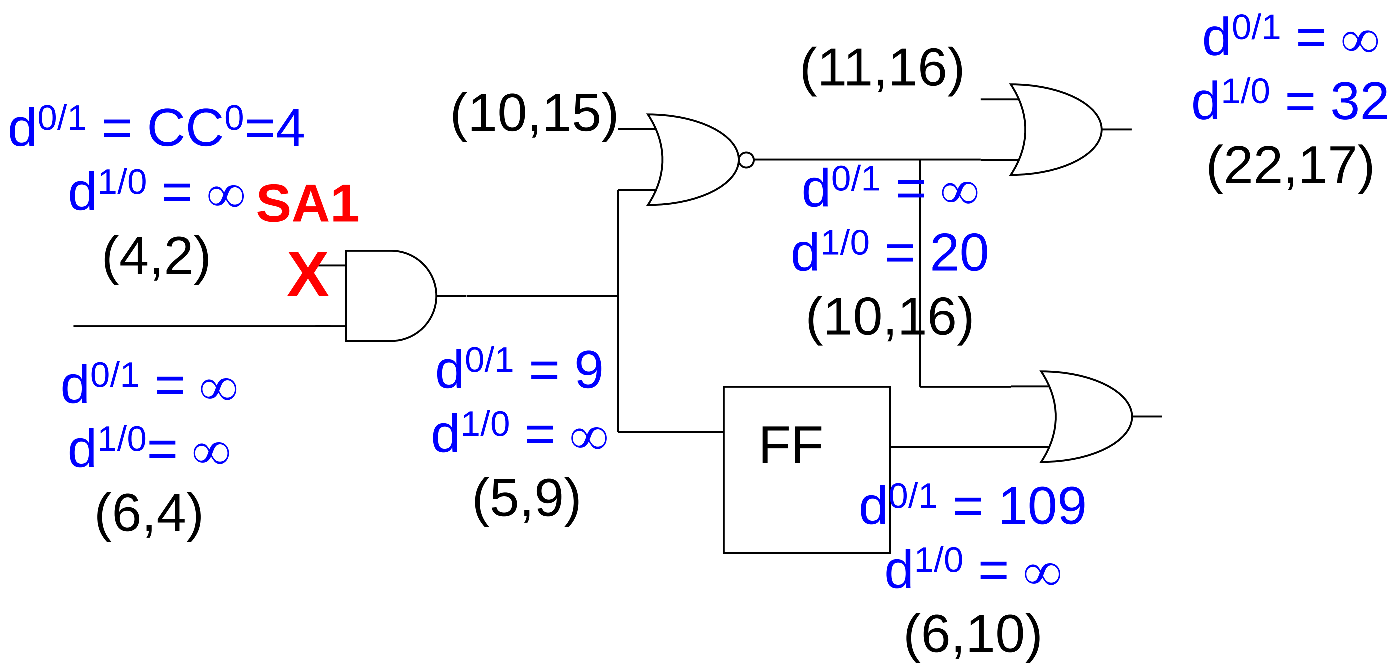
Drivability is like SCOAP for a Specific Fault

Drivability: d(0/1) & d(1/0)

- $d^{0/1}(Y)$ = *drivability of D'*
  - **Effort to bring node Y to D' when the fault presents**
- $d^{1/0}(Y)$ = *drivability of D*

