(IP Protection using Fingerprinting)

IP Piracy Detection: Fingerprinting

- **Why Fingerprinting ?**
  - Watermark cannot distinguish different IP users (or copies of the same IP).
  - If IP infringement is discovered, how to determine which IP user has misused the IP?
  - Need to identify each copy of the IP

- Digital fingerprinting is a protocol that makes each copy of the object unique and distinguishable.

- It helps the defender to track the source of piracy by embedding the signature of the buyer (i.e. his public key) along with the watermark of the designer.

- When challenged, the designer can reveal the watermark to claim the ownership and the buyer's signature to reveal the source of piracy.

- **For example**, the power, timing, or thermal fingerprint of an IC is revealed on applying a set of input vectors.

Fingerprinting Vs Watermarking

- **Fingerprinting vs. Watermarking**
  - Both are (invisible) identification codes permanently embedded as an integral part within a design for IPP.
  - **Watermark:** same for all copies
  - **Fingerprint:** unique for each copy

- **Fingerprint = multiple distinct watermarks**

- **Basic needs for fingerprinting methods:**
  - Effective method to create fingerprints
  - Collusion-free distribution of fingerprints

Fingerprinting : Requirements

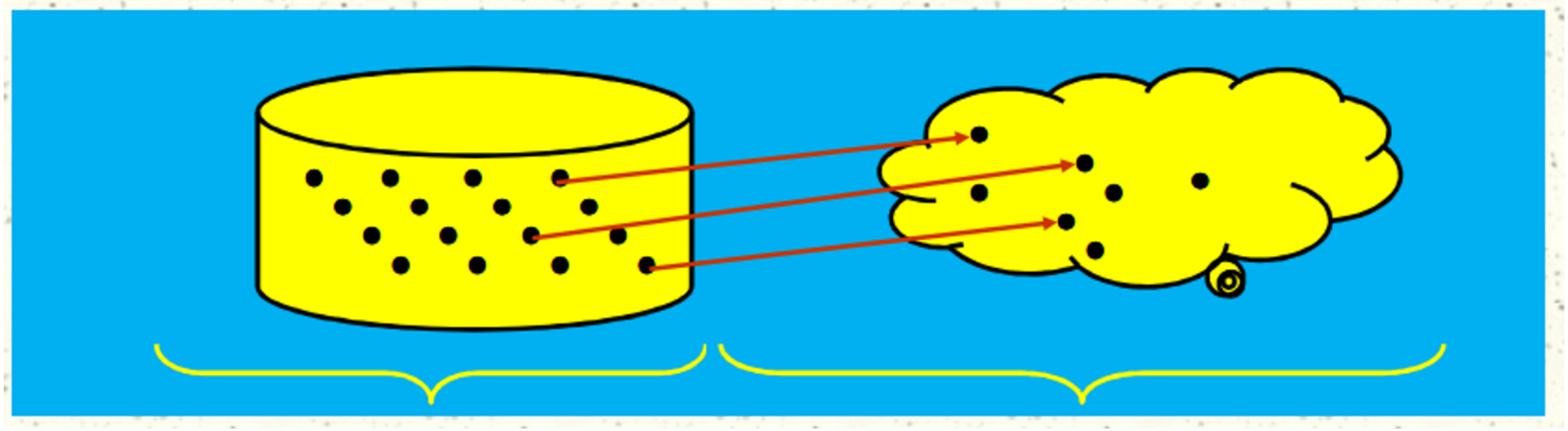
- A fingerprint, being the signature of the buyer, should satisfy all the requirements of any effective watermarks:-
  - **High credibility:-** The fingerprint should be readily detectable in proving legal ownership, and the probability of coincidence should be low.
  - **Low overhead:-** To minimize the impact of fingerprinting on the quality of the software or design.
  - **Resilience:-** The fingerprint should be difficult or impossible to remove without complete knowledge of the software or design.
  - **Transparency:-** The addition of fingerprints to software and designs should be completely transparent, so that fingerprinting can be used with existing design tools.
  - **Part protection:-** Ideally, a good fingerprint should be distributed all over the software or design in order to identify the buyer from any part of it.
  - **Preserving watermarks:-** Fingerprinting should not diminish the strength of the author's watermark.

Fingerprinting : Requirements

- Three key requirements for fingerprinting protocols:
  - A fingerprinting protocol must be capable of generating solutions that are ”faraway” from each other. If solutions are too similar, it will be difficult for the seller to identify distinct buyers and it will be easy for dishonest buyers to collude.
  - A fingerprinting protocol should be nonintrusive to existing design optimization algorithms, so that it can be easily integrated with existing software tool flows.

    The fingerprinting system must seamlessly integrate with existing design optimization algorithms and software tool flows without disrupting their normal operation
  - The cost of the fingerprinting protocol should be kept as low as possible. Ideally, it should be negligible compared to the original design effort.

