1.Introduction

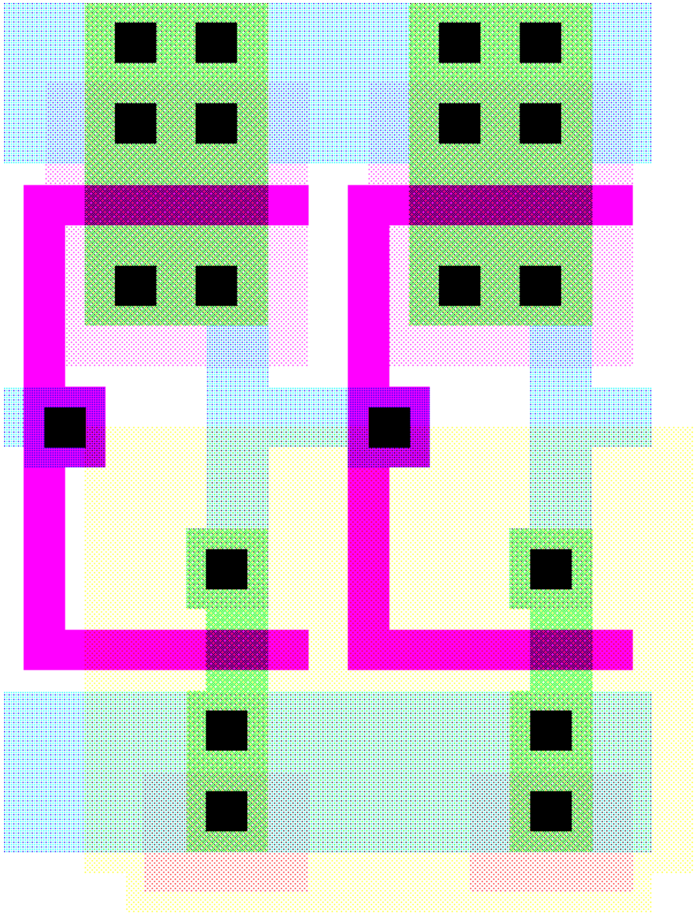
If the automobile had followed the same development cycle as the computer, a Rolls-Royce would today cost $100, get one million miles to the gallon and explode once a year

Most of slides come from Semiconductor Manufacturing Technology
by Michael Quirk and Julian Serda

outline

1. *Course Introduction*
2. *a brief history of IC*
3. *DIC characteristics/Design partitioning*
4. *Semiconductor processing*
5. *Layout and Testing*

How to estimate the area?



Gate Layout

- Layout can be very time consuming
  - Design gates to fit together nicely
  - Build a library of standard cells
- Standard cell design methodology
  - V_{DD} and GND should abut (standard height)
  - Adjacent gates should satisfy design rules
  - nMOS at bottom and pMOS at top
  - All gates include well and substrate contacts

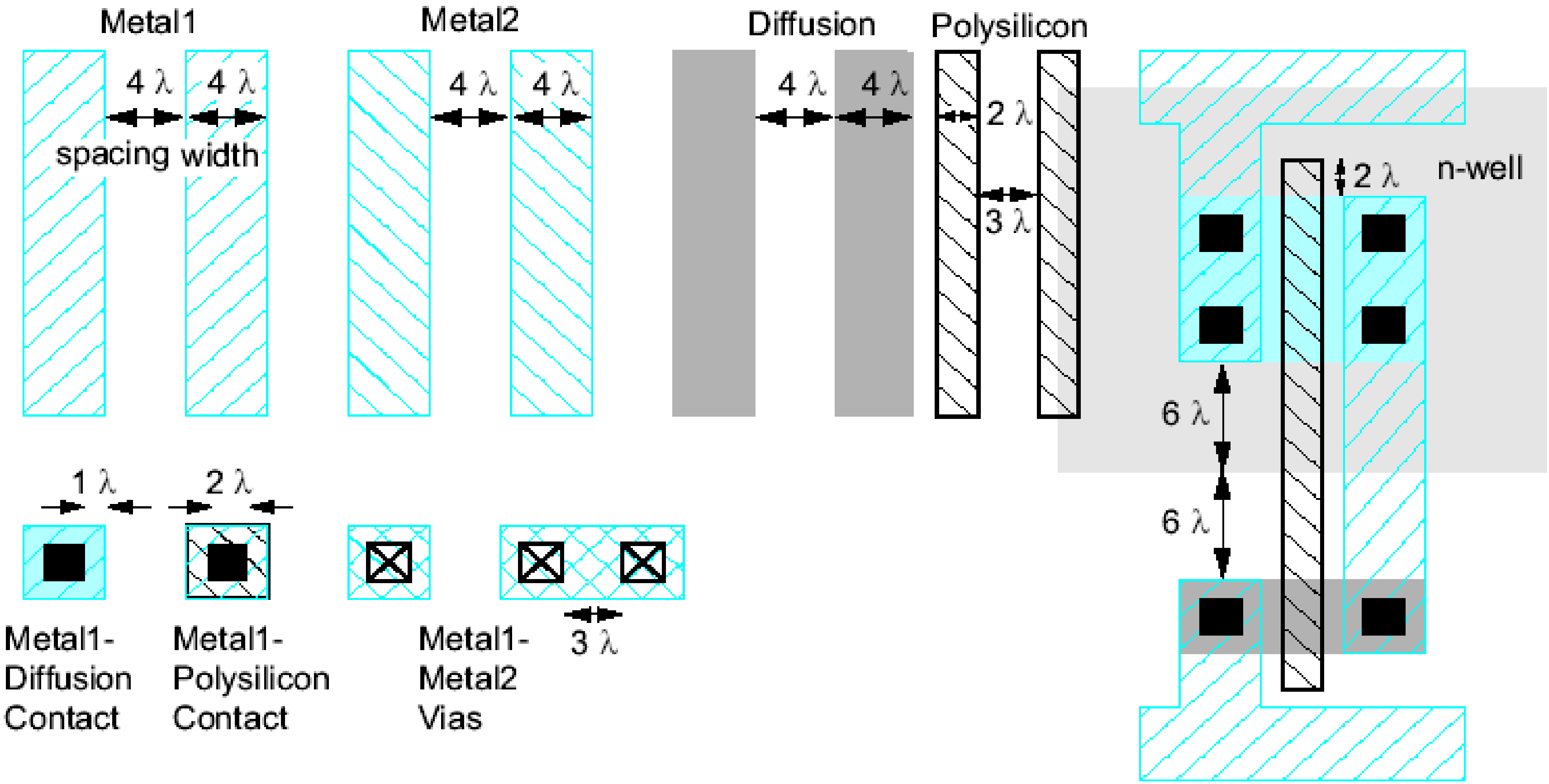
Layout

- Chips are specified with set of masks
- Mini. dimensions of masks determine transistor size
- Feature size f = distance between source and drain
  - *Set by minimum width of polysilicon*
  - *CD- Critical Dimension*
- ***Feature size improves 30% every 3 years or so***
- Normalize feature size for Design Rules
  - Mead Conway *VLSI introduction* 1979
  - Express rules in terms of **λ**= f/2
    - E.g. **λ**= 30nm in 60 nm process

Simplified Design Rules

- Conservative rules to get you started

制程：指的就是polysilicon的宽

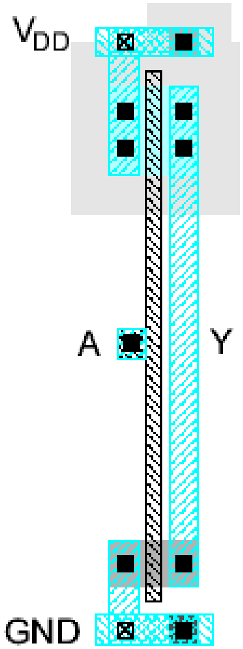


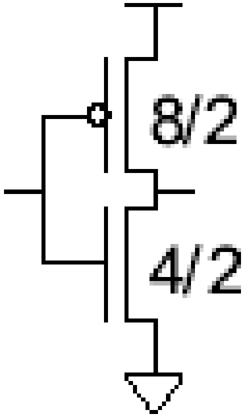
Some Annotations

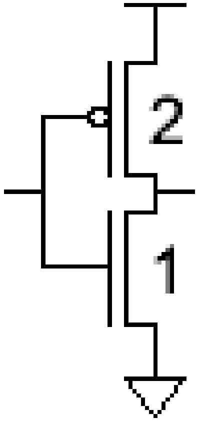
- Metal/diffusion have min. width 4λ
- Contacts are 2λ*2λ and be surrounded by 1λ
- *Polysilicon uses a width of 2λ*
- Polysilicon overlaps diffusion by 2λ and has a spacing of 2λ away
- Polysilicon and contacts have a spacing of 3λ from other poly or contacts
- N-well surrounds pMOS transistors by 6λ and avoids nMOS transistors by 6λ
- CONNECT is[metal-diffusion or metal-poly]
- VIA is metal-metal

Inverter Layout

- Transistor dimensions specified as Width / Length
- ***Minimum size*** is 4λ/ 2λ, sometimes called 1 unit
- In f=60nm process, this is 120nm wide, 60nm long

