520.492 Mixed-Signal VLSI Systems

Week 9

Structured and Testable Design

References

1. Geiger, Allen and Strader: Chapters 6 and 7.
2. C. Mead and L. Conway, *Introduction to VLSI Systems*, Chapter 9, Allison-Wesley, 1980.

• STRUCTURED DESIGN

objective : to reduce the global amount of effort involved in the design and layout, and to allow for direct extension and alteration

- HIERARCHY
- REGULARITY
- MODULARITY
- LOCALITY

• TESTABLE DESIGN (Design for testability)

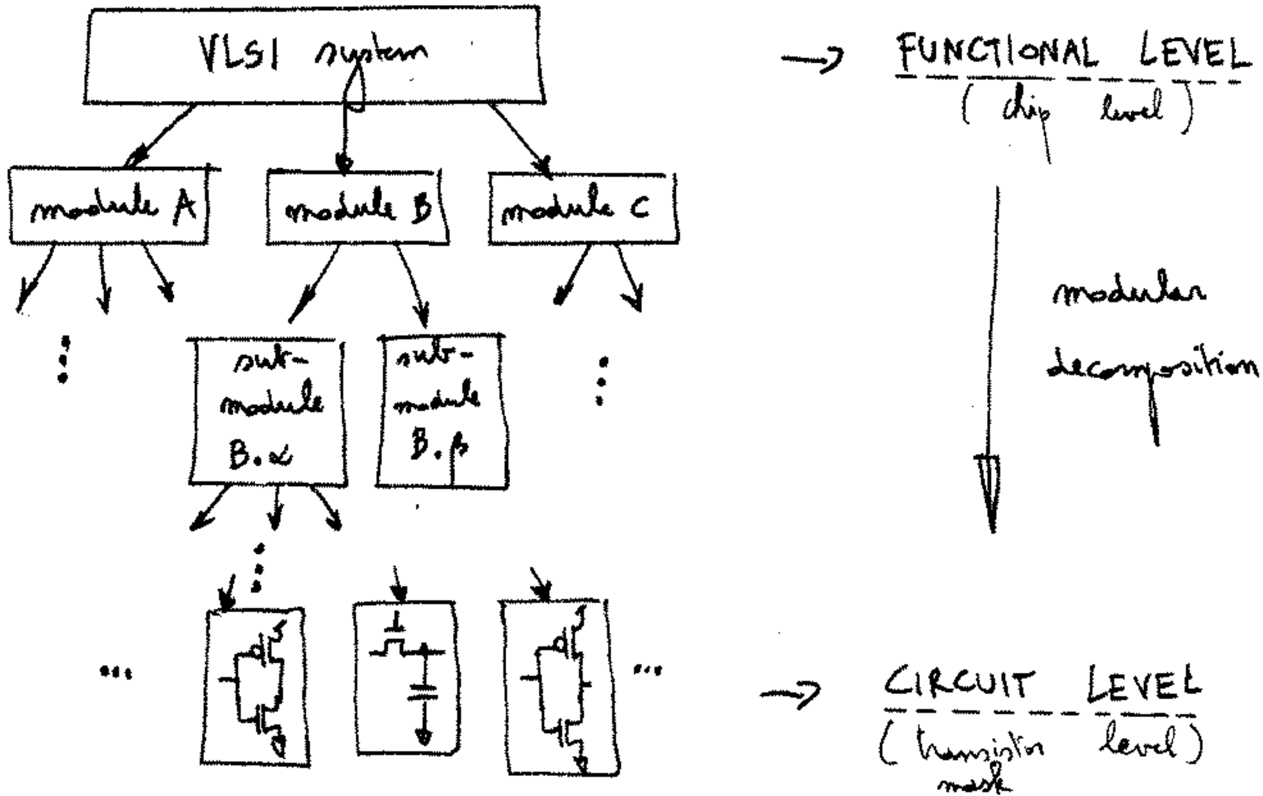
objective : to reduce the burden of exhaustively testing the fabricated VLSI circuit for structural faults and other non-idealities

- OBSERVABILITY
- CONTROLLABILITY

(The above requirements for structured design are useful here as well)