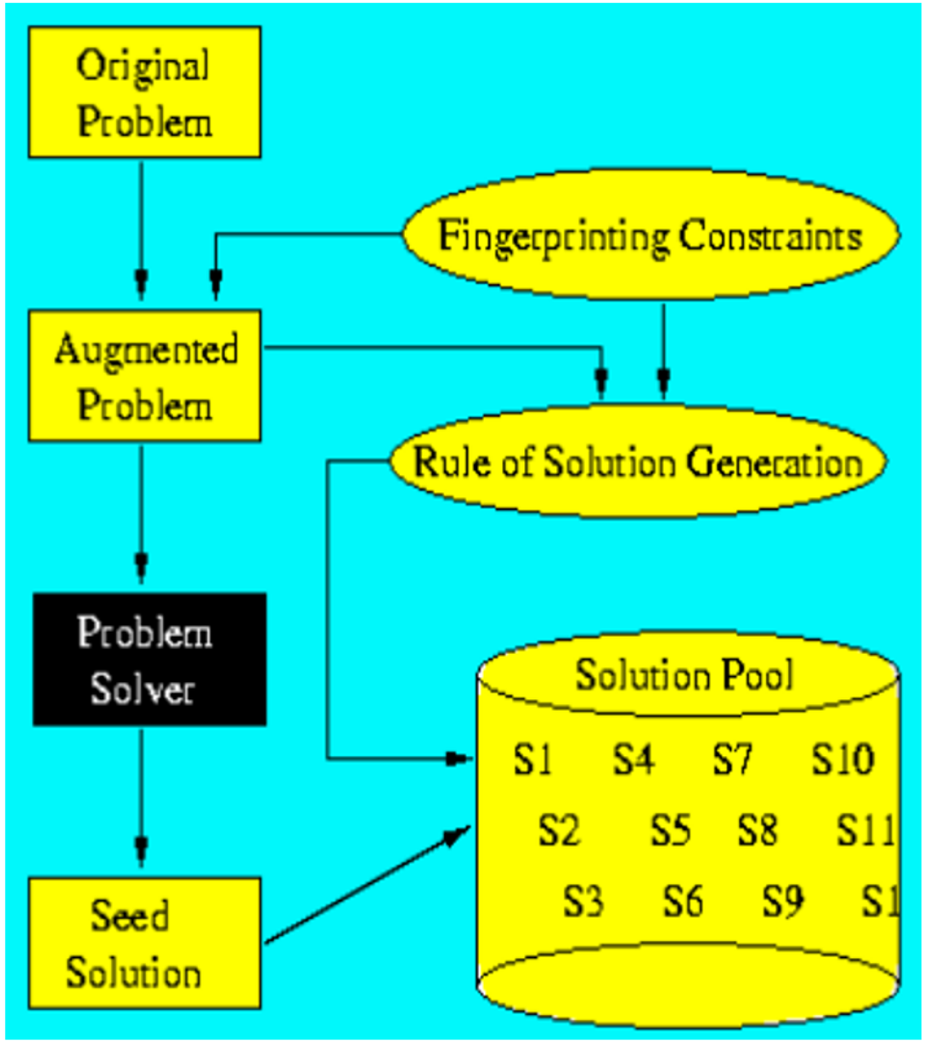
Fingerprinting Approach and Challenges



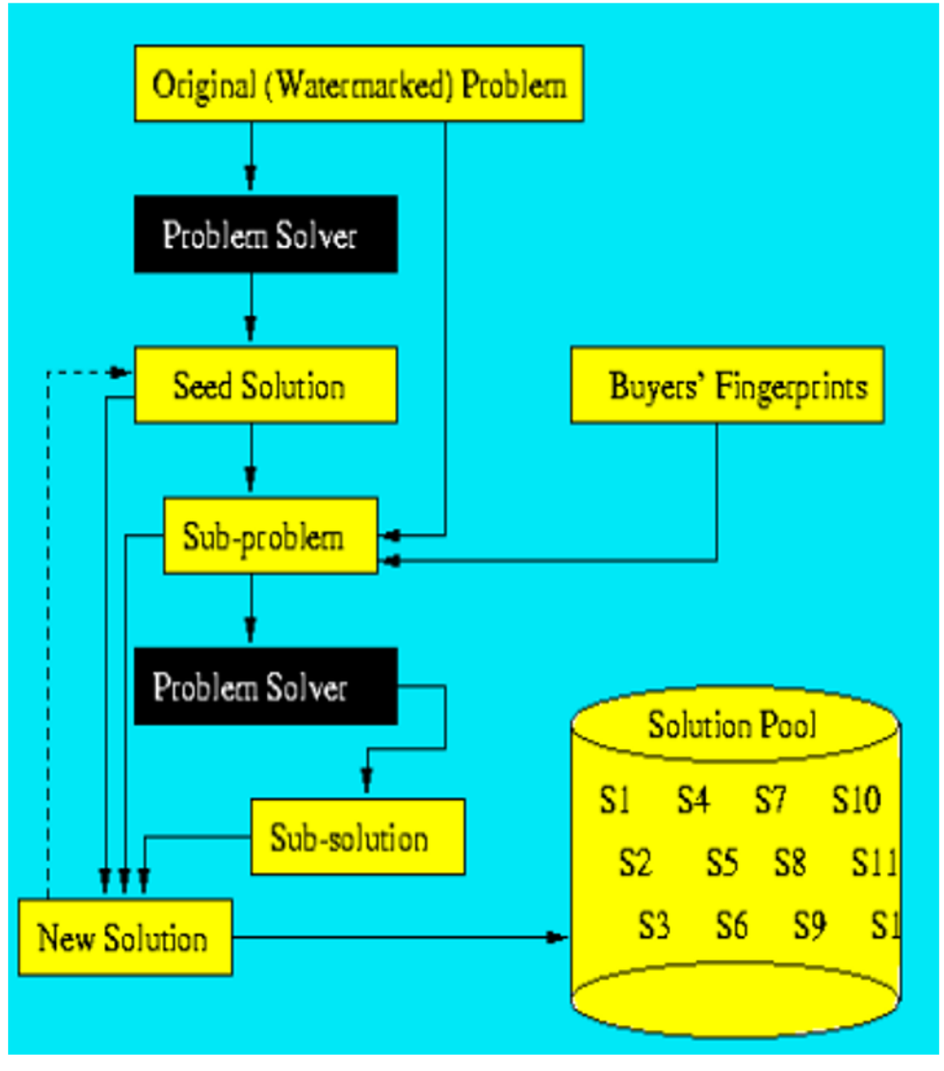
- Generation protocol
  - Quantity
  - Quality
  - Run-time
- Distribution protocol
  - Uniqueness
  - Robustness
  - Collusion-free

