COE 571 Digital System Testing

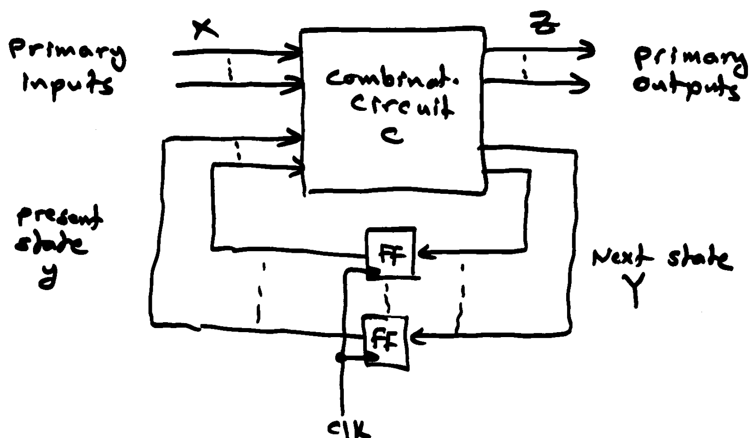
Dr. Aiman El-Maleh

Testing of Sequential Circuits

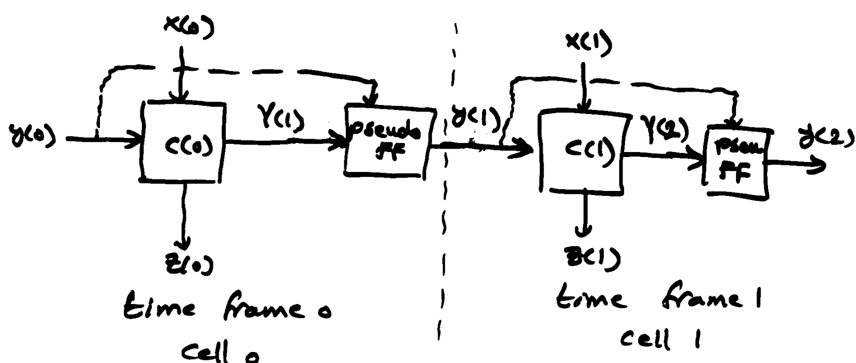
1. **Iterative array model**
2. **The unknown logic value**
3. **Initialization/Synchronization sequence**
4. **3-valued logic**
5. **Fault detection in sequential circuits**
6. **Sequential untestability and redundancy**
7. **Partially testable faults**
8. **C-cycle redundancy**
9. **Fault equivalence in sequential circuits**

Testing of Sequential Circuits

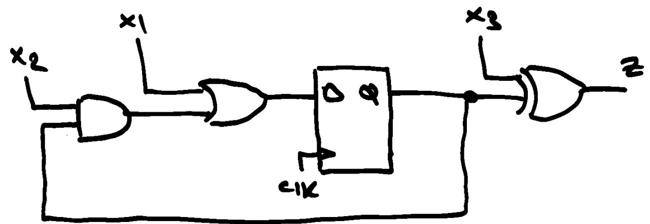
- A synchronous sequential circuit has the following canonical structure



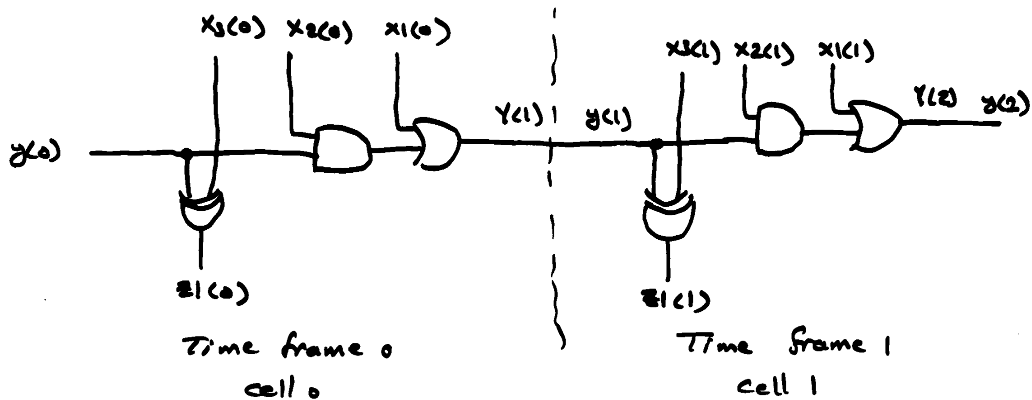
- For test generation, a synchronous sequential circuit S can be modeled by an iterative array model