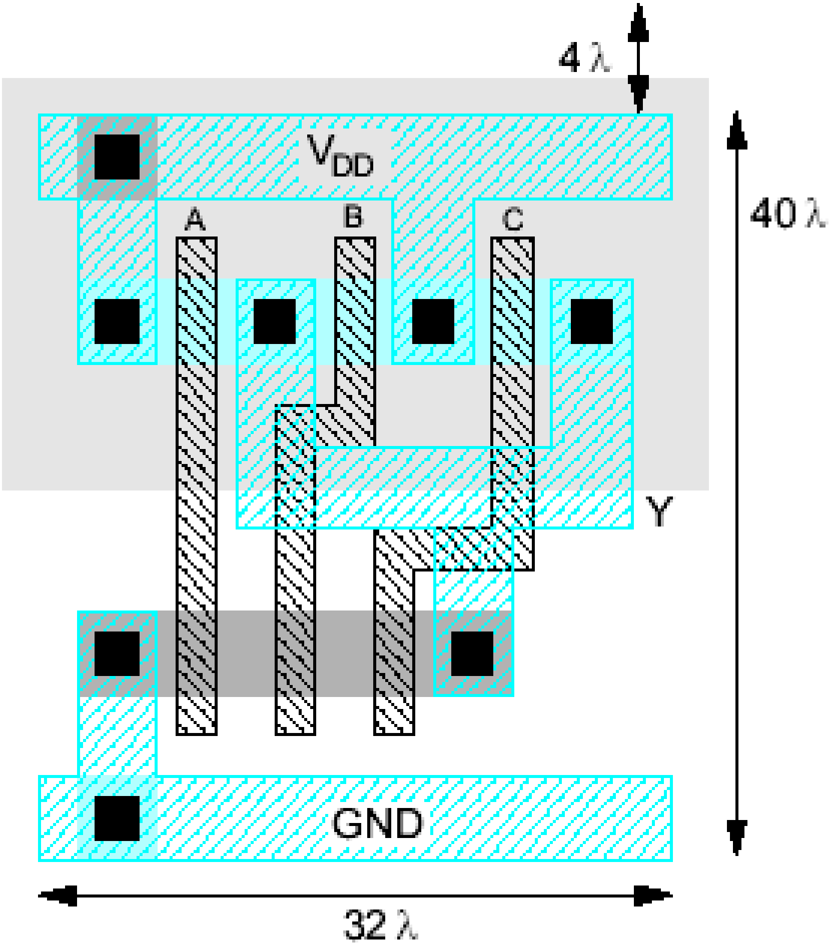






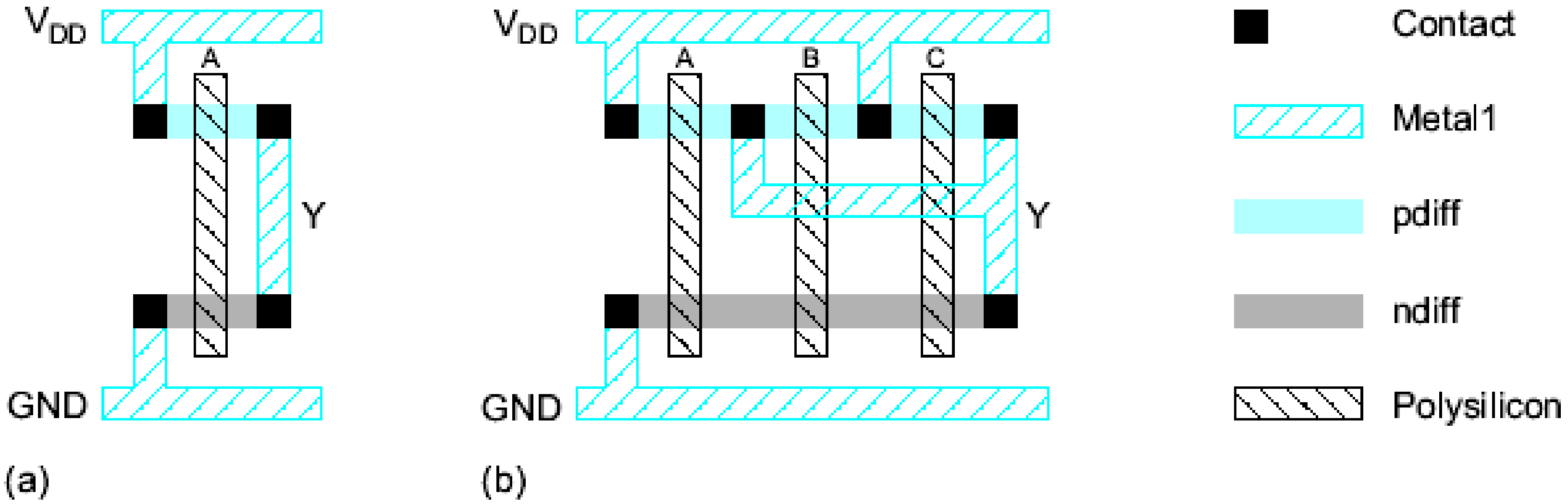
Example: NAND3

- Horizontal N-diffusion and P-diffusion strips
- Vertical polysilicon gates
- Metal1 V_{DD} rail at top
- Metal1 GND rail at bottom
- 32λ by 40λ



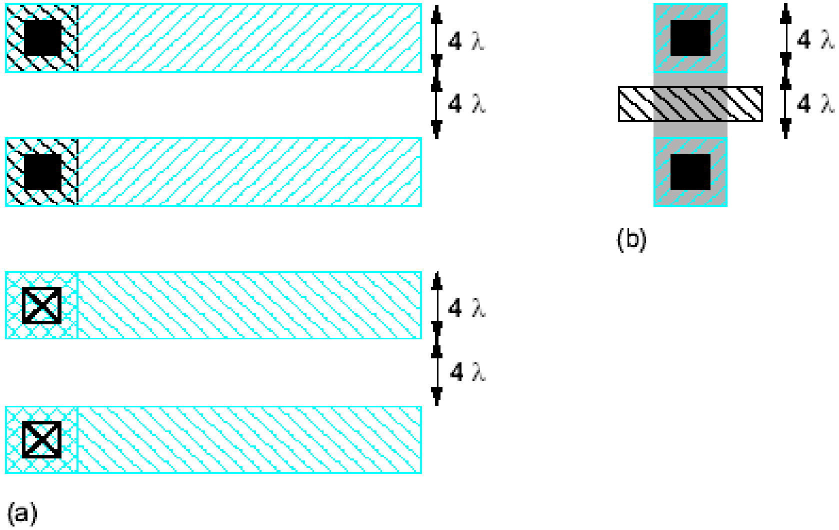
Stick Diagrams

- Stick diagrams help plan layout quickly
  - Need not be to scale
  - Draw with color pencils or dry-erase markers



Wiring Tracks

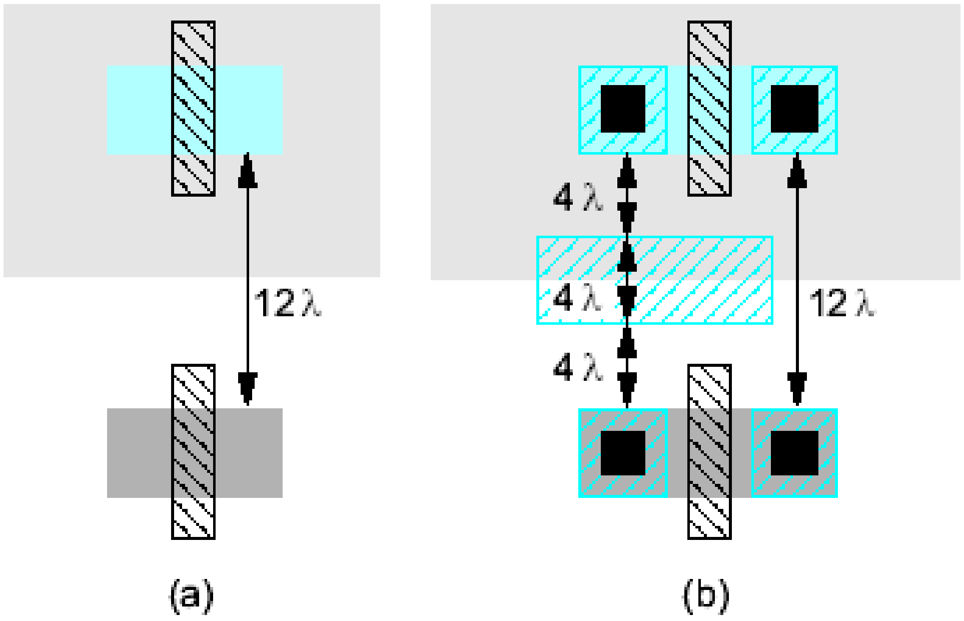
- A wiring track is the space required for a wire
    - 4λ width, 4λ spacing from neighbor = 8λ pitch
- Transistors also consume one wiring track



A rule of thumb: it is reasonable to estimate the height and width of a cell by counting the number of metal tracks and multiplying by **8λ**

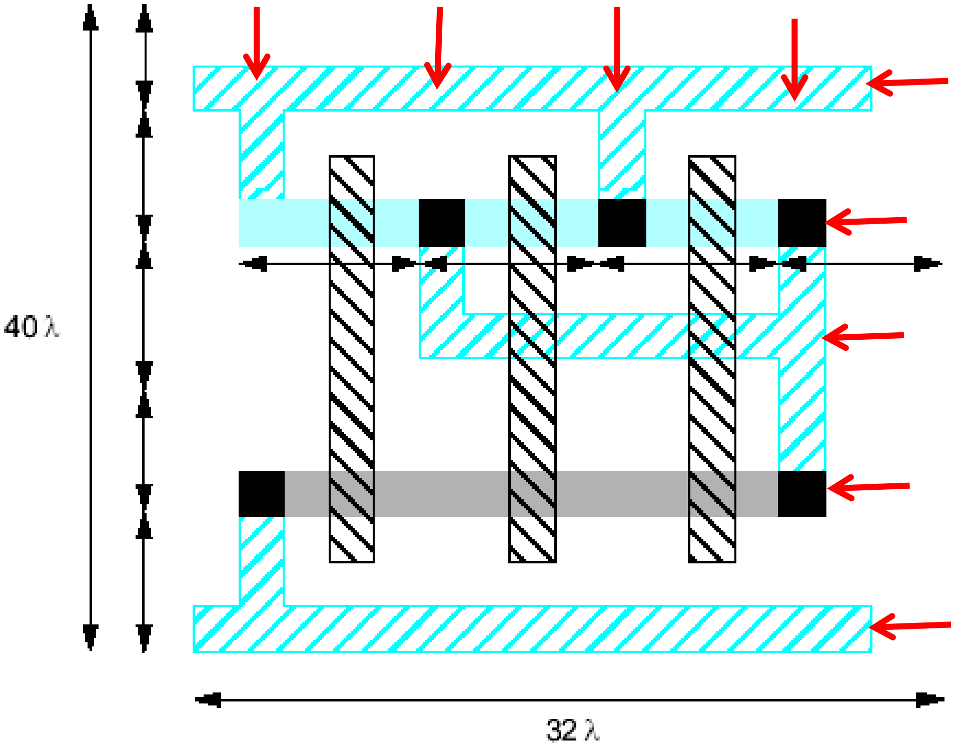
Well spacing

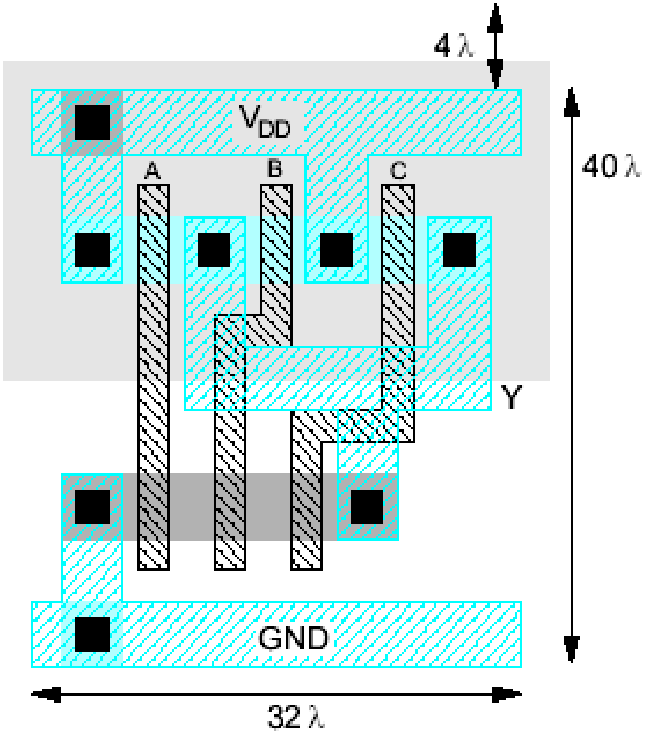
- Wells must surround transistors by 6λ
    - Implies 12λ between opposite transistor flavors
    - Leaves room for one wire track