Two Fingerprinting Techniques

Constraint addition

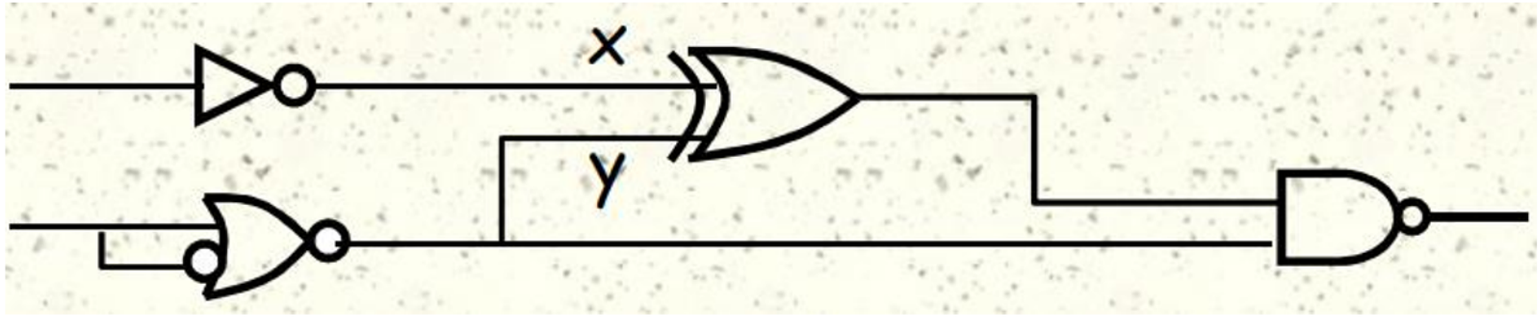


Constraint addition



Don’t Cares: SDC

- **Satisfiability don’t cares (SDC):** SDC of a subcircuit/subsystem consist of all input patterns that will never occur.
- **Example:**