- **Example:** Modeling a sequential circuit with DFFs



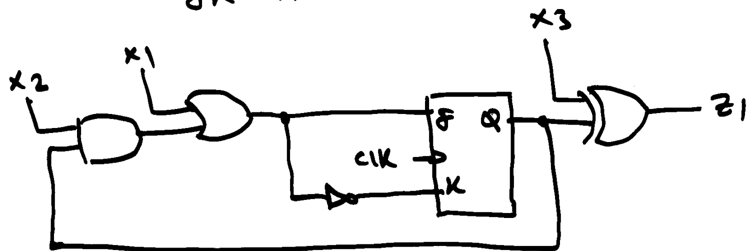
Iterative array model:



- This modeling technique maps the time domain response of the sequential circuit into a space domain response.

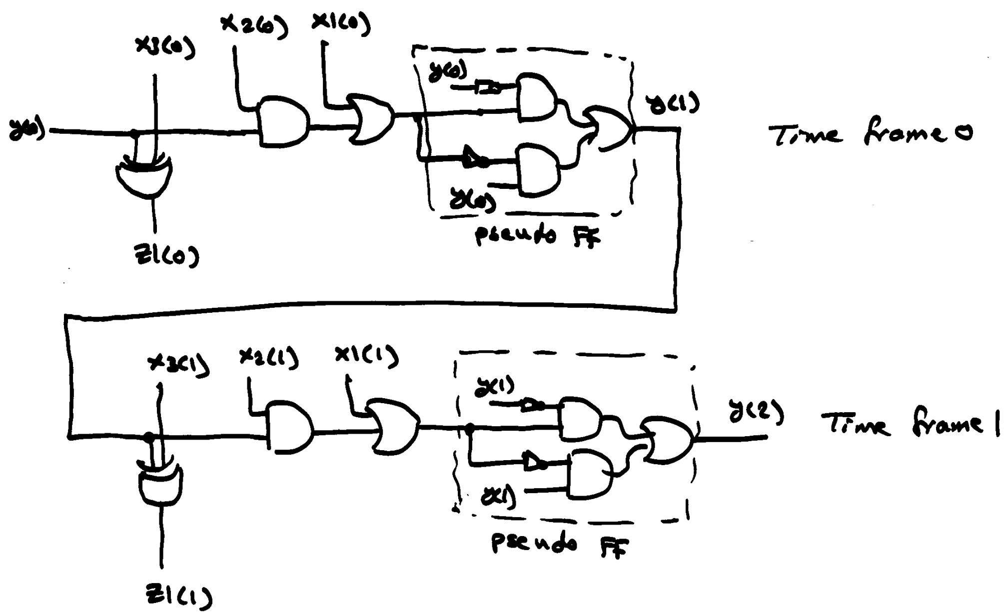
- It allows test generation methods developed for combinational circuits to be extended to synchronous sequential circuits.

<u>Example</u>: Modeling a sequential circuit with JK-FFs.



<u>Iterative array model</u>

$q^+ = \bar{q}J + q\bar{K}$



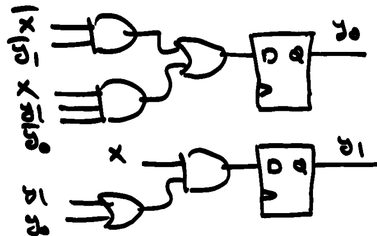
The Unknown Logic Value

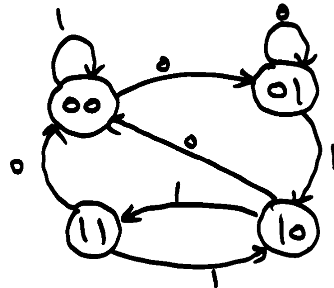
- The response of a sequential circuit to an input sequence depends on its initial state.

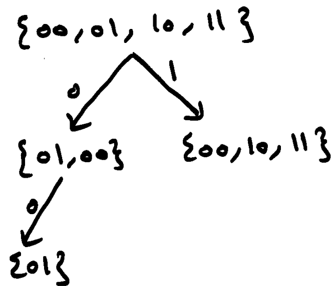
- When a circuit is powered-up, the initial state of memory elements such as FFs & RAMs is unpredictable.

- Before normal operation of a sequential circuit, an initialization sequence is applied to bring the circuit into a known reset state.

- Example





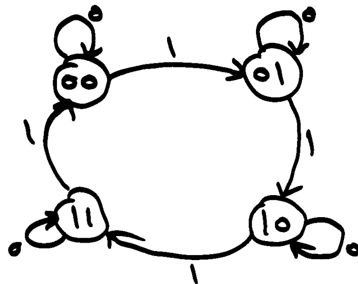


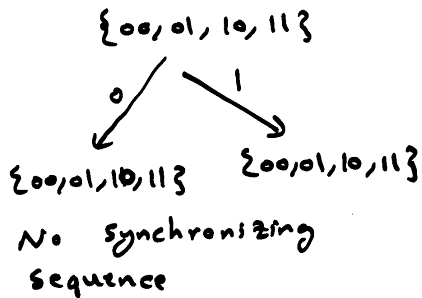
- A minimal length initialization (synchronizing) sequence is $x = \{0, 0\}$

- This sequence initializes the circuit to the state (01), called the reset state.

- Note that not all synchronous sequential circuits have a synchronizing sequence.

- Example : Counters

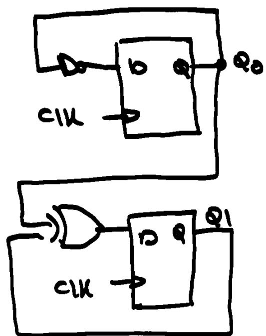




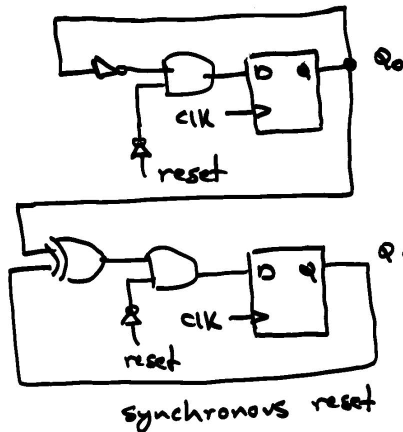
No synchronizing sequence

- Often when a circuit has no synchronizing sequence, an explicit reset line is added.

2-bit counter



2-bit counter with reset



synchronous reset

- To process the unknown initial state, simulation algorithms use a separate logic value denoted by $\underline{\underline{\mu}}$, to indicate <u>unknown logic value</u>.

- The $\underline{\underline{\mu}}$ logic value is processed together with the binary logic values during simulation

- The value $\underline{\underline{\mu}}$ represents one value in the set $\{0,1\}$.

- $AND(0,\mu) = AND(\{0\},\{0,1\}) = \{AND(0,0), AND(0,1)\}$
  $= \{0,0\} = \{0\} = 0$

- $OR(0,\mu) = OR(\{0\},\{0,1\}) = \{OR(0,0), OR(0,1)\}$
  $= \{0,1\} = \mu$

- $NOT(\mu) = NOT(\{0,1\}) = \{NOT(0), NOT(1)\}$
  $= \{1,0\} = \mu$

AND	0	1	μ
0	0	0	0
1	0	1	μ
μ	0	μ	μ

OR	0	1	μ
0	0	1	μ
1	1	1	1
μ	μ	1	μ

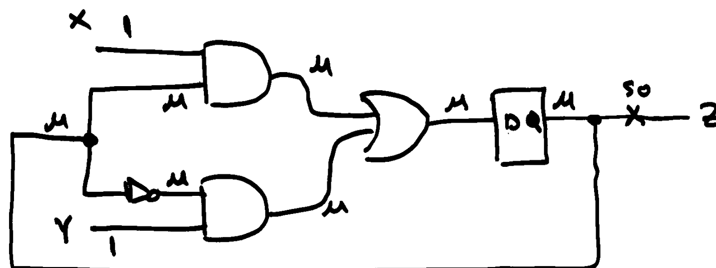
NOT	0	1	μ
	1	0	μ

Truth Tables for 3-valued logic

- There is a loss of information associated with the use of 3-valued logic.

- This loss of information may lead to <u>pessimistic results</u>.

- <u>Example</u>

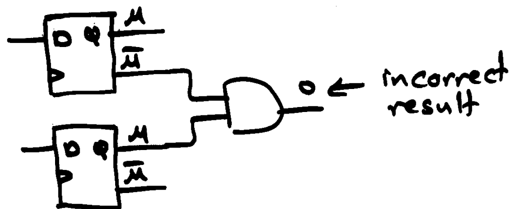


- using 3-valued simulation, the test XY = 11 does not detect the fault Z s-a-0

- However, in reality this test detects the fault since regardless of the initial state, the next state will be 1. Thus, the fault will be excited & detected.

- It may appear that the use of complementary unknown values μ and $\bar{\mu}$ along with the rule $\mu \cdot \bar{\mu} = 0$ and $\mu + \bar{\mu} = 1$ will solve the problem.

- However, this solution leads to incorrect results if we have more than one FF.



- A correct solution is to use several distinct unknown values $\mu_1, \mu_2, ---, \mu_k$ (one for every FF) and the rules $\mu_i \cdot \bar{\mu}_i = 0$ and $\mu_i + \bar{\mu}_i = 1$

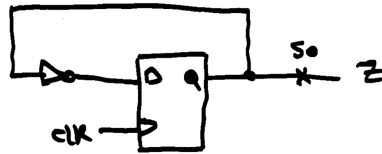
- This solution is not practical for large circuits since the values of some lines would be represented by large Boolean expressions of μ_i variables.

- Structural Automatic Test Pattern Generation (ATPG) algorithms are based on 3-valued logic simulation i.e. they are pessimistic.

Fault Detection in Sequential Circuits

- Testing sequential circuits is considerably more difficult than testing combinational circuits.
- To detect a fault, a test sequence is usually required rather than a single input vector.
- The response of a sequential circuit is a function of its initial state.
- Let T be a test sequence and $R(q,T)$ be the response to T of a sequential circuit N starting with the initial state q.
- Let N_f be the obtained circuit in presence of the fault f. We denote $R_f(q_f,T)$ the response of N_f to T starting in the initial state q_f.
- <u>Definition:</u> A test sequence T <u>strongly</u> <u>detects</u> the fault f iff the output sequences $R(q,T)$ and $R_f(q_f,T)$ are different for every possible pair of initial states q and q_f.
  (Multiple observation time test strategy)

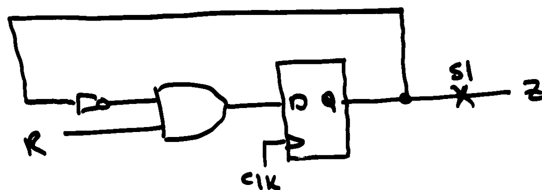
Example



state	fault-free response	faulty response
		0 0
$q_0 = 0$	0 1	0 0
$q_1 = 1$	1 0	

- The fault Z s-a-0 is strongly detected if we apply two clock cycles
- Although the fault Z s-a-0 is strongly detected, the error symptom cannot be simply specified.
- We cannot say that at some point in the output sequence the normal circuit will have a 1 output and the faulty circuit a 0 output or vice versa.
- We must list all possible responses of the normal and faulty machines which is impractical.
- A tester compares the obtained and the expected output sequences on a vector-by-vector basis.

- <u>Definition</u> A test sequence T <u>detects</u> the fault f iff for every possible pair of initial states q and q_f, the output sequences $R(q, T)$ and $R_f(q_f, T)$ are different for some specified vector $t_i \in T$.
(single observation time test strategy)



State	R=0, R=x	fault-free response	faulty response
$q_0 = 0$		0 0	1 1
$q_1 = 1$		1 0	1 1

- The fault Z s1 is detected by the test sequence $R = \{0, x\}$

- Note that detection implies strong detection but not vice versa.

- To determine the vector t_i when an error caused by f can be observed at POs independent of the initial states q and q_f, testing experiment is divided into two phases.

- Initialization phase:
  - We apply an initialization sequence T_I such that at the end of T_I, both N and N_f are brought to known states q_I and q_{If}.
  - The output responses are ignored during T_I since they are not predictable.

- Distinguishing phase:
  - We apply a distinguishing sequence T' that distinguishes between the fault-free state q_I and the faulty state q_{If}.

    i.e. $R(q_I, T') \neq R(q_{If}, T')$ and t_i is taken as the first vector of T' for which the error is observed.

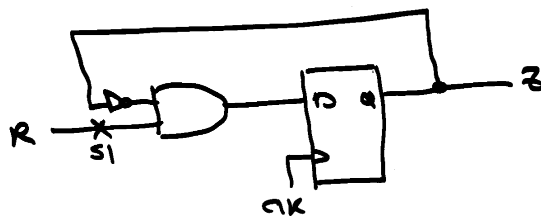
- This type of testing is based on the assumption that an initialization sequence T_I exists.

- Most practical circuit have an initialization sequence, employed to start its operation from a known state.

- An often used technique is to provide a common reset line to every FF.

- An initialization sequence for the fault-free circuit N may fail to initialize some faulty circuit N_f. Such a fault is said to prevent initialization.

- Faults that prevent initialization are considered undetectable by edge-pin testing.



- The fault R s-a-1 prevents initialization
- It is not detectable but strongly detectable.

Sequential Untestability & Redundancy

- A fault is redundant if the behavior of the faulty circuit cannot be distinguished from that of the fault-free circuit.

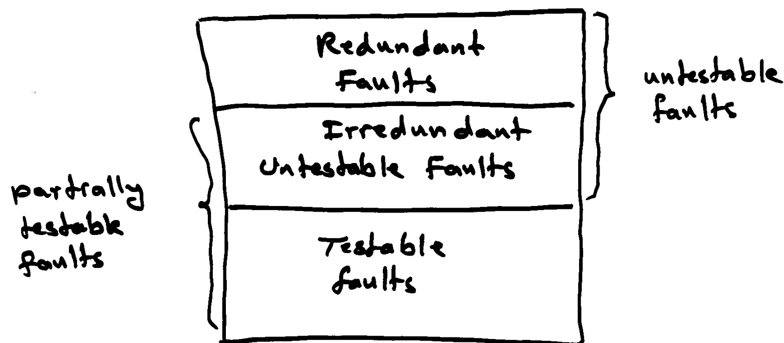
- Note that relying on 3-valued logic simulation to verify this fact may lead to erroneous results.

- Definition:
  A fault f is said to be detectable if there exists an input sequence I such that for every pair of initial states S and S^f of the fault-free and faulty circuit, the response $Z(I,S)$ of the fault-free circuit to I is different from $Z^f(I,S^f)$ of the faulty circuit (at some time on some output).

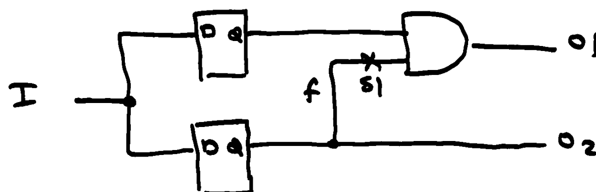
- Definition:
  A fault is untestable if it is not detectable.

- **Definition:**
A fault is <u>partially testable</u> if there exists an initial state S^f of the faulty circuit and an input sequence I such that for every fault-free initial state S, the response $Z(I,S)$ is different from $Z^f(I, S^f)$.

- **Definition**
A fault is <u>redundant</u> if it is not partially testable.



- Example of a partially testable fault:



consider the fault f s-a-1

P. state	I=0	I=1	I=0	I=1
00	00,00	11,00	00,00	11,00
01	00,01	11,01	00,01	11,01
(10)	00,00	11,00	00,10	11,10
11	00,11	11,11	00,11	11,11
	fault free N. state, output		faulty N. state, output	

- The fault f s-a-1 is neither strongly detectable nor detectable. This is because re. state $q = 00$ is equivalent to $q^f = 00$.

- However, the fault is partially testable. The faulty state $q^f = 10$ is distinguishable from all fault-free states.

- Note that the fault f s-a-1 in the previous example is partially testable and irredundant by definition.

- However, intutively the circuit seems redundant.

- <u>Definition</u>:
  Consider the set of states $\{S_c\}$ reachable after powering up the faulty circuit and applying any sequence of c input vectors.
  (Note that $\{S_c\}$ shrinks as c increases).
  Then, a fault is <u>c-cyle redundant</u> if it is not partially testable under the assumption that the initial states of the faulty circuit are restricted to $\{S_c\}$.

- Note that the fault f s-a-1 is 1-cycle redundant.

- A c-cycle redundant fault can be simplified by removing the redundant region associated with that fault.

- Removing a c-cycle redundant fault requires the application of c arbitrary input vectors before the existing initialization sequence.

Fault Equivalence in sequential Circuits

- **Definition:** Two faults f and g are said to be strongly functionally equivalent iff the corresponding sequential circuits N_f and N_g have equivalent state tables.

- This definition is not practical and the practical defintion of fault equivalence is based on the responses of N_f and N_g to a test sequence T.

- We assume that first an initialization sequence T_I is applied that brings N_f and N_g to known states q_{If} and q_{Ig}.

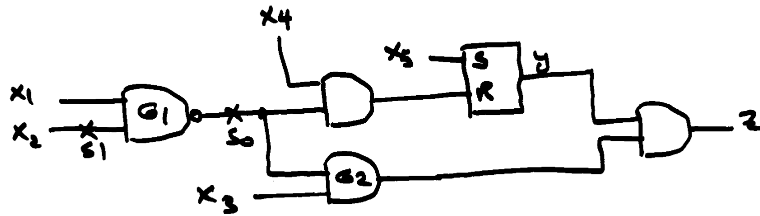
- Let T' be the sequence applied after T_I.

- **Definition:** Two faults f and g are said to be functionally equivalent iff $R_f(q_{If}, T') = R_g(q_{Ig}, T')$ for any T'.

- Note that this definition does not apply to faults preventing initialization

- While equivalence fault-collapsing techniques for combinational circuits remain valid for sequential circuits, dominance fault collapsing techniques are no longer applicable.

<u>Example</u>: consider the test set {10010, 01100} & assume y is initially 1.



Test $x_1 x_2 x_3 x_4 x_5$	fault-free output	x_2 s-a-1 faulty output	G_1 s-a-0 faulty output
10010	0	0	0
01100	0	1	0

So, in a combinational circuit G_1 s-a-0 dominates x_2 s-a-1. However, this in not valid when the combinational circuit is embedded in a sequential circuit.