Area Estimation

- Estimate area by counting wiring tracks
  - Multiply by 8 to express in λ

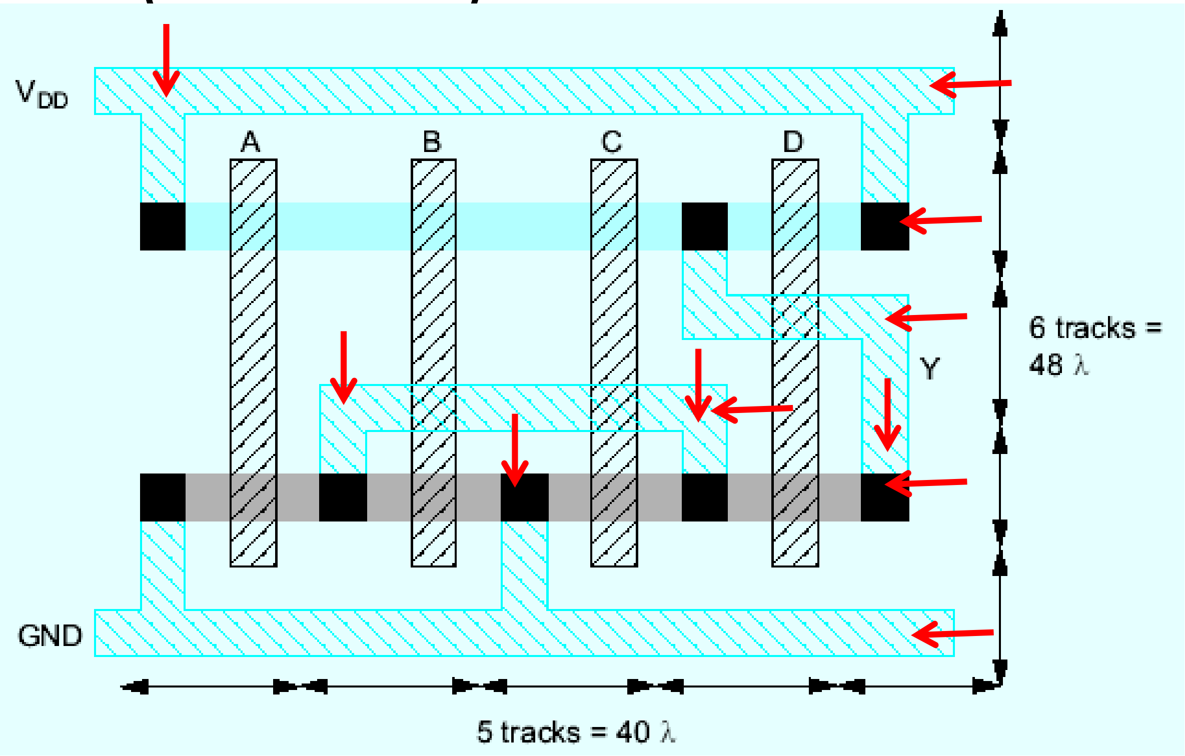




Example: O3AI

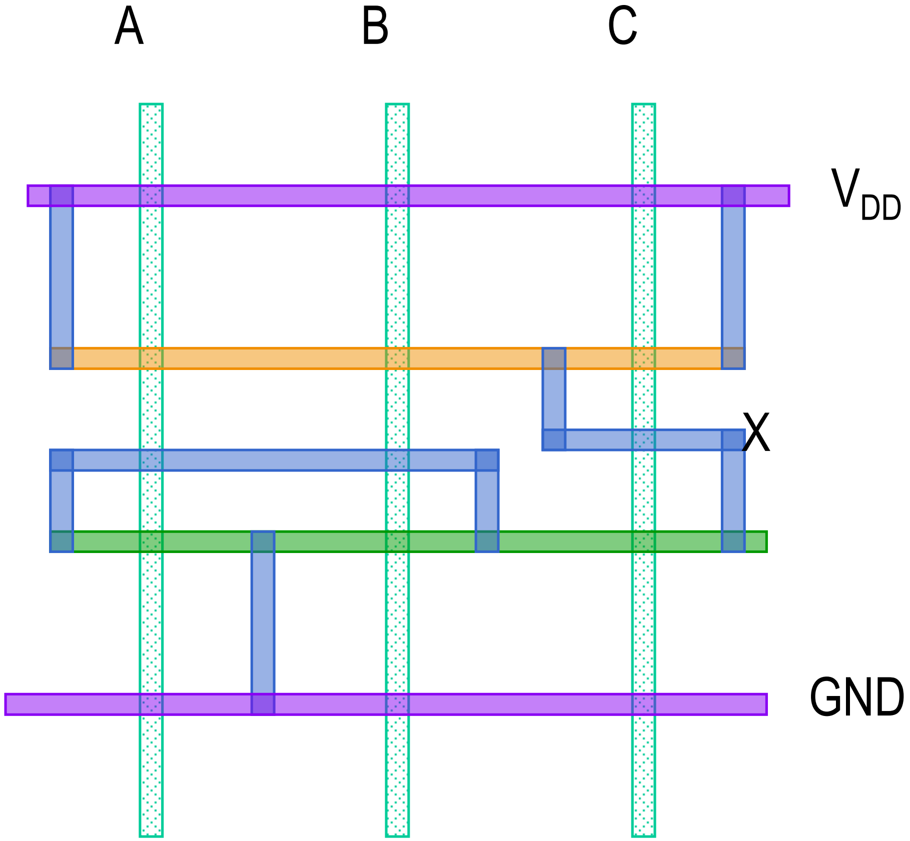
- Sketch a stick diagram for O3AI and estimate area

$$Y = \overline{(A+B+C)\cdot D}$$



tricky for horizontal size

- Diffusion should be counted as a wire
- If no wire between pMOS and nMOS,additional track of wire still should be counted
- A track should be counted between V_{DD} and the next metal line
- If the transistors are wider than 4λ, the extra width should be factored into the area estimate


A
B
C
V_{DD}
X
GND

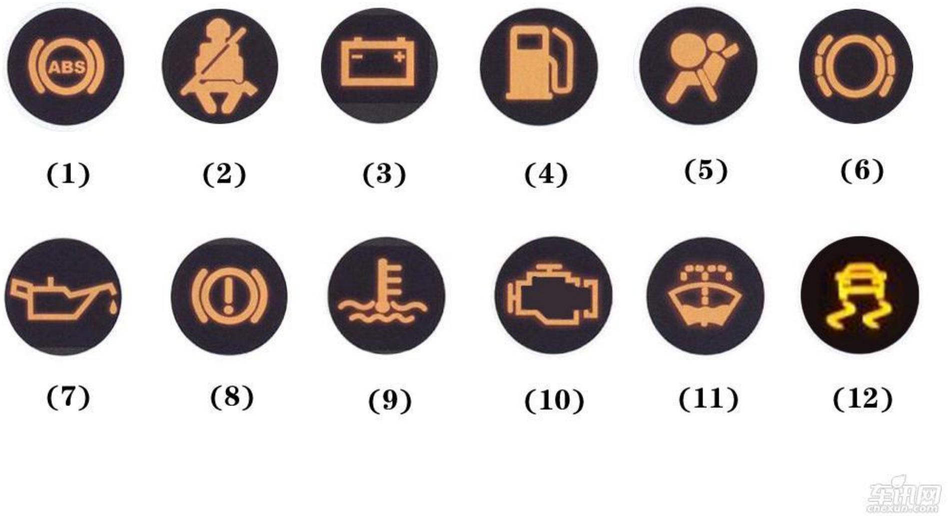
Summary

- MOS Transistors are stack of gate, oxide, silicon
  Can be viewed as electrically controlled switches
- Build logic gates out of switches
- Draw masks to specify layout of transistors
- Using different packaging&assembing tech.
- to start designing schematics and layout for a simple chip!

Processing Testing

Semiconductor processing

- Semiconductor fabrication
- Layout fundamental
- ***Semiconductor testing***
- Semiconductor assembling