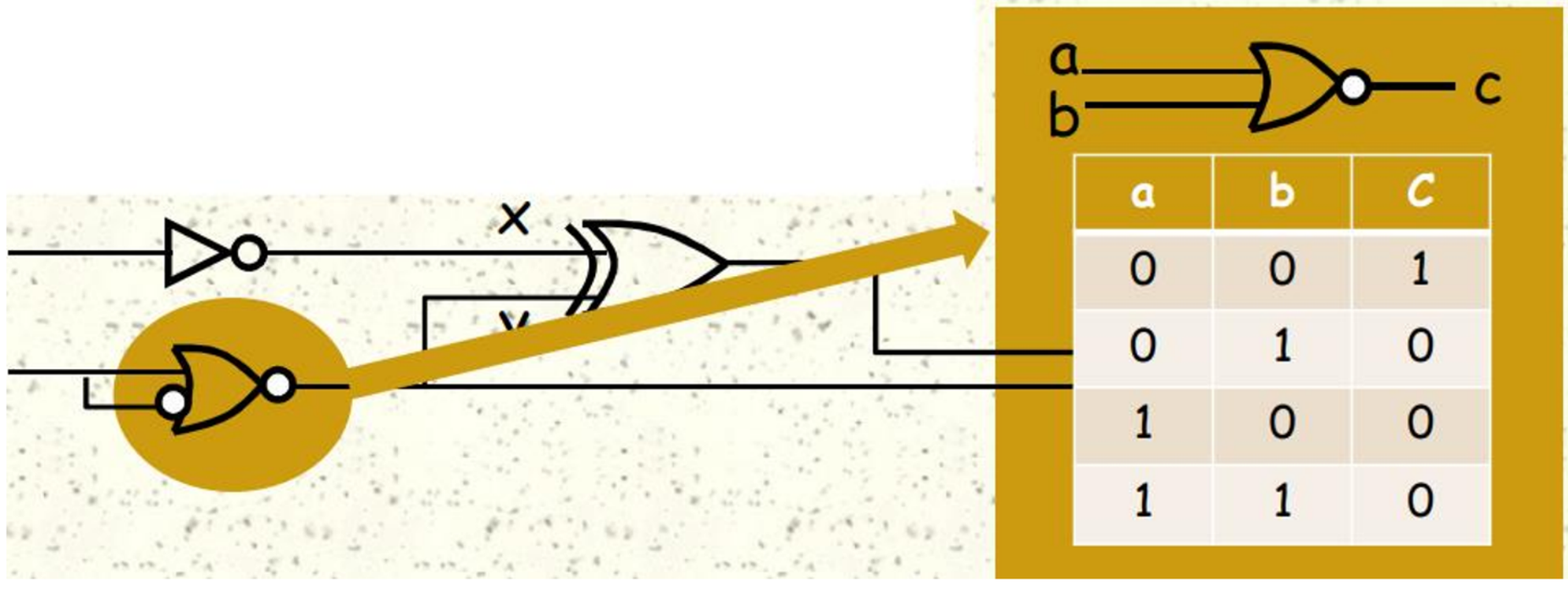


- **In this example circuit**, signal y cannot be 1.

Don’t Cares: SDC

- **Satisfiability don’t cares (SDC)** of a subcircuit/subsystem consist of all input patterns that will never occur.

- **Example:** from definition, signal y cannot be 1.

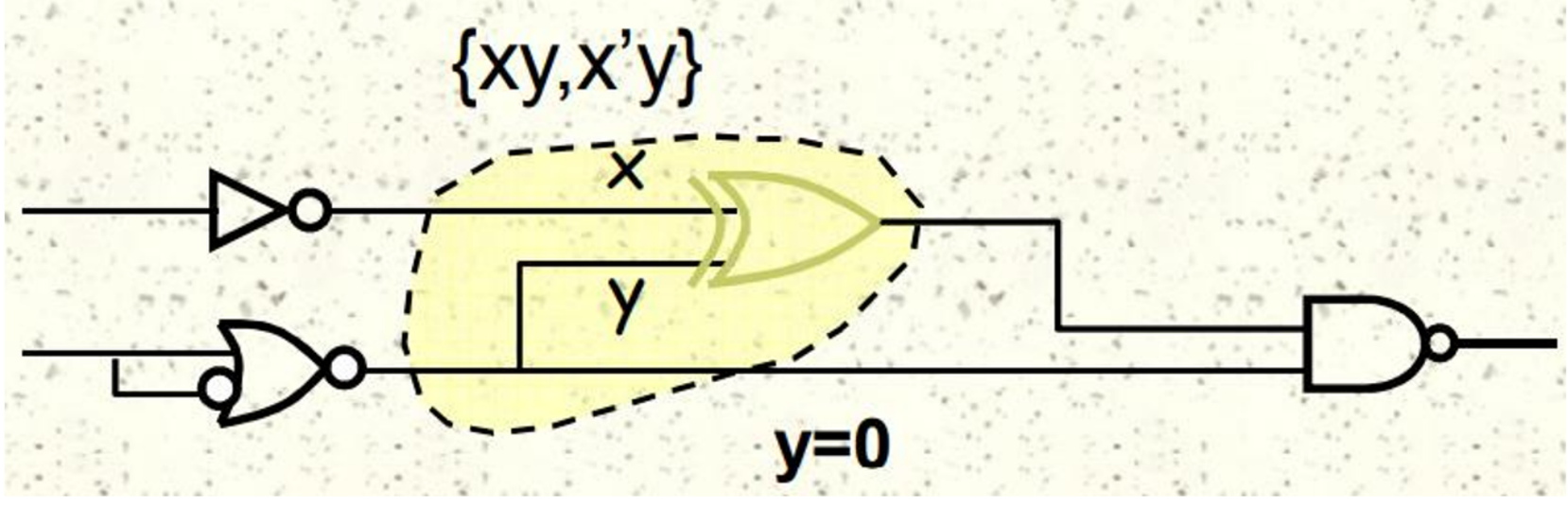


a	b	c
0	0	1
0	1	0
1	0	0
1	1	0

Don’t Cares: SDC

- Satisfiability don’t cares (SDC) of a subcircuit/subsystem consist of all input patterns that will never occur.