STRUCTURED DESIGN

a) Hierarchy and Regularity

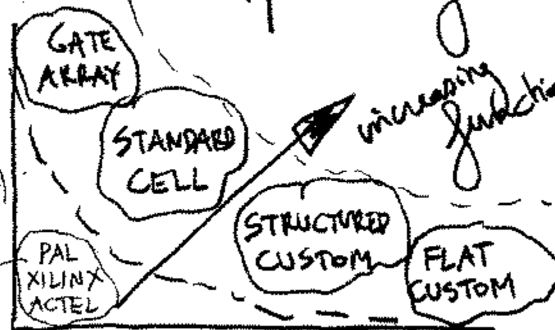


HIERARCHY: tree-based structured decomposition towards higher detail in the description

REGULARITY: parallel instances of identical cells are used throughout for the modules and sub-modules, at different levels in the hierarchy

$$R = \frac{\text{total \# of transistors}}{\text{\# actually designed trans.}} \approx \text{average \# instances/cell}$$

⇒ drastically reduces the "explicit" layout and design work

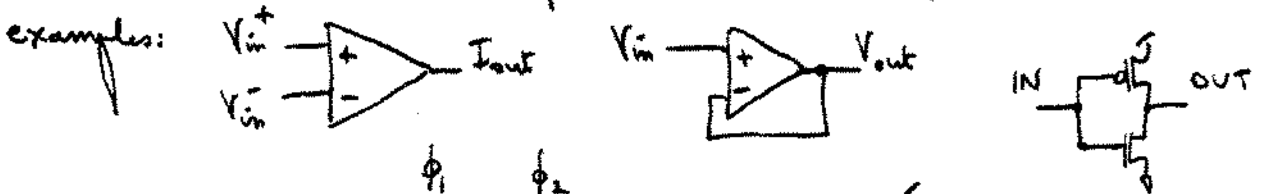


b) Modularity and Locality

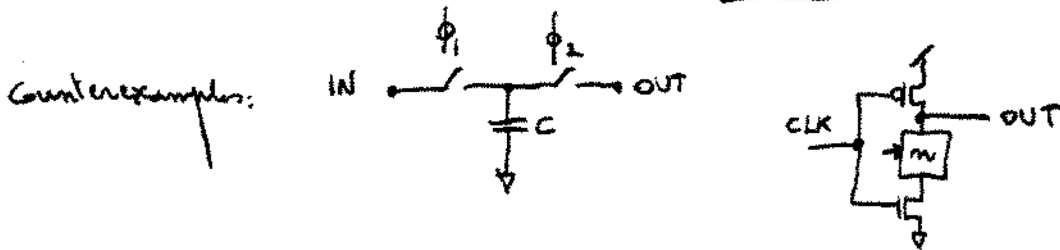
MODULARITY: Each block in the hierarchical decomposition has a well-defined internal functionality and a suitable external interface (self-contained externally)

typically:
- voltages : input : HI – Z output : LO – Z
- currents : input : LO – Z output : HI – Z

examples:



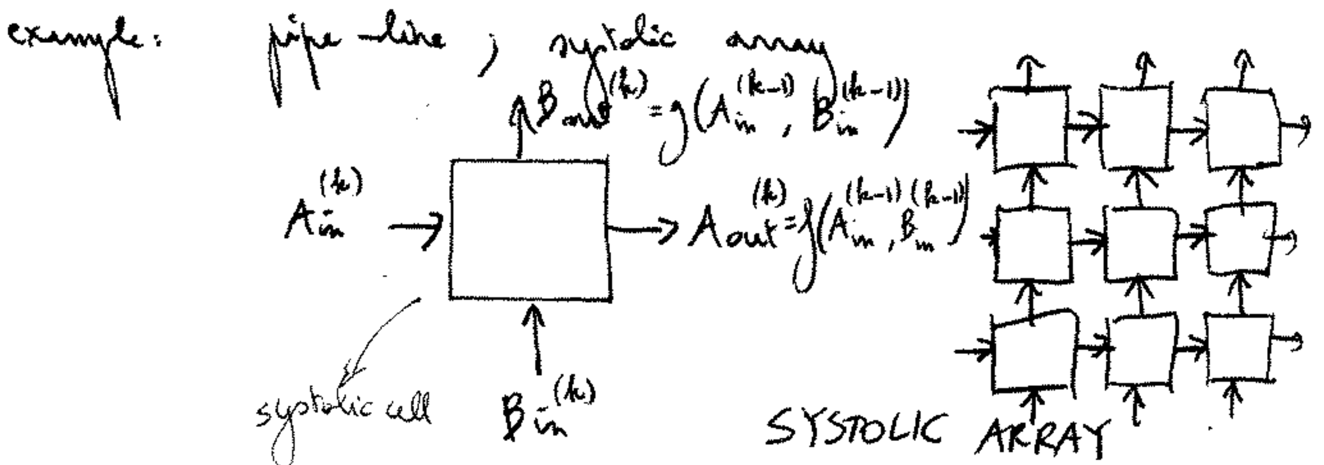
counterexamples:



LOCALITY: Modules (or sub-modules) are constructed as homogeneous "clusters" of internally self-contained functionality in order to reduce interconnections between modules to a minimum level

→ mostly local signals
→ a few global signals for module interconnect and for bias / timing generation and control

example: pipe-line, systolic array

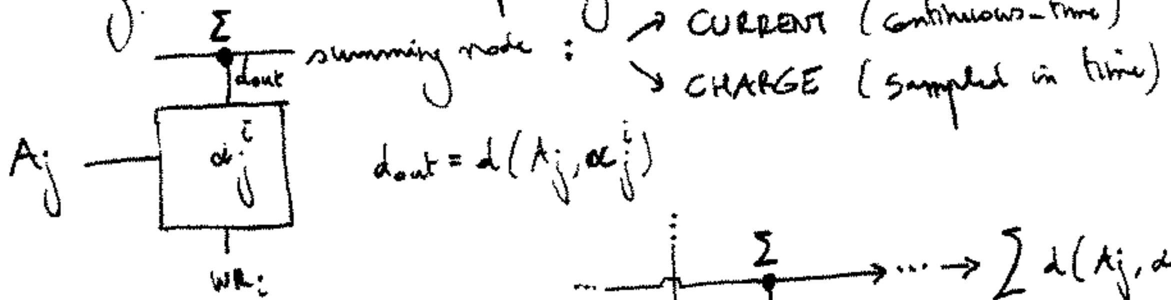


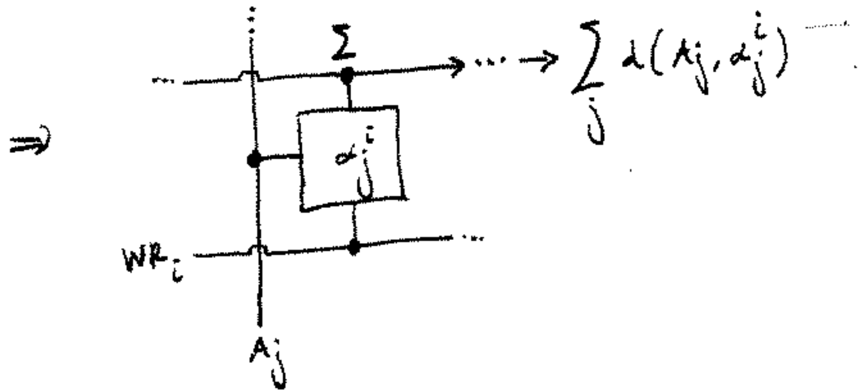
NOTE : restriction on locality does NOT exclude global computation schemes which exploit a DISTRIBUTED REPRESENTATION, nor does the requirement of modularity exclude cases where boundary values are influenced by other values in the same class sharing the global distributed computational function.