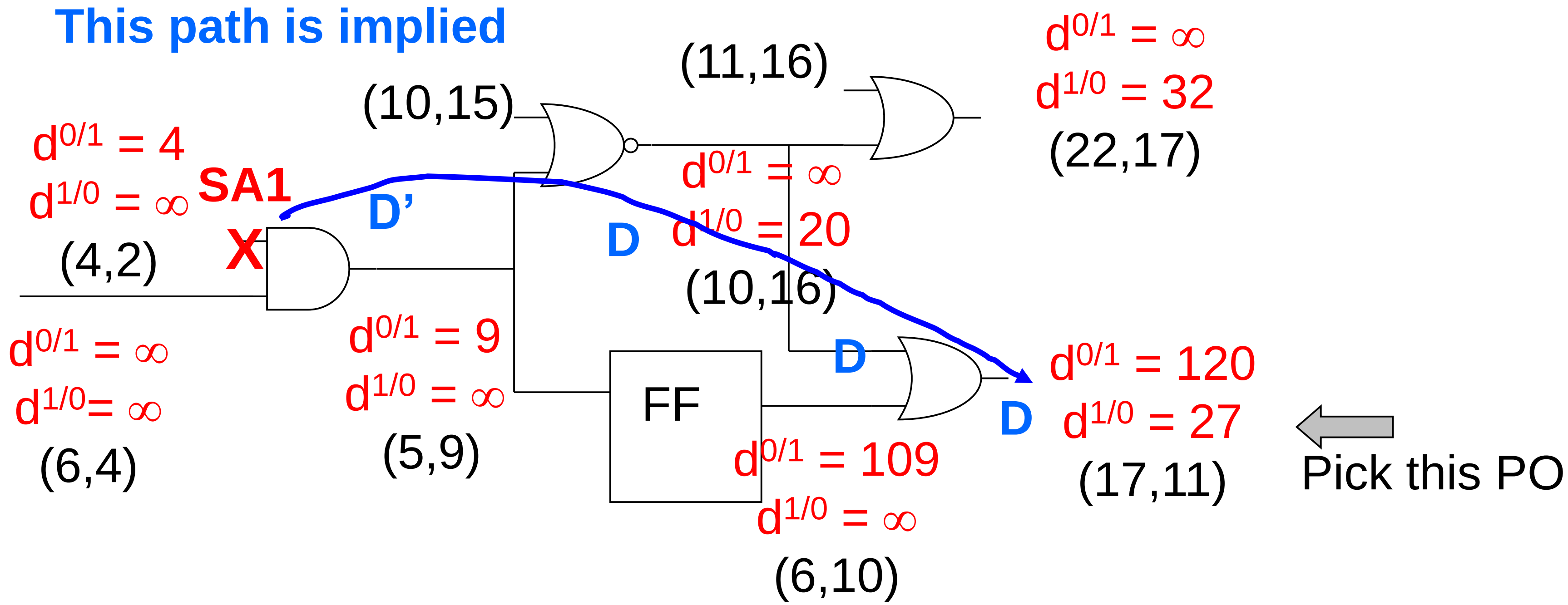
	$d^{0/1}(Y)$	$d^{1/0}(Y)$
Y = X1 AND X2	$\min\{CC^1(X_1) + d^{0/1}(X_2),$ $d^{0/1}(X_1)+CC^1(X_2),$ $d^{0/1}(X_1)+d^{0/1}(X_2)\}+1$	$\min\{CC^1(X_1)+d^{1/0}(X_2),$ $d^{1/0}(X_1)+CC^1(X_2),$ $d^{1/0}(X_1)+d^{1/0}(X_2)\}+1$
Y = X1 OR X2	$\min\{CC^0(X_1) + d^{0/1}(X_2),$ $d^{0/1}(X_1)+CC^0(X_2),$ $d^{0/1}(X_1)+d^{0/1}(X_2)\}+1$	$\min\{CC^0(X_1)+d^{1/0}(X_2),$ $d^{1/0}(X_1)+CC^0(X_2),$ $d^{1/0}(X_1)+d^{1/0}(X_2)\}+1$
Y = X'	$d^{1/0}(X) + 1$	$d^{0/1}(X) + 1$
Y = FF out X = FF in	$d^{0/1}(X)$ + K (constant)	$d^{1/0}(X)$ + K (constant)

Drivability Example 1/2 (BA Fig. 8.7)



- **K=100**
- **When D impossible, $d^{1/0} = \infty$**

Drivability Example 2/2 (BA Fig. 8.7)



- **To justify a D or D' at gate output, select gate input with smallest drivability as D or D' input**