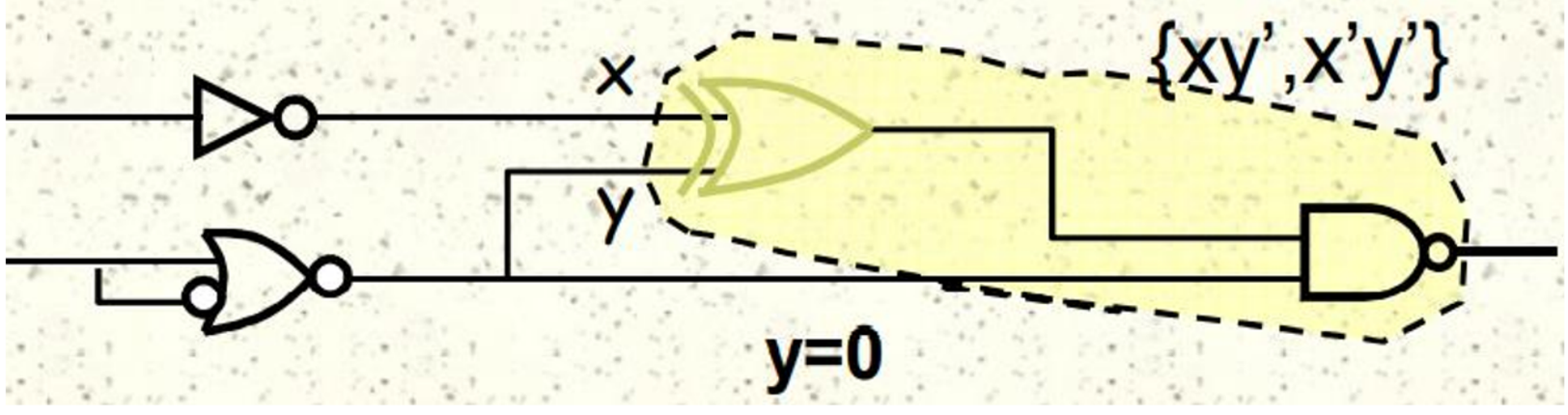
- **Example:** for XOR, y input cannot be 1, so {x=1,y=1} and {x=0,y=1} will be SDCs.



Don’t Cares: ODC

- **Observability don’t cares (ODC)** of a subcircuit/subsystem are the input patterns that represent situations when an output is not observed.

- **Example:** when y=0, output of the XOR cannot be observed. {x=1,y=0}, {x=0,y=0}

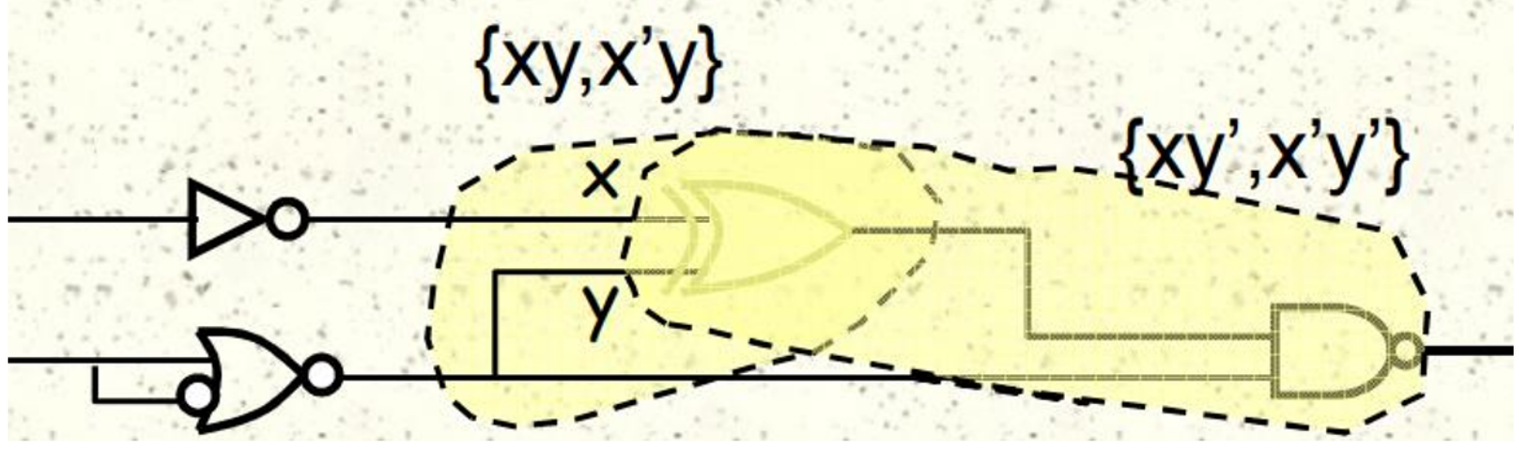


Don’t Cares: SDC and ODC

- **Satisfiability don’t cares (SDC)** of a subcircuit/subsystem consist of all input patterns that will never occur.