EXAMPLES :

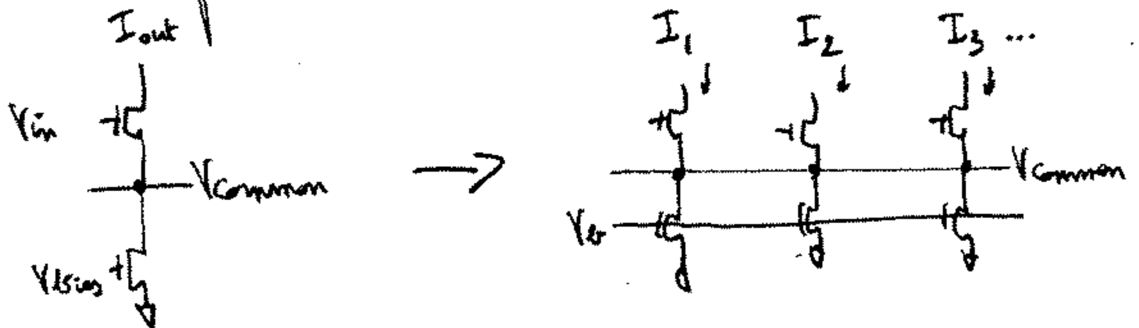
1) vector quantizer:

a) summing node on the output of the distance cells:





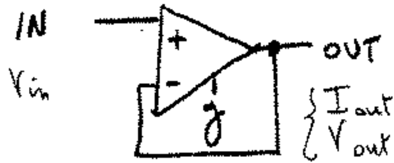
b) common "competition" line in winner-take-all



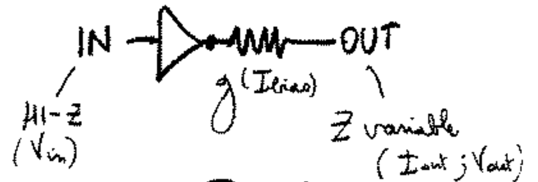
EXAMPLES (continued):

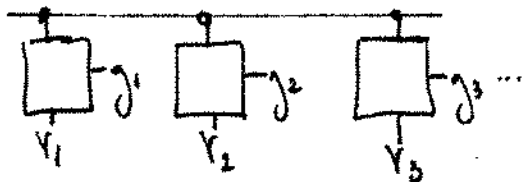
2) follower aggregator (centroid detector, ...):

a) modularity "violation": (cheating)



$$Z_{out} = \frac{1}{g} = \frac{2\frac{kT}{q\kappa}}{I_{bias}}$$

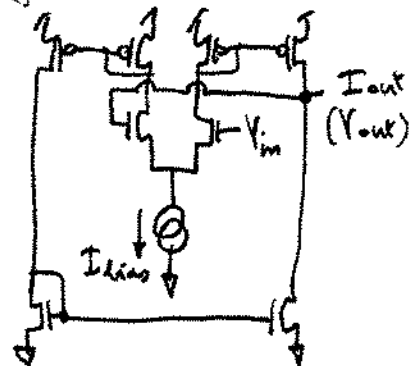




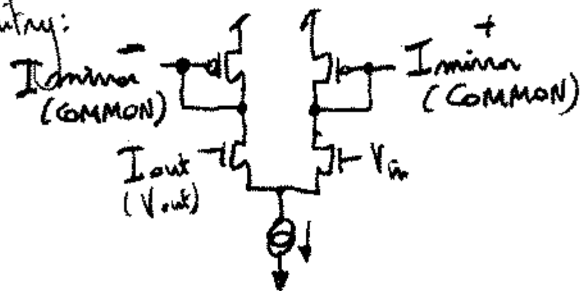
$$V_{out} \cong \frac{\sum g_i \cdot V_i}{\sum g_i} \quad (|V_{out} - V_i| \text{ small})$$

b) possible locality "violation": (cheating)

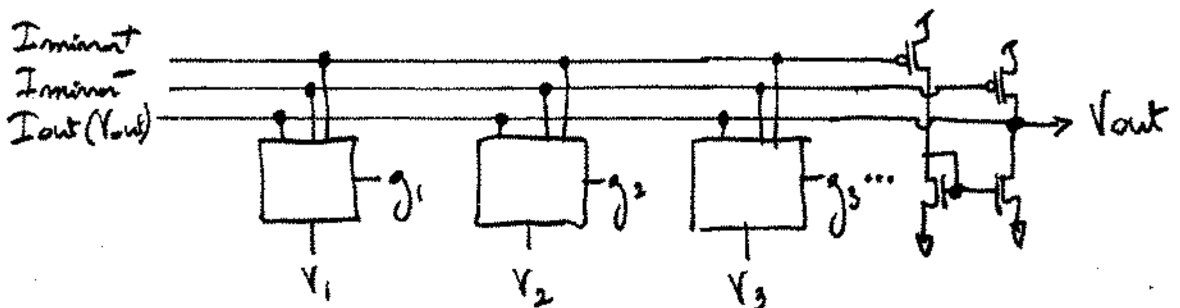
≡



reduction of local circuitry:



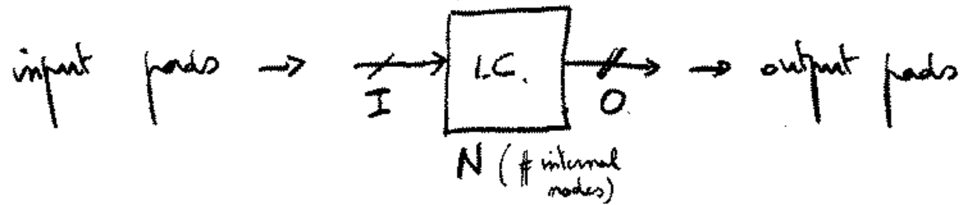
⇓



DESIGN FOR TESTABILITY

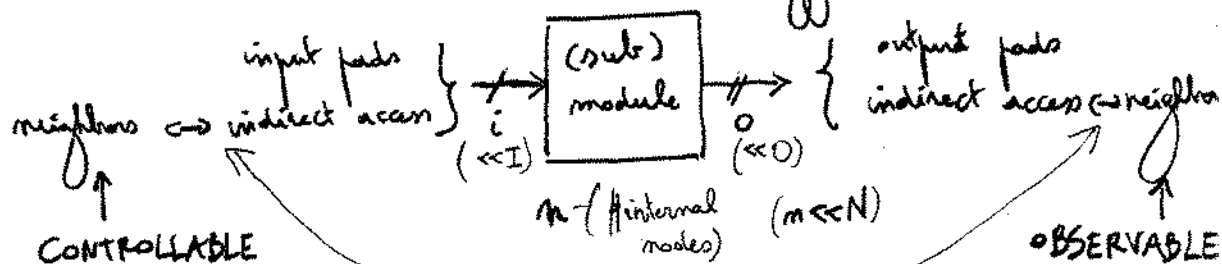
Ideally: make it possible to test all the individual functions in isolated fashion.

Test for:
- Physical defects (SHORTS ; OPENS , more or less)
- Mismatches ; offsets (in analog circuitry)
- Functionality (global ; in principle exhaustive ∀ conditions)



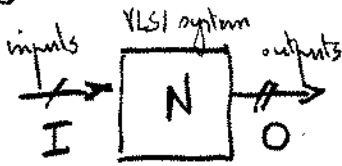
For a complete and efficient check, following requirements will do:

- CONTROLLABILITY : ability to set all (relevant) internal nodes and branches to (more or less) arbitrary values
- OBSERVABILITY : ability to (directly or indirectly) obtain the values of internal nodes & branches
- INDUCTIVE CAUSALITY: ability to decompose the circuit into individually testable parts, each part being accessible individually through induction (assuming under correct operation a given model is in effect)

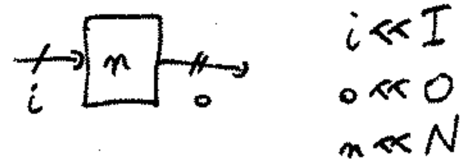


Recursive modular testing assuming induction in causality:

Objective: reduce the combinatorial explosion for testing the flat structure
(n-p completeness)



$\longrightarrow$ decomposition



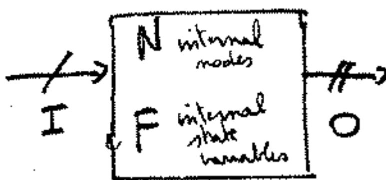
α^I test patterns needed $\longrightarrow$ $\sum_{\text{all (sub)modules}} \alpha^i \ll \alpha^I$

α = # levels of inputs (analog depth)