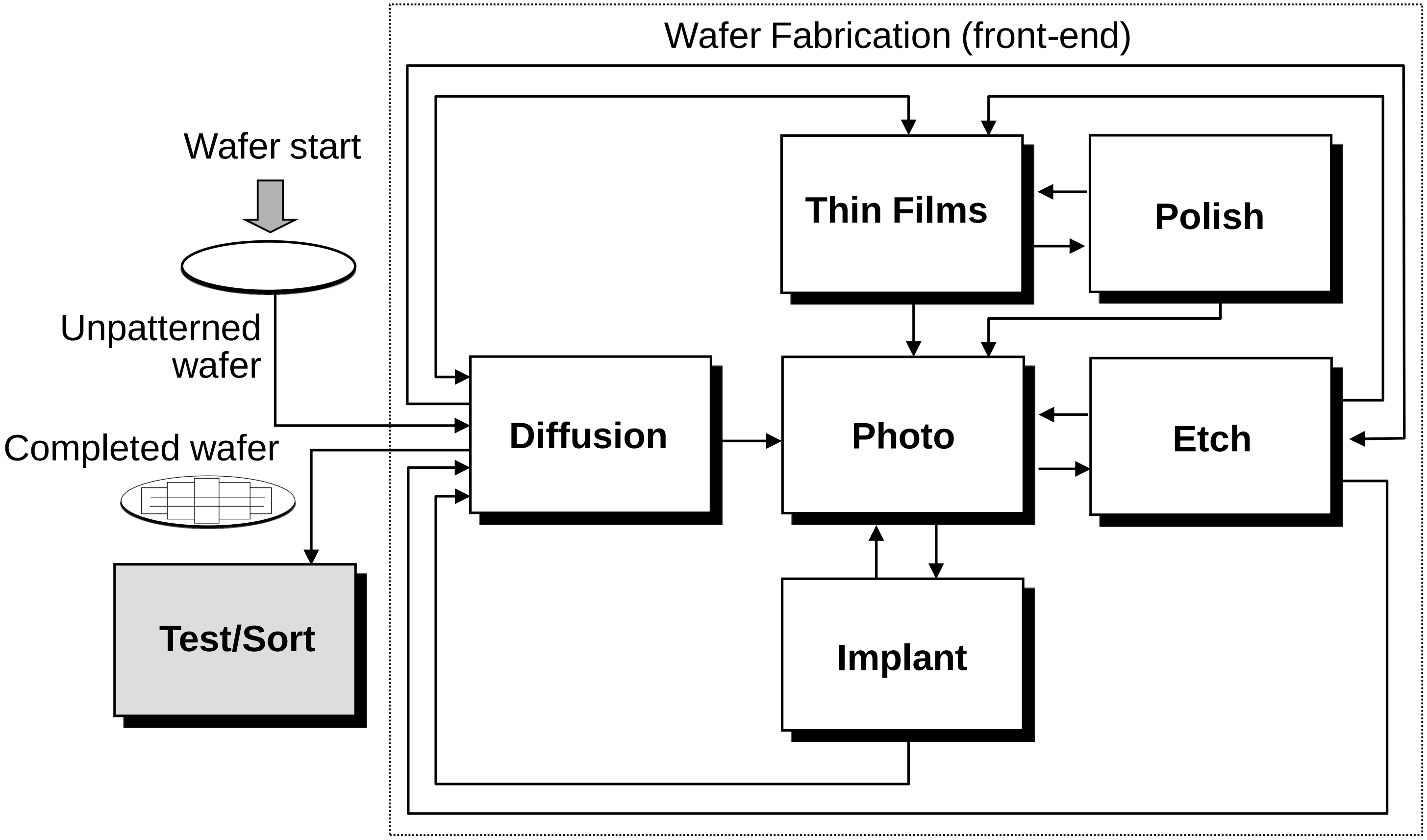


Different Electrical Tests for IC Production (From Design Stage to Packaged IC)

Test	Stage of IC Manufacture	Wafer- or Chip-Level	Test Description
IC Design Verification	Pre-Production	Wafer level	Characterize, debug and verify new chip design to insure it meets specifications.
In-Line Parametric Test	Wafer fabrication	Wafer level	Production process verification test performed early in the fabrication cycle (near front-end of line) to monitor process.
Wafer Sort (Probe)	Wafer fabrication	Wafer level	Product functional test to verify each die meets product specifications.
Burn-In Reliability	Packaged IC	Packaged chip level	ICs powered up and tested at elevated temperature to stress product to detect early failures (in some cases, reliability testing is also done at the wafer level during in-line parametric testing).
Final Test	Packaged IC	Packaged chip level	Product functionality test using product specifications.

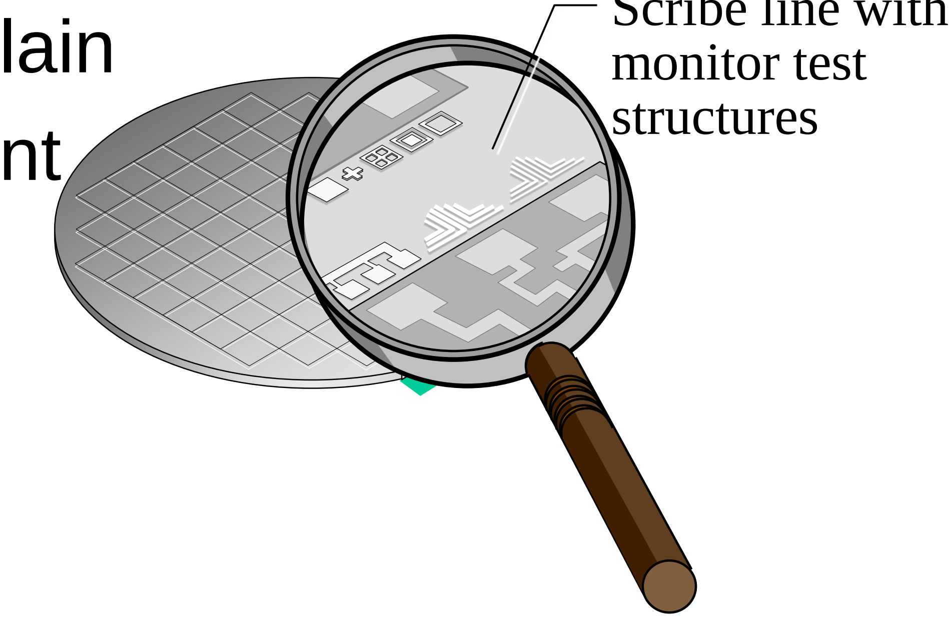
Automated Electrical Tester

Wafer Fab Process Flow with Test



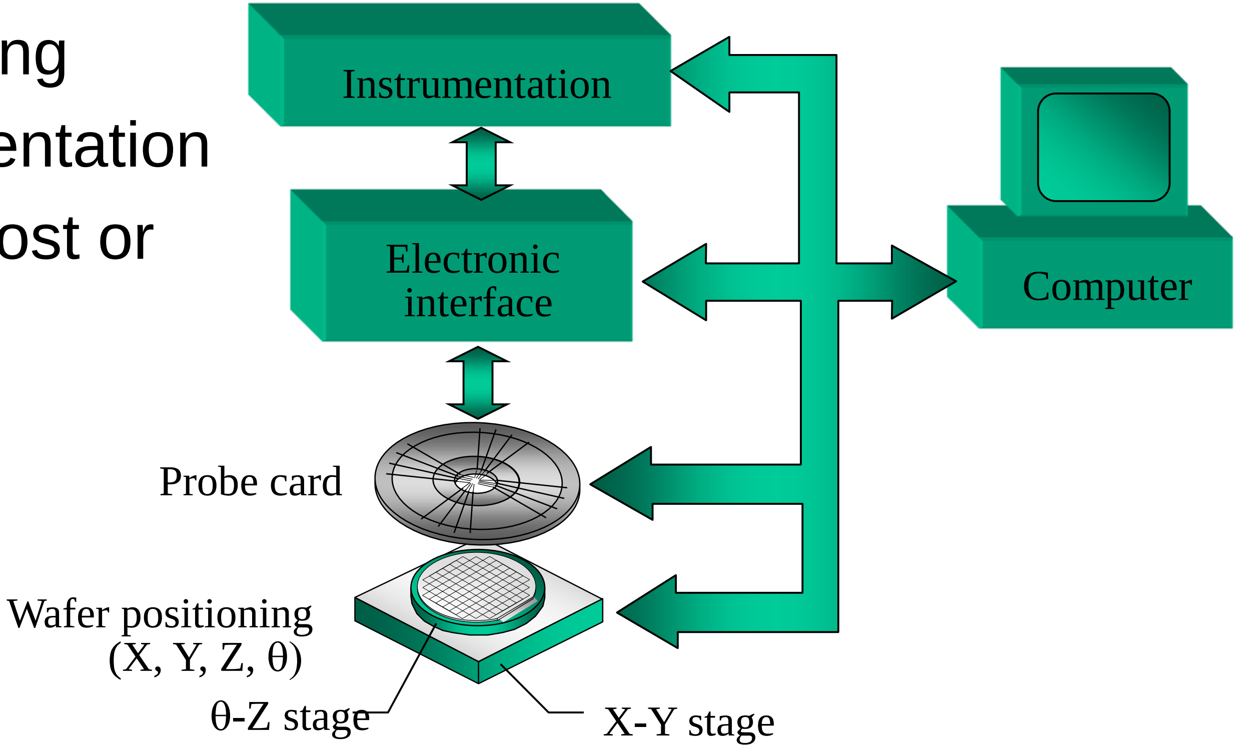
Wafer Test

- In-line Parametric Test (a.k.a. wafer electrical test, WET)
  - In-line test structure
  - In-line test type
  - In-line test data explain
  - In-line test equipment



In-line Parametric Test Systems

- Probe card interface
- Wafer positioning
- Tester instrumentation
- Computer as host or server/network



Examples of Test Structure

Test Structure	Fault Measurement
Discrete transistors	Leakage current, breakdown voltage, threshold voltage and effective channel length
Various line widths	Critical dimensions
Box in a box	Critical dimensions and overlay registration
Serpentine structure over oxide steps	Continuity and bridging
Resistivity structure	Film thickness
Capacitor array structure	Insulator materials and oxide integrity
Contact or via string	Contact resistance and connections

Data Trends

- The same die location keeps failing a parameter on a wafer.
- The same parameter is consistently failing on different wafers.
- There is excessive variation (e.g., > 10%) in measurement data from wafer to wafer.
- Lot-to-lot failure for the same parameter, indicating a major process problem.

Wafer Sort

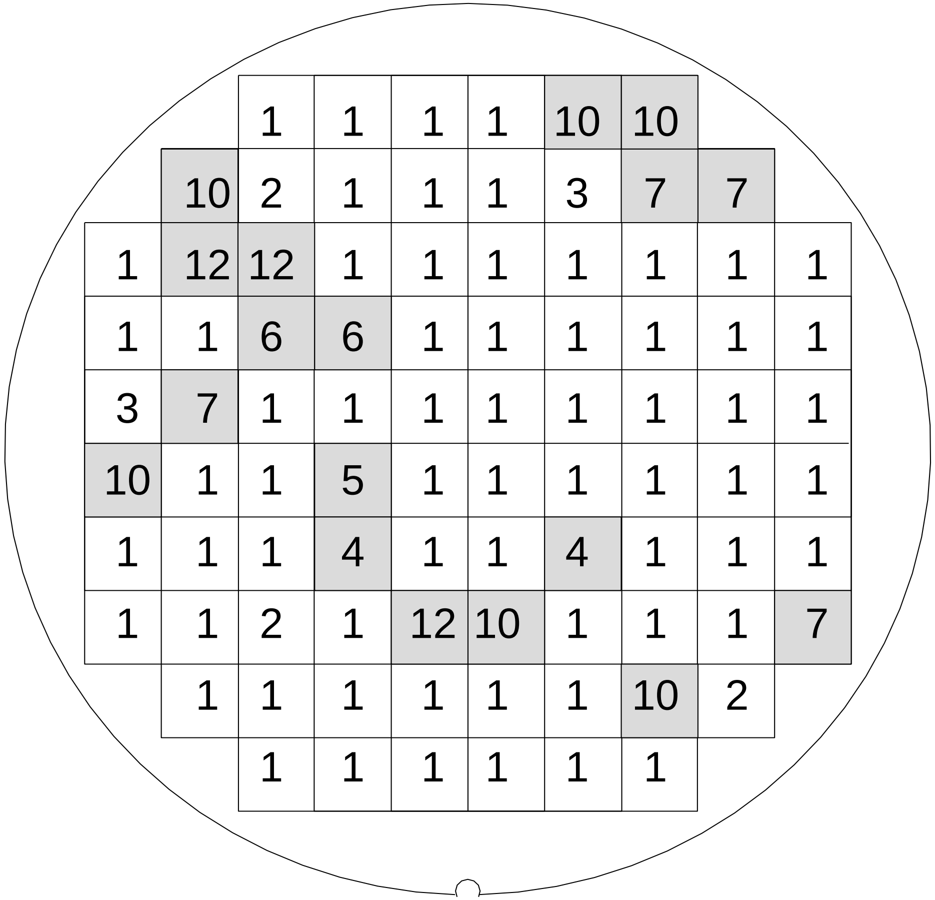
- Wafer Sort (a.k.a. wafer probe)
  - DC testing
  - Output checking
  - Function testing
- The Objectives of Wafer Sort
  - Chip functionality: verify the operation of all chip functions to insure only good chips are sent to the next IC manufacturing stage of assembly and packaging.
  - Chip sorting: sort good chips based on their operating speed performance (this is done by testing at several voltages and varying timing conditions).
  - Fab yield response: Provide important fab yield information to assess and improve the performance of the overall fabrication process.
  - Test coverage: Achieve high test coverage of the internal device nodes at the lowest cost.

Wafer Bin Map with Bin Failures

Device:	Example
Lot:	Example
Wafer:	200 mm
Layer:	Hardware Bins
Yield:	79.54%
Good:	70
Total:	88

🟩 Good

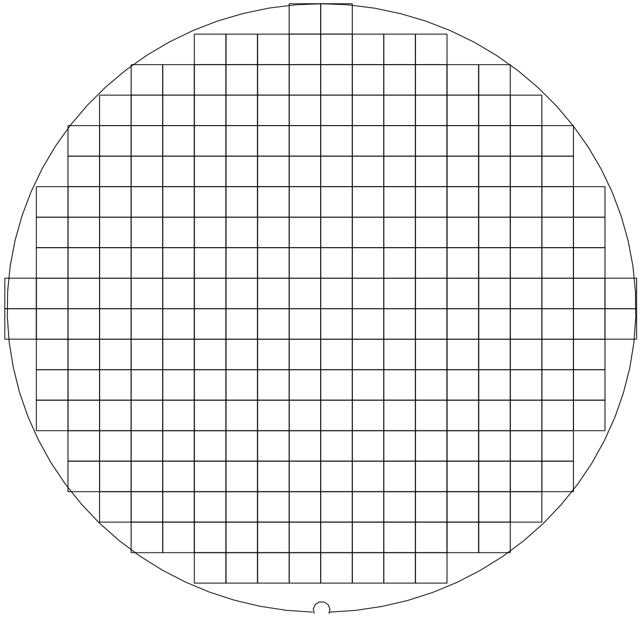
⬜ Bad



		1	1	1	1	10	10		
	10	2	1	1	1	3	7	7	
1	12	12	1	1	1	1	1	1	1
1	1	6	6	1	1	1	1	1	1
3	7	1	1	1	1	1	1	1	1
10	1	1	5	1	1	1	1	1	1
1	1	1	4	1	1	4	1	1	1
1	1	2	1	12	10	1	1	1	7
	1	1	1	1	1	1	10	2	
		1	1	1	1	1	1		

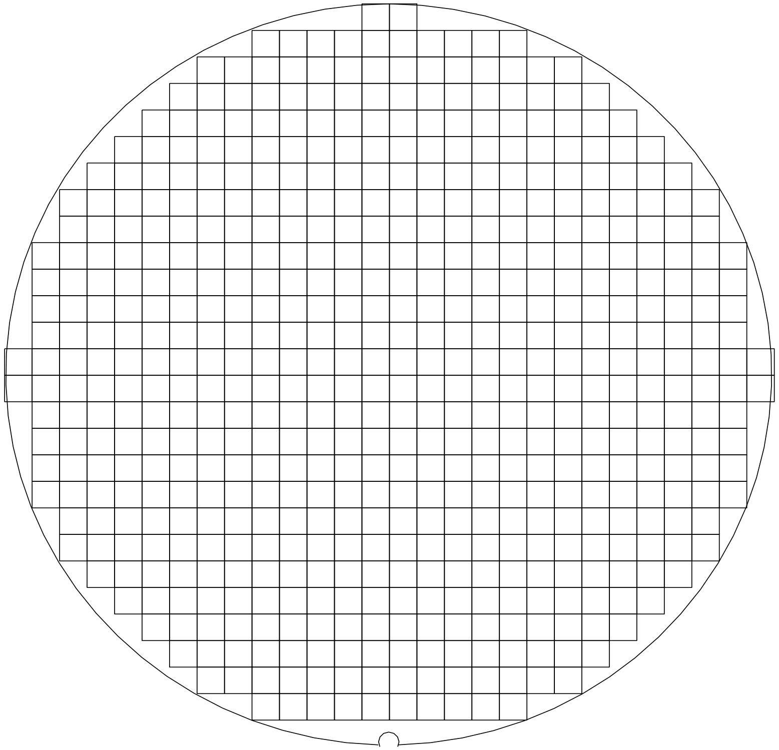
Reduced Partial Die on Large Wafer

14.5% partial die



200 mm

10.8% partial die



300 mm

Design for Testability

Outline

- Testing
  - Logic Verification
  - Silicon Debug
  - Manufacturing Test
- Fault Models
- Observability and Controllability
- Design for Test
  - Scan
  - BIST
- Boundary Scan

Testing

- Testing is the most expensive parts of chips
  - Logic verification accounts for > 50% of design effort for many chips
  - Debug time after fabrication has enormous opportunity cost
  - Shipping defective parts can sink a company
- Example: Intel FDIV bug
  - Logic error not caught until > 1M units shipped
  - Recall cost $450M (!!!)

Logic Verification

- Does the chip simulate correctly?
  - Usually done at HDL level
  - Engineers write test bench for HDL
    - Can’t test all cases
    - Look for corner cases
    - Try to break logic design
- Ex: 32-bit adder,test all combinations of corner cases as inputs:
    - 0, 1, 2, 2^{31}-1, -1, -2^{31}, a few random numbers

Silicon Debug

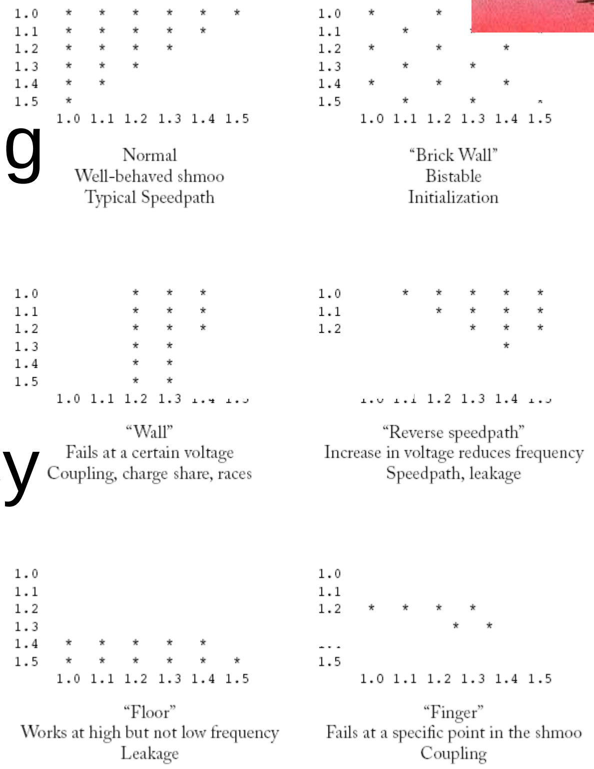
- Test the first chips back from fabrication
  - If you are lucky, they work the first time,If not…
- Logic bugs vs. electrical failures
  - Most chip failures are logic bugs from inadequate simulation
  - Some are electrical failures
    - Crosstalk,leakage, charge sharing
    - Ratio failures
  - A few are tool or methodology failures (e.g. DRC)
- Fix the bugs and fabricate a corrected chip

Shmoo Plots

- How to diagnose failures?
  - Hard to access chips
    - Picoprobes
    - Electron beam
    - Laser voltage probing
    - Built-in self-test
- Shmoo plots
  - Vary voltage, frequency
  - Look for cause of electrical failures

Clock period in ns on the left, frequency increases go
Voltage on the bottom, increase left to right

** indicates a failure*



```
1.0  *   *   *   *   *   *
1.1  *   *   *   *   *
1.2  *   *   *   *
1.3  *   *   *
1.4  *   *
1.5  *
    1.0 1.1 1.2 1.3 1.4 1.5
```

Normal
Well-behaved shmoo
Typical Speedpath

```
1.0  *       *
1.1      *
1.2  *       *       *
1.3      *       *
1.4  *       *       *
1.5      *       *       *
    1.0 1.1 1.2 1.3 1.4 1.5
```

"Brick Wall"
Bistable
Initialization

```
1.0          *   *   *
1.1          *   *   *
1.2          *   *   *
1.3          *   *
1.4          *   *
1.5          *   *
    1.0 1.1 1.2 1.3 1.4 1.5
```

"Wall"
Fails at a certain voltage
Coupling, charge share, races

```
1.0      *   *   *   *   *
1.1          *   *   *   *
1.2              *   *   *
                     *

    1.0 1.1 1.2 1.3 1.4 1.5
```

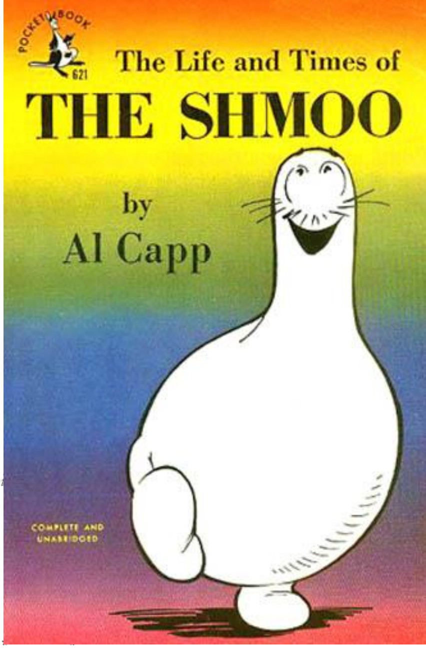
"Reverse speedpath"
Increase in voltage reduces frequency
Speedpath, leakage

```
1.0
1.1
1.2
1.3
1.4  *   *   *   *   *
1.5  *   *   *   *   *   *
    1.0 1.1 1.2 1.3 1.4 1.5
```

"Floor"
Works at high but not low frequency
Leakage

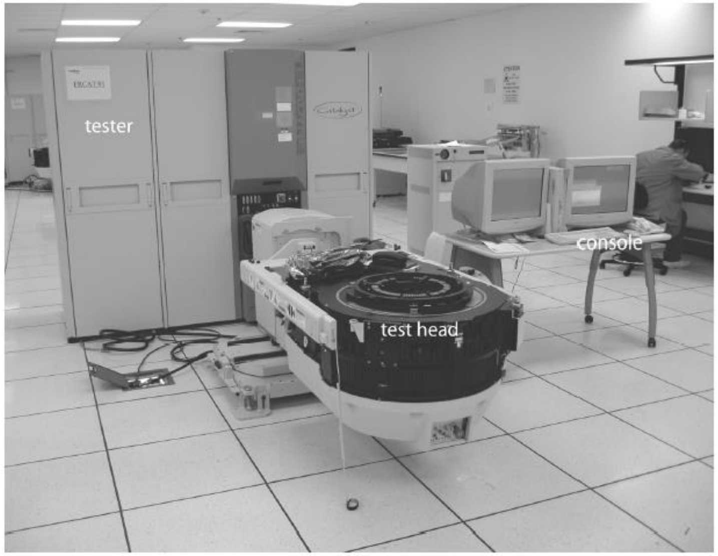
```
1.0
1.1
1.2  *   *   *   *
               *   *
...
1.5
    1.0 1.1 1.2 1.3 1.4 1.5
```

"Finger"
Fails at a specific point in the shmoo
Coupling



Manufacturing Test

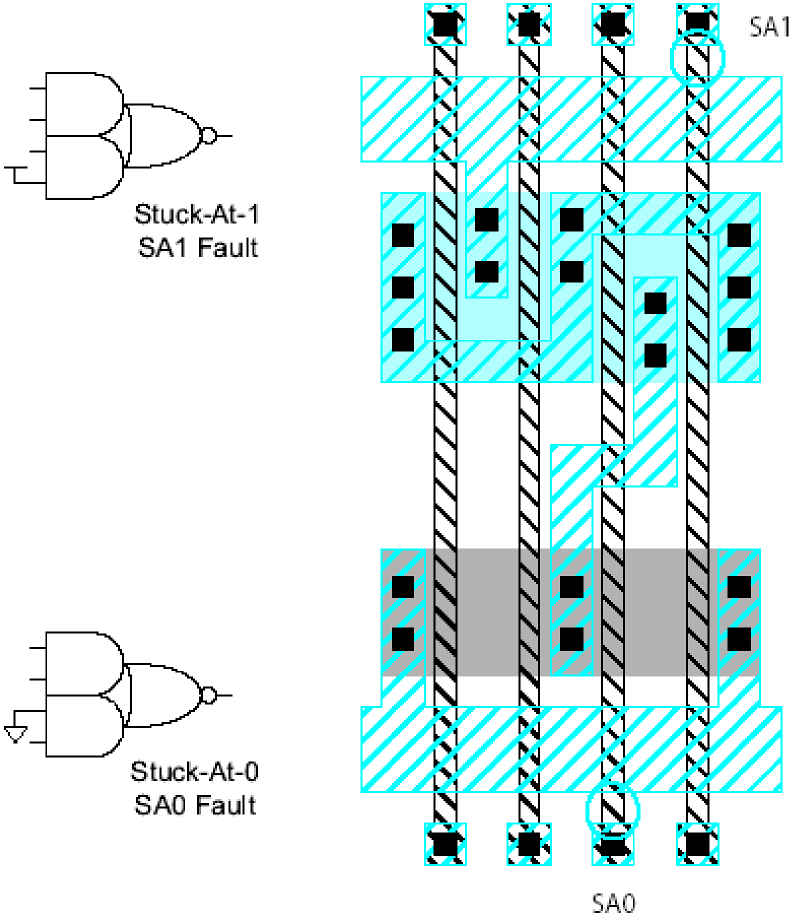
- A speck of dust on a wafer is sufficient to kill chip
- *Yield* of any chip is < 100%
    - Must test chips after manufacturing before delivery to customers to only ship good parts
- Manufacturing testers are very expensive
    - Minimize time on tester
    - Careful selection of *test vectors*



Stuck-At Faults

- How does a chip fail?
    - Usually failures are shorts between two conductors or opens in a conductor
    - This can cause very complicated behavior
- A simpler model: *Stuck-At*
    - Assume all failures cause nodes to be "stuck-at" 0 or 1, i.e. shorted to GND or V_{DD}
    - Not quite true, but works well in practice

Examples



Observability & Controllability

- ***Observability:*** ease of observing a node by watching external output pins of the chip
- ***Controllability:*** ease of forcing a node to 0 or 1 by driving input pins of the chip
- Combinational logic is usually easy to observe and control
- Finite state machines can be very difficult, requiring many cycles to enter desired state
  - Especially if state transition diagram is not known to the test engineer

Test Pattern Generation

- Manufacturing test ideally would **check every node** in the circuit to prove it is not stuck.
- **Apply the smallest sequence** of test vectors necessary to prove each node is not stuck.
- Good observability and controllability reduces number of test vectors required for manufacturing test.
  - Reduces the cost of testing
  - Motivates design-for-test

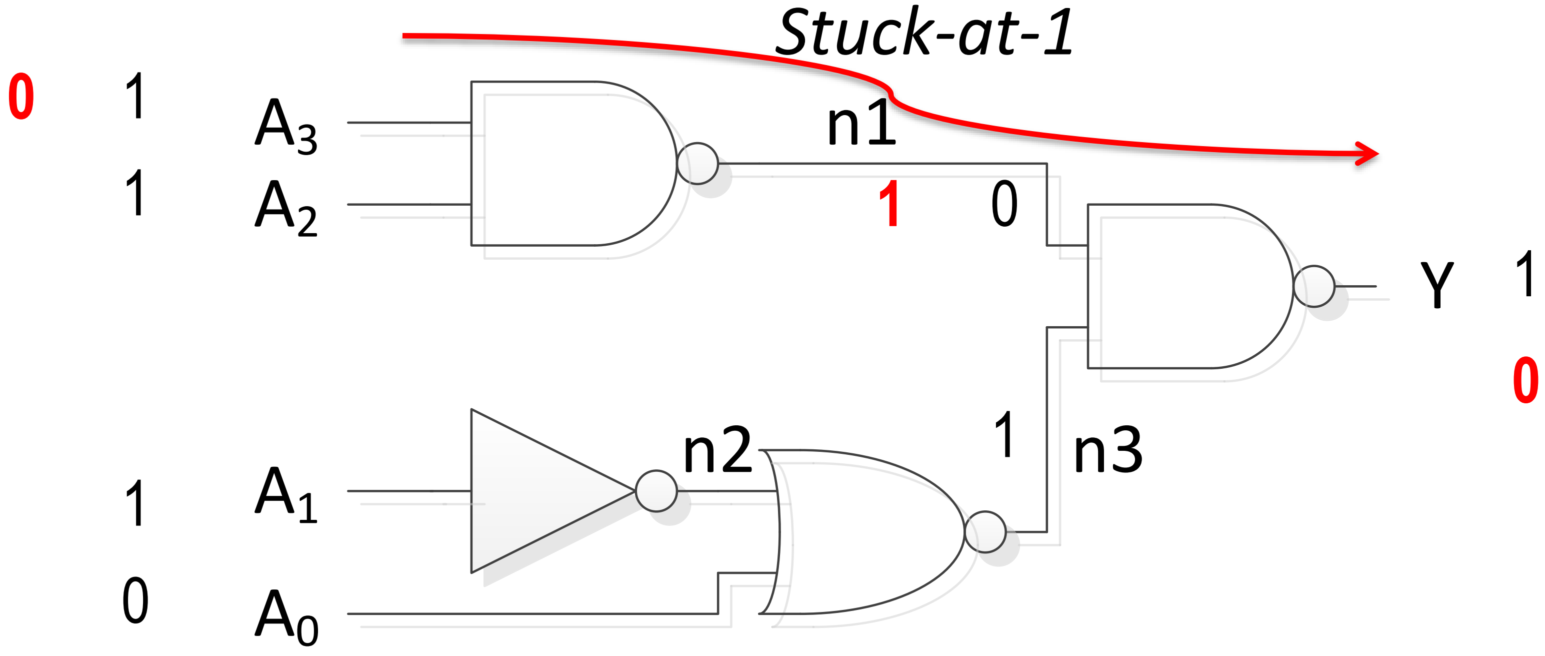
Design for Test

- Design the chip to increase observability and controllability

- If each register could be observed and controlled, test problem reduces to testing combinational logic between registers.

- Better yet, logic blocks could enter test mode where they generate test patterns and report the results automatically.

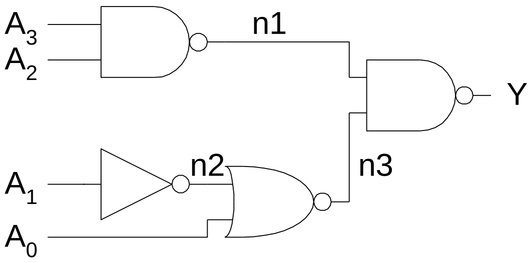
Test Example



Test Example

	SA1	Y	SA0	Y
A_3	{0110}	0/1	{1110}	
A_2	{1010}	0/1	{1110}	
A_1	{0100}	1/0	{0110}	
A_0	{0110}	0/1	{0111}	
n1	{1110}	1/0	{0110}	
n2	{0110}	0/1	{0100}	
n3	{0101}	1/0	{0110}	
Y	{0110}	0/1	{1110}	

Controllability ***Observability***



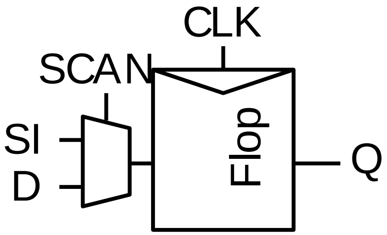
- Minimum set: {0100, 0101, 0110, 0111, 1010, 1110}

Design for Test

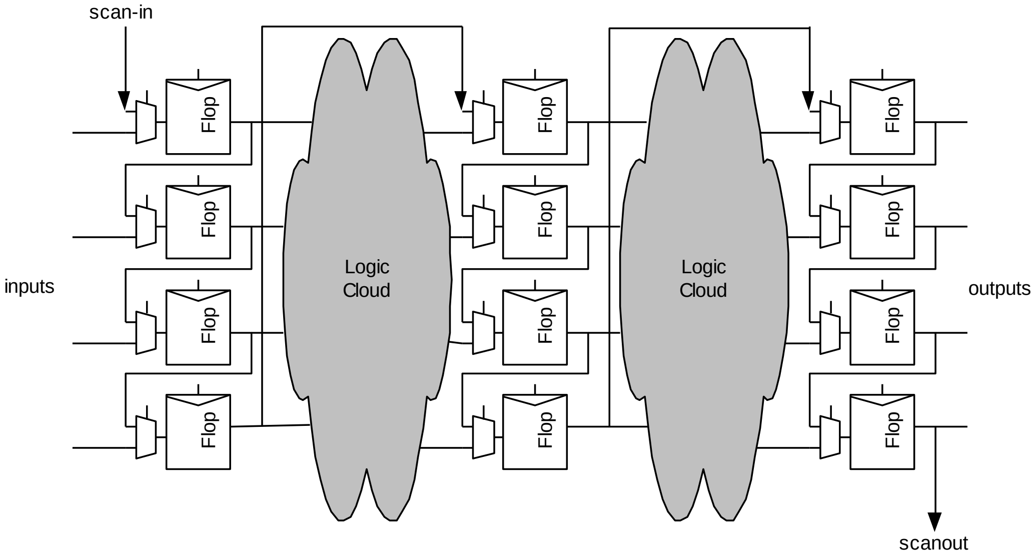
- Design the chip to increase **observability** and **controllability**
- If each register could be observed and controlled, test problem reduces to testing combinational logic between registers.
- Better yet, logic blocks could enter test mode where they generate test patterns and report the results automatically.

Scan

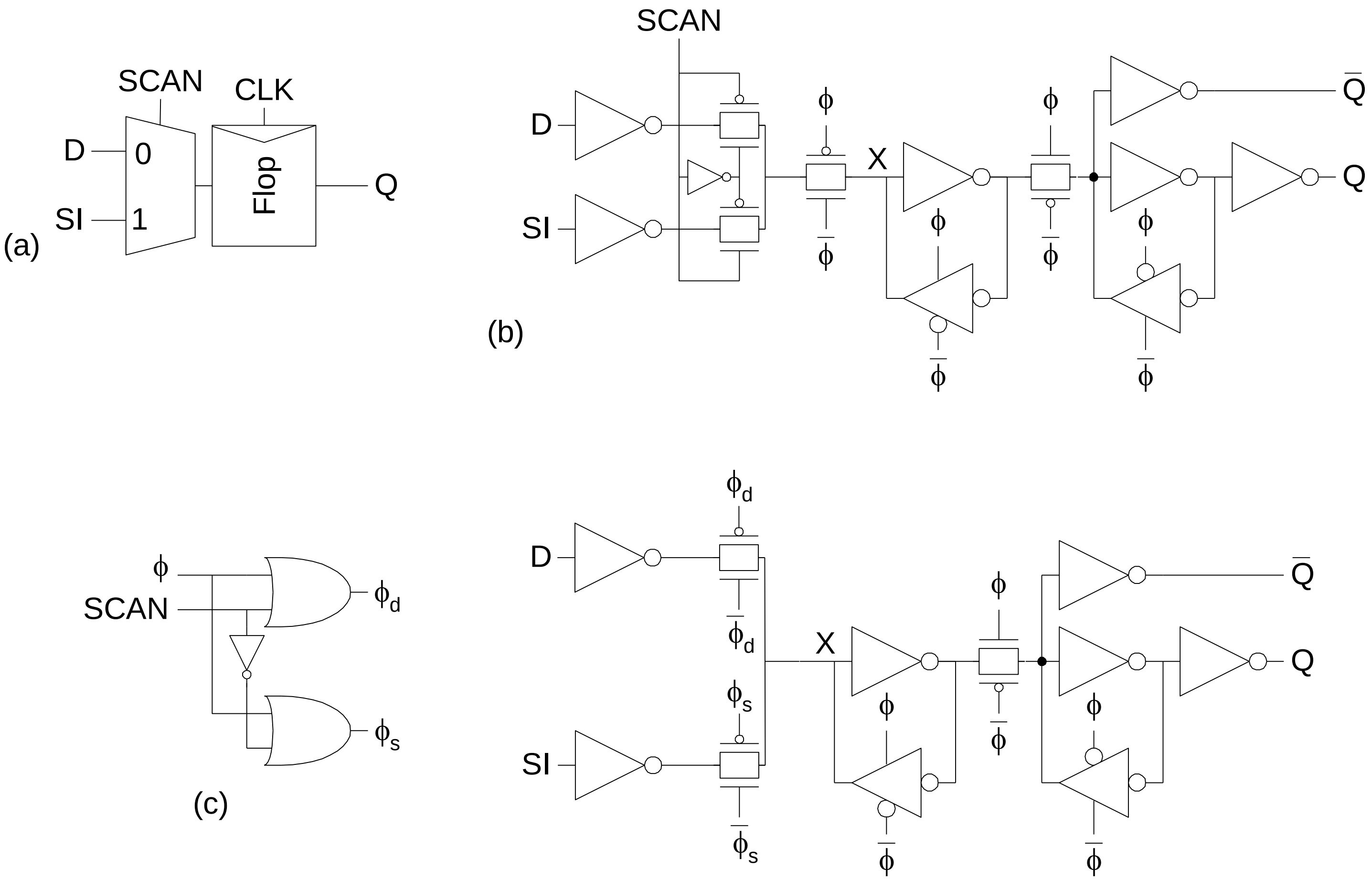
- Convert each flip-flop to a scan register
  - Only costs one extra multiplexer
- Normal mode: flip-flops behave as usual
- Scan mode: flip-flops behave as shift register



- Contents of flops can be scanned out and new values scanned in



Scannable Flip-flops

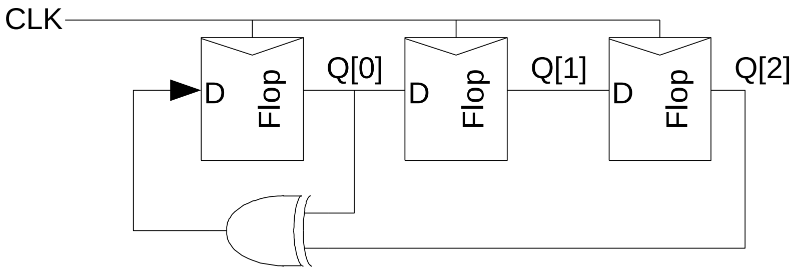


Built-in Self-test

- Built-in self-test lets blocks test themselves
  - Generate pseudo-random inputs to comb. logic
  - Combine outputs into a *syndrome*
  - With high probability, block is fault-free if it produces the expected syndrome

PRSG

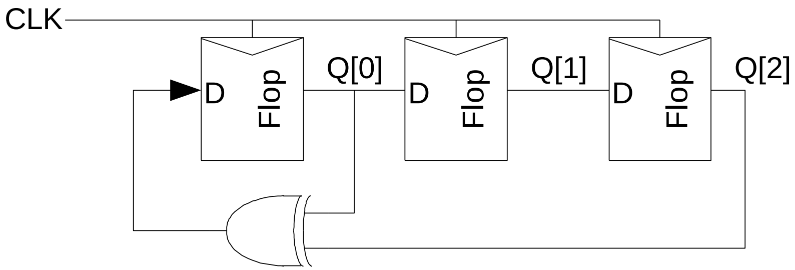
- *Linear Feedback Shift Register*
  - Shift register with input taken from XOR of state
  - *Pseudo-Random Sequence Generator*



Step	Q
0	111
1	
2	
3	
4	
5	
6	
7	

PRSG

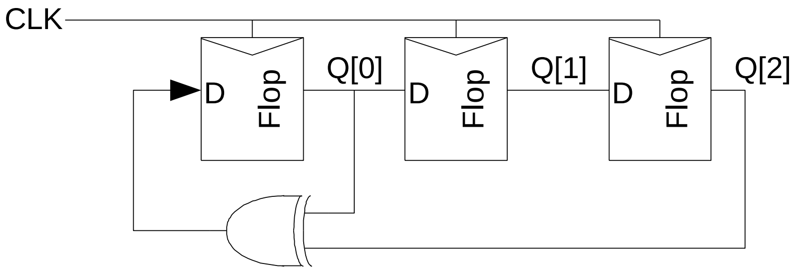
- *Linear Feedback Shift Register*
  - Shift register with input taken from XOR of state
  - *Pseudo-Random Sequence Generator*



Step	Q
0	111
1	110
2	
3	
4	
5	
6	
7	

PRSG

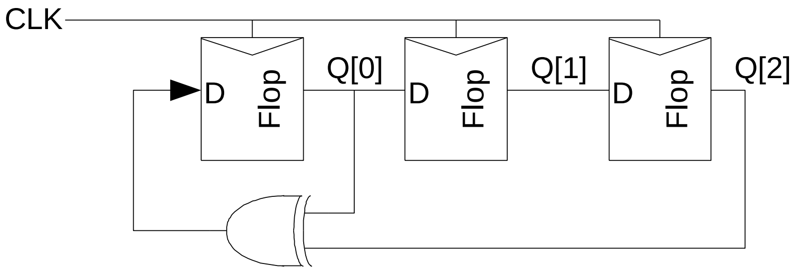
- *Linear Feedback Shift Register*
  - Shift register with input taken from XOR of state
  - *Pseudo-Random Sequence Generator*



Step	Q
0	111
1	110
2	101
3	
4	
5	
6	
7	

PRSG

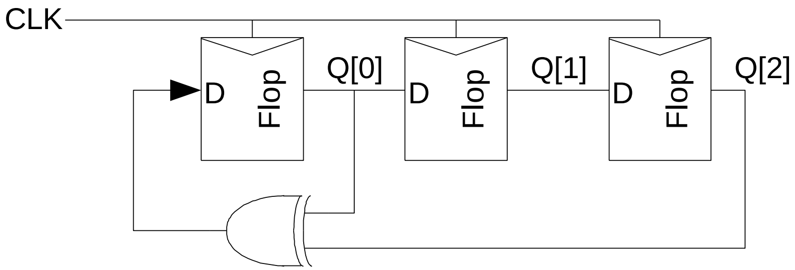
- *Linear Feedback Shift Register*
  - Shift register with input taken from XOR of state
  - *Pseudo-Random Sequence Generator*



Step	Q
0	111
1	110
2	101
3	010
4	
5	
6	
7	

PRSG

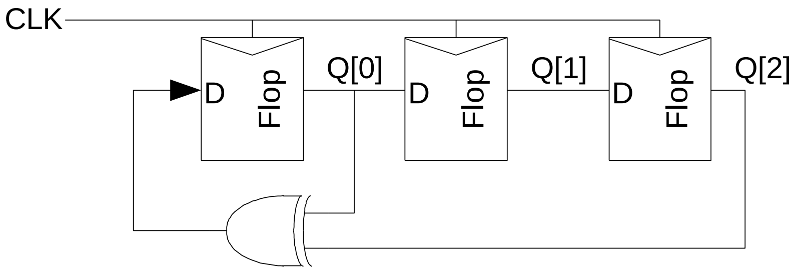
- *Linear Feedback Shift Register*
  - Shift register with input taken from XOR of state
  - *Pseudo-Random Sequence Generator*



Step	Q
0	111
1	110
2	101
3	010
4	100
5	
6	
7	

PRSG

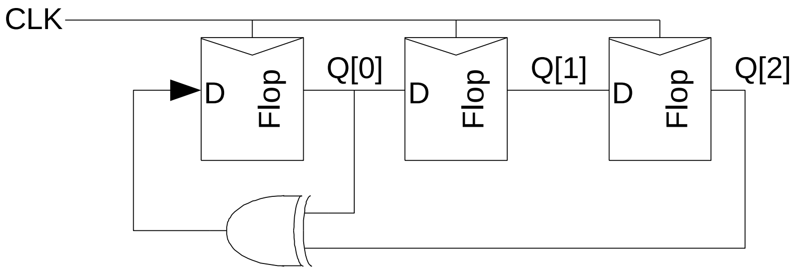
- *Linear Feedback Shift Register*
  - Shift register with input taken from XOR of state
  - *Pseudo-Random Sequence Generator*



Step	Q
0	111
1	110
2	101
3	010
4	100
5	001
6	
7	

PRSG

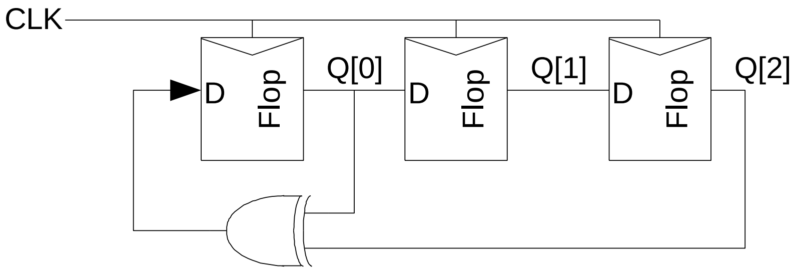
- *Linear Feedback Shift Register*
  - Shift register with input taken from XOR of state
  - *Pseudo-Random Sequence Generator*



Step	Q
0	111
1	110
2	101
3	010
4	100
5	001
6	011
7	

PRSG

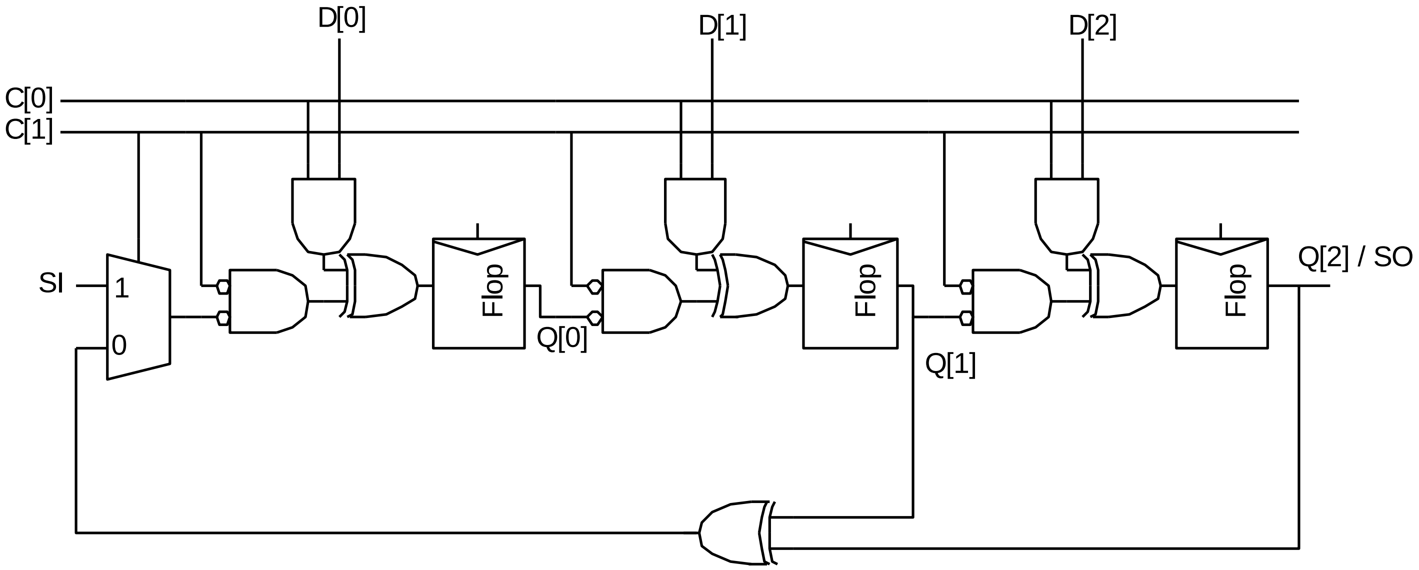
- *Linear Feedback Shift Register*
  - Shift register with input taken from XOR of state
  - ***P**seudo-**R**andom **S**equence **G**enerator*

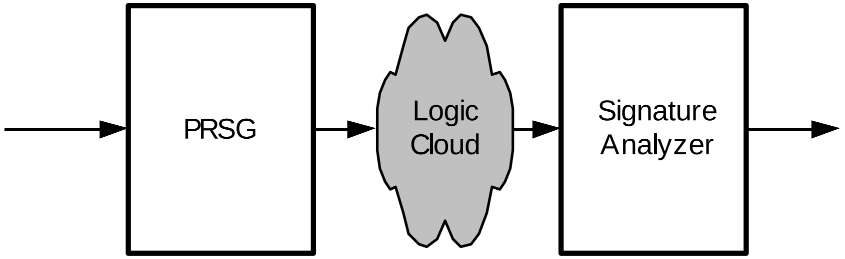


Step	Q
0	111
1	110
2	101
3	010
4	100
5	001
6	011
7	111 (repeats)

BILBO

- Built-in Logic Block Observer
  - Combine scan with PRSG & signature analysis



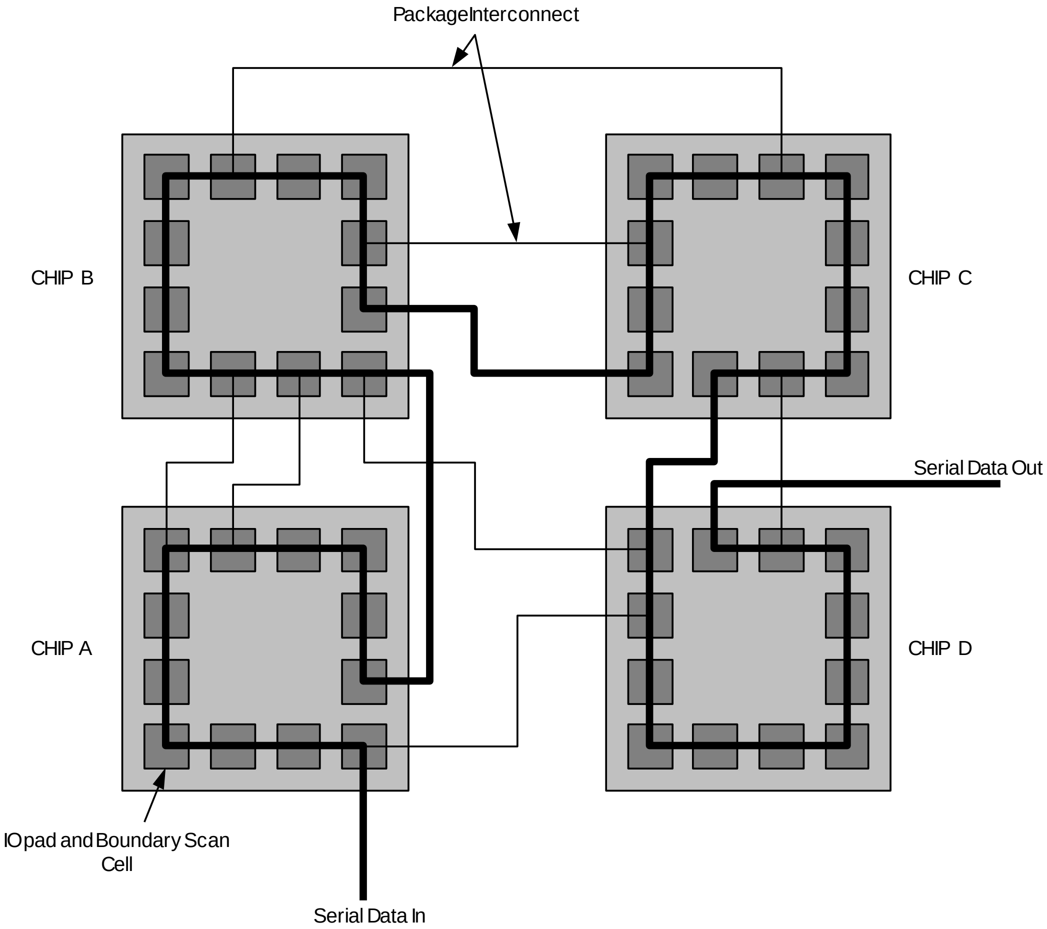


MODE	**C[1]**	**C[0]**
Scan	0	0
Test	0	1
Reset	1	0
Normal	1	1

Boundary Scan

- Testing boards is also difficult
    - Need to verify solder joints are good
        - Drive a pin to 0, then to 1
        - Check that all connected pins get the values
- Through-hold boards used “bed of nails”
- SMT and BGA boards cannot easily contact pins
- Build capability of observing and controlling pins into each chip to make board test easier

Boundary Scan Example



Boundary Scan Interface

- Boundary scan is accessed through five pins
  - TCK: test clock
  - TMS: test mode select
  - TDI: test data in
  - TDO: test data out
  - TRST*: test reset (optional)

- Chips with internal scan chains can access the chains through boundary scan for unified test strategy.

Summary

- Think about testing from the beginning
  - Simulate as you go
  - Plan for test after fabrication

“If you don´ t test it, it won´ t work! (Guaranteed)”