- **Observability don’t cares (ODC)** of a subcircuit/subsystem are the input patterns that represent situations when an output is not observed.

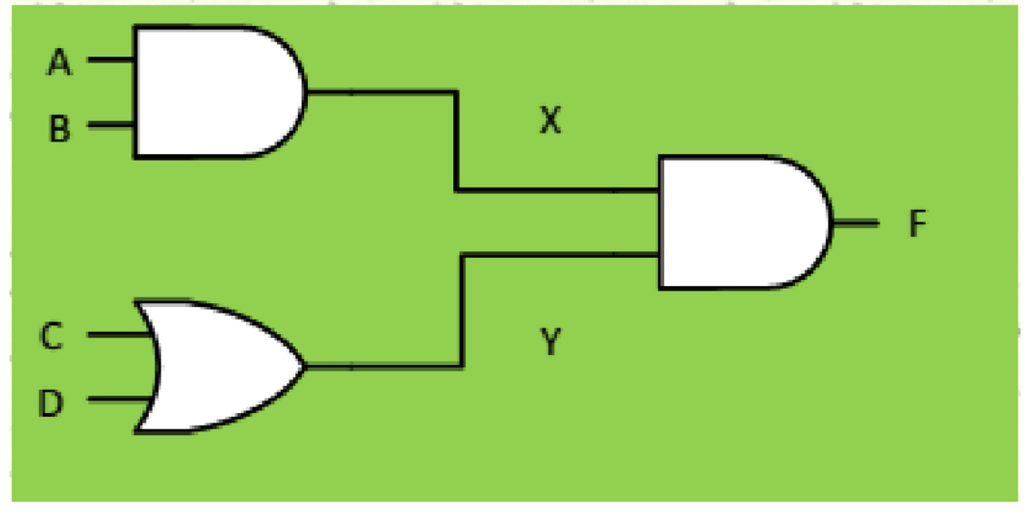
- **SDCs: {x=1,y=1}, {x=0,y=1}**
- **ODCs: {x=1,y=0}, {x=0,y=0}**

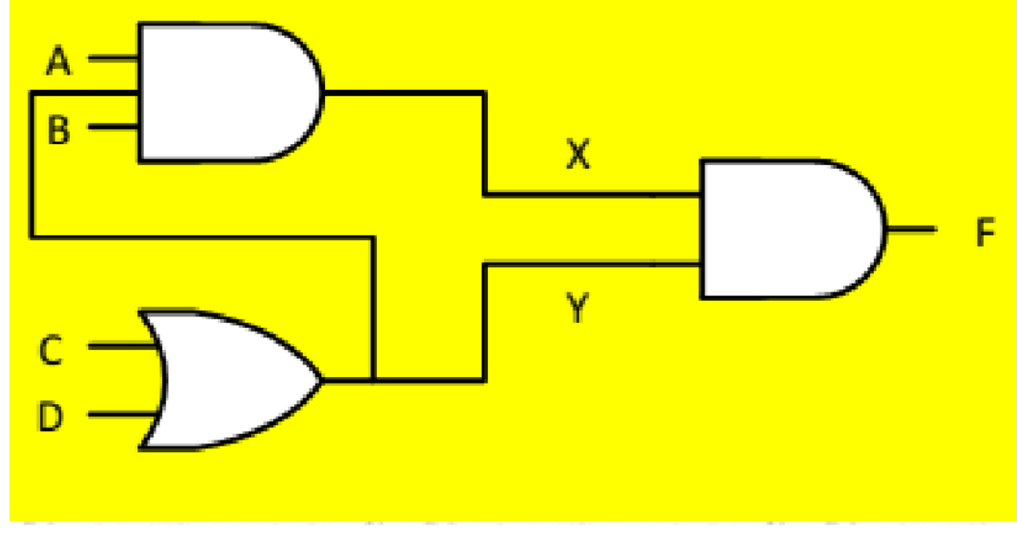


- **All the four input combinations are don’t care, so the XOR gate can be removed!**

Fingerprinting: Don’t Cares

- **Fingerprinting: Don’t Cares (I)**
  - **Observability don’t care:** Input patterns that represent situations when an output is not observed
    - **X = AB, Y= C+D**
    - When Y=0, signal X cannot be observed
    - **ODCs:** XY’, X’Y’
    - **X = ABY**
      - When Y=1, X = AB
      - When Y=0, ODCs
    - Both are functionally identical
- 2 distinct IPs are possible
- For n such sub circuits, 2^n distinct copies are possible