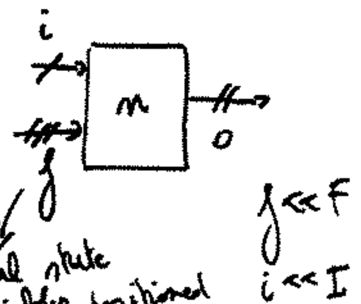
$\alpha \equiv 2$ for binary inputs

Equivalence is only guaranteed if the decomposition satisfies MODULARITY and if the model is correctly implemented (except for the faults under investigation)

Extension to SEQUENTIAL (memory) systems rather than COMBINATORIAL (instantaneous):



$\longrightarrow$ decomposition (along internal state variables also)

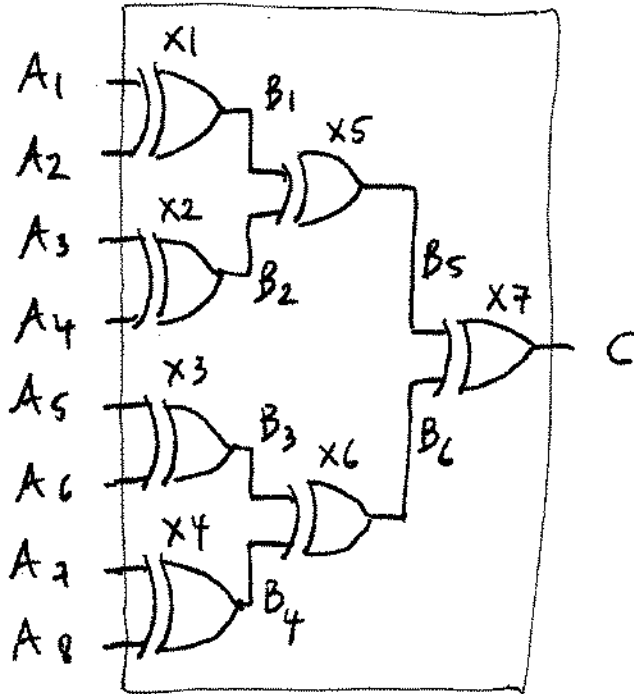


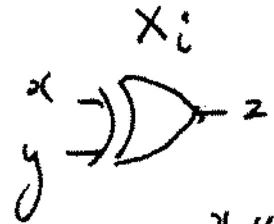
local state variables positioned at boundaries of modules

α^{I+F} patterns needed $\longrightarrow$ $\sum_{\text{all (sub)modules}} \alpha^{i+f} \ll \alpha^{I+F}$

$\Rightarrow$ TESTABILITY also requires { MODULARITY, LOCALITY }

example : PARITY TREE (8-bit, say)





x	y	z
0	0	0
0	1	1
1	0	1
1	1	0

4 test vectors

controllability:

to make $B_5 = 1$: $B_1 \oplus B_2 = 1$ $\Rightarrow$ (A_1, A_2, A_3, A_4) ODD PARITY
$\quad\quad\quad\;\; 0 \qquad\qquad\qquad\; 0 \qquad\qquad\qquad\qquad\qquad\quad$ EVEN
etc..

observability:

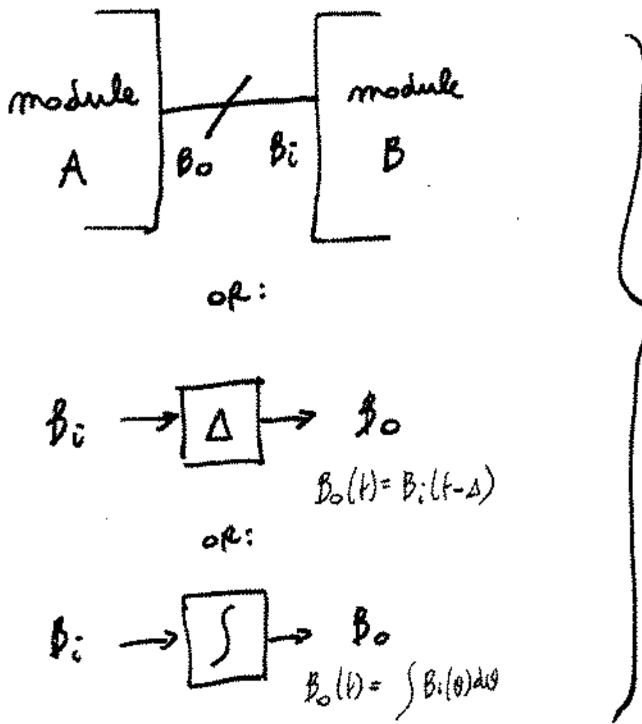
to observe B_2 : set B_1 and B_6 (to whatever) $\Rightarrow B_2 = B_1 \oplus B_6 \oplus C$
etc..