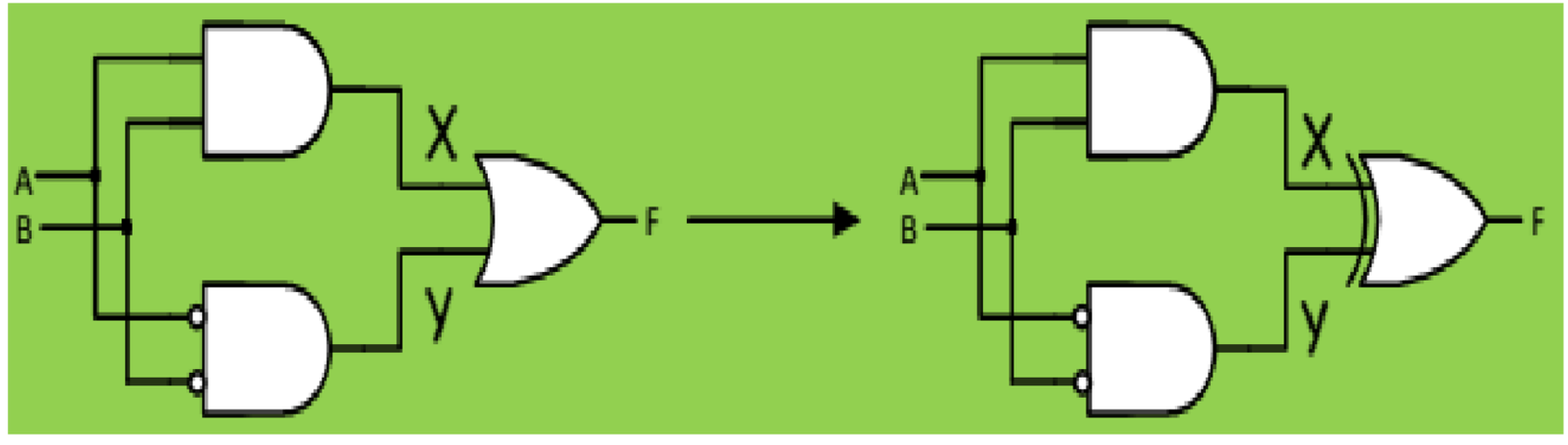




Fingerprinting: Don’t Cares (Cont…)

- **Fingerprinting: Don’t Cares (II)**
  - **Satisfiability don’t care:** Input patterns that will never occur
    - **X = AB, Y= A’B’**

A	B	X	Y
0	0	0	1
0	1	0	0
1	0	0	0
1	1	1	0



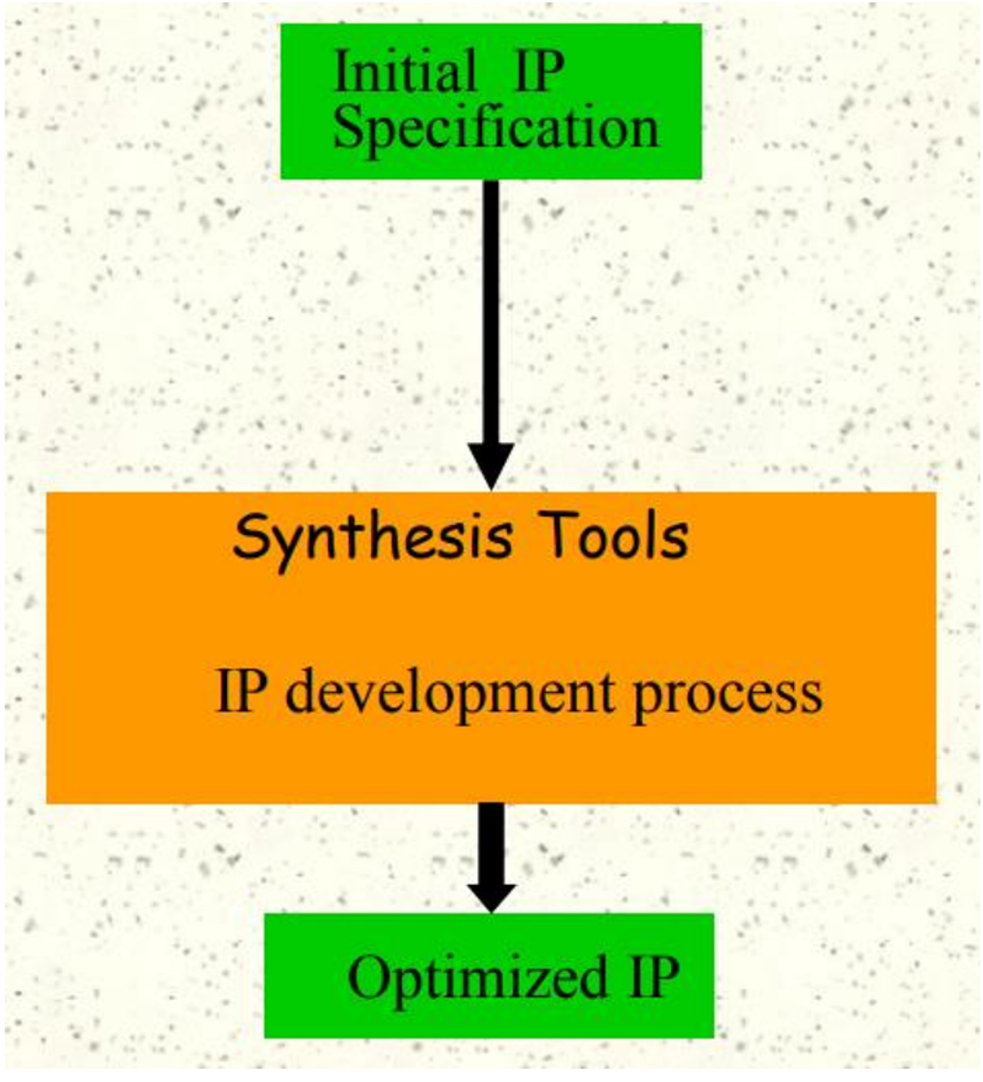
- SDCs: XY
- OR(X,Y) ≠ XOR(X,Y)

X	Y	OR	XOR
0	0	0	0
0	1	1	1
1	0	1	1
1	1	1	0

- 2 distinct IPs are possible
- For *n* SDCs, 2^n distinct copies are possible

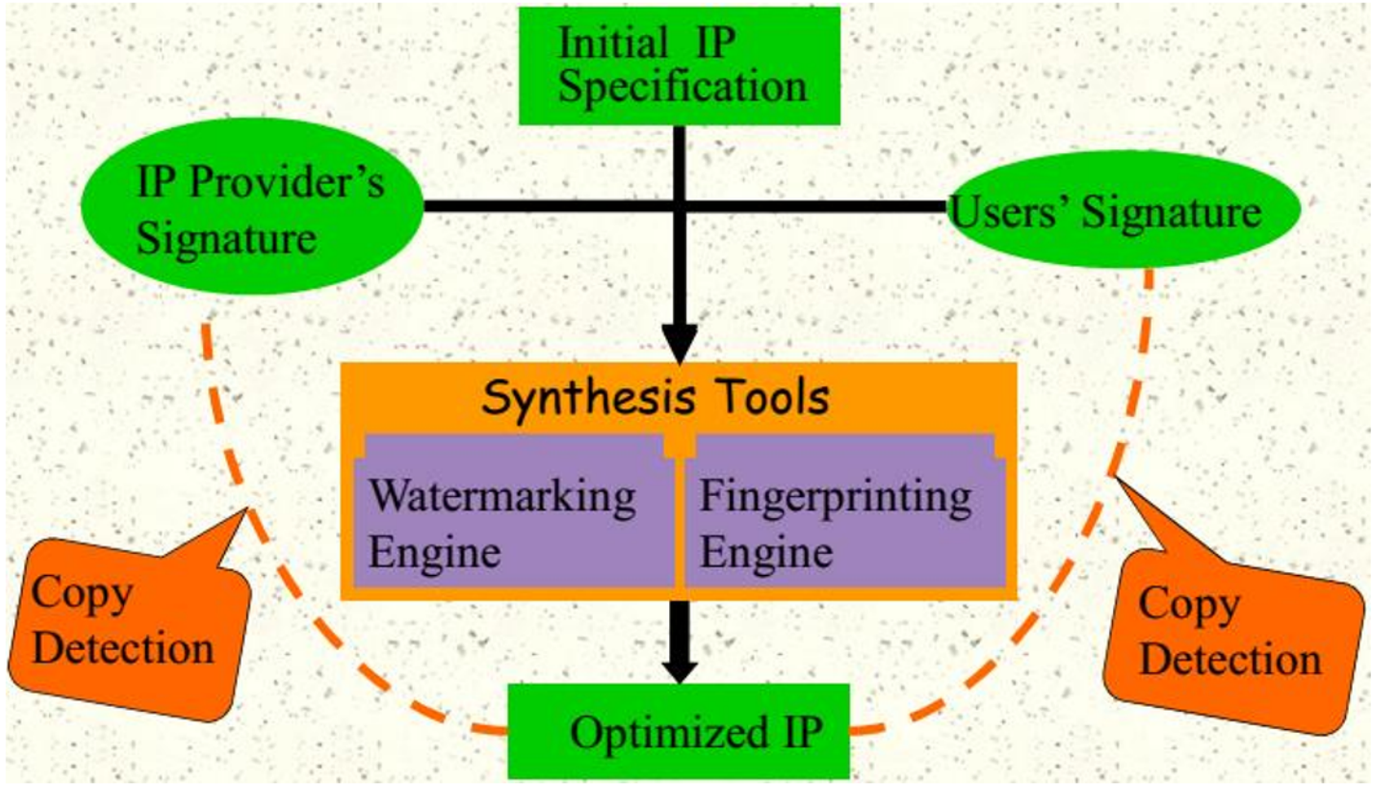
Design Without IP Protection

- **Design without IP Protection:**



Fingerprinting and Watermarking

- **Design With IP Protection using Watermarking and Fingerprinting:**



- ***Note:*** *Both digital watermarking and digital fingerprinting are deterrents to IP piracy, they do not prevent attacker from misusing the IPs.*

Thank You!