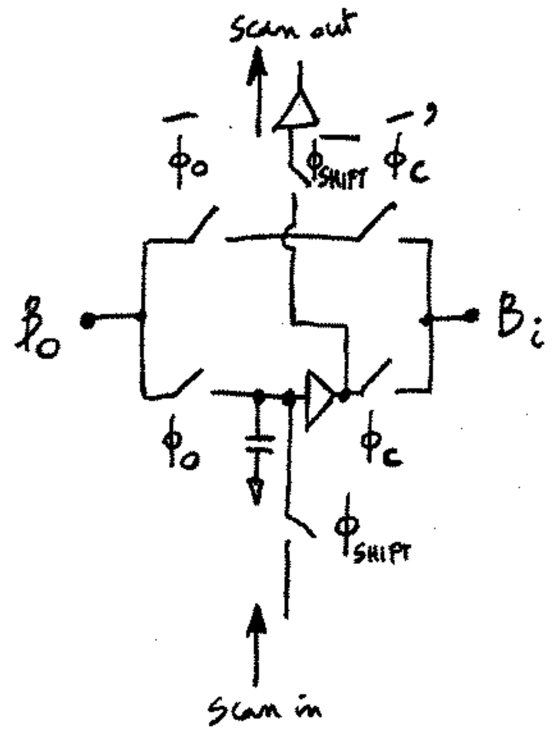
inductive causality:

7×4 test vectors for all individual X_i $i = 1$ through 7

28 test cycles rather than 256 (2^8) for exhaustive test

BOUNDARY SCAN REGISTERS FOR TESTABILITY IN GENERAL ARCHITECTURES



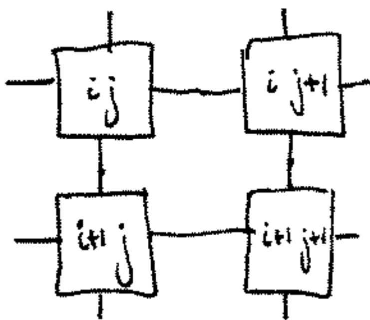


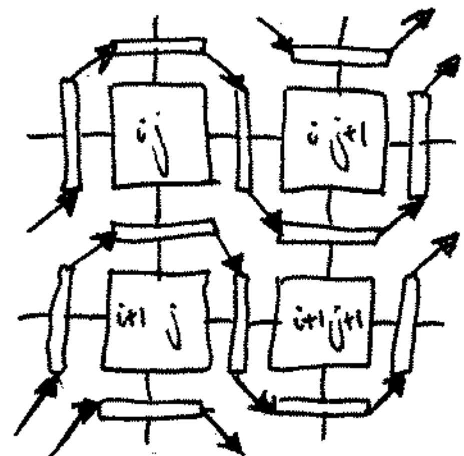
B_i ; B_o : BOUNDARY REGISTER TRANSFER NODES FOR

- interconnect
- internal state variables

- pulse on ϕ_c (*): CONTROL B_i
- pulse on ϕ_o: OBSERVE B_o
- ϕ_{SHIFT} and $\overline{\phi}_{SHIFT}$: scan in B_i's and scan out B_o's

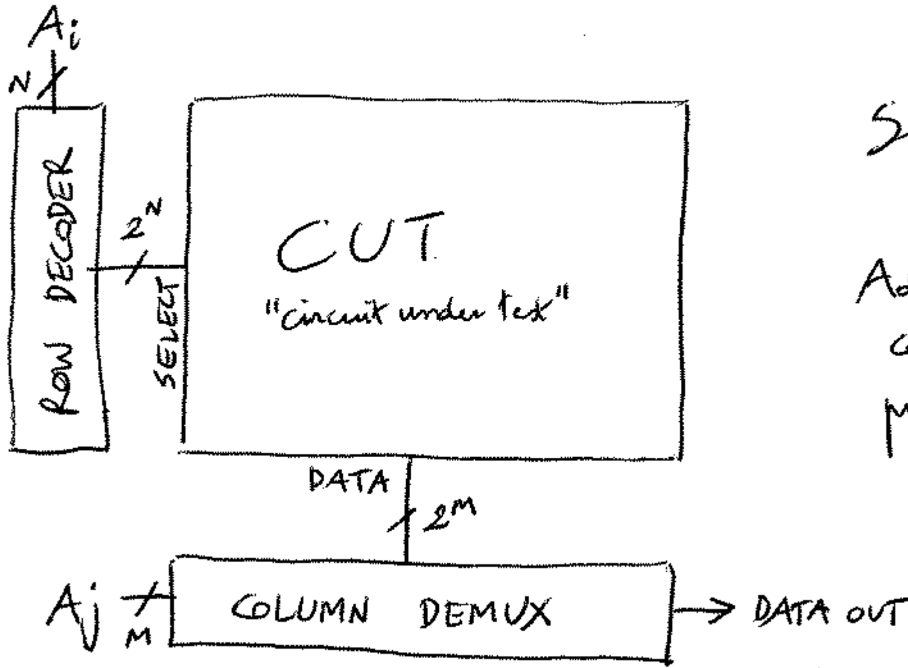
example :





(*) and then retain $\bar{\phi}_c$ LOW

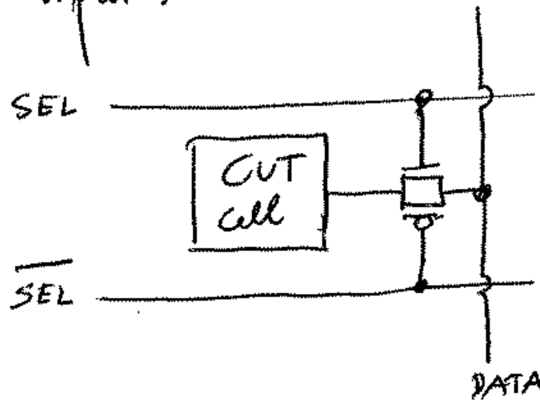
RANDOM-ACCESS TESTABILITY



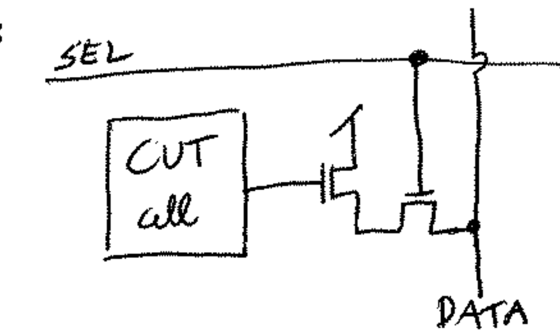
See Week 3 - memory architecture

Address decoders / demux can be used for other purposes! (e.g., I/O multiplexing, programming etc.)

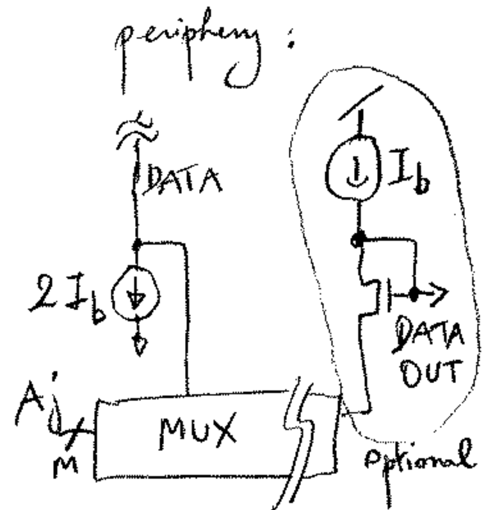
1) current / voltage input:



2) voltage buffer:



periphery:



3) current replica: