IIT ROORKEE

FREE ONLINE EDUCATION swayam शिक्षित भारत, उन्नत भारत

NPTEL ONLINE CERTIFICATION COURSE

VLSI Physical Design with Timing Analysis

Lecture – 8: Timing Arcs and Unateness

Bishnu Prasad Das

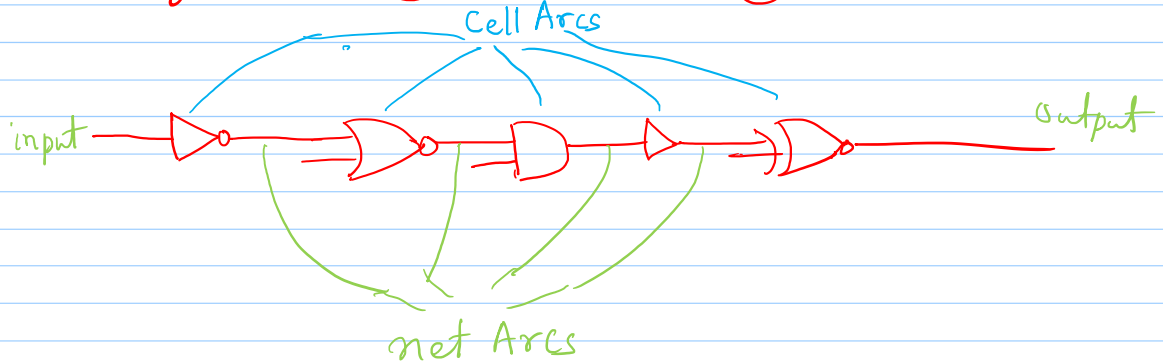
Department of Electronics and Communication Engineering

Contents

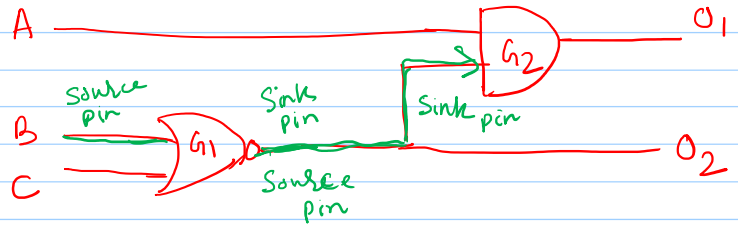
- Definition of Timing arcs
- Source Pin and Sink pin
- Unatenesss
  - (1) Positive Unate,
  - (2) Negative Unate
  - (3) Non-Unate
- Use of Unateness

Timing Arcs

- It is defined as the relationship between 2-pins of a logic gate
- It is useful for the STA tools to do timing analysis
- It is basically a part of the timing path
- Two types – (1) cell Arcs (2) net Arcs



Source Pin and Sink Pin



For gate G_1, B and C are source pin and O_2 is the sink pin

For gate G_2, A and O_2 are source pin and O_1 is the sink pin

In case of cell Arcs; input pins of cell are source pin
Output pins of cell are sink pin

In case of net Arcs; Output pin of cell are source pin
input pin of the cells are sink pin

Unateness

Each timing Arcs has a **timing sense.**

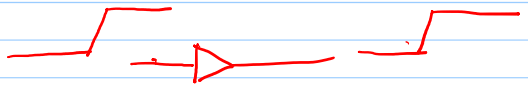
⇓

Change in the output transition w.r.t input transition.

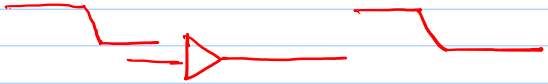
(1) Positive Unate

input signal is "Rising" → Output signal is "rising" or No change.
" " is "falling" → " " " "falling or No change

Buffers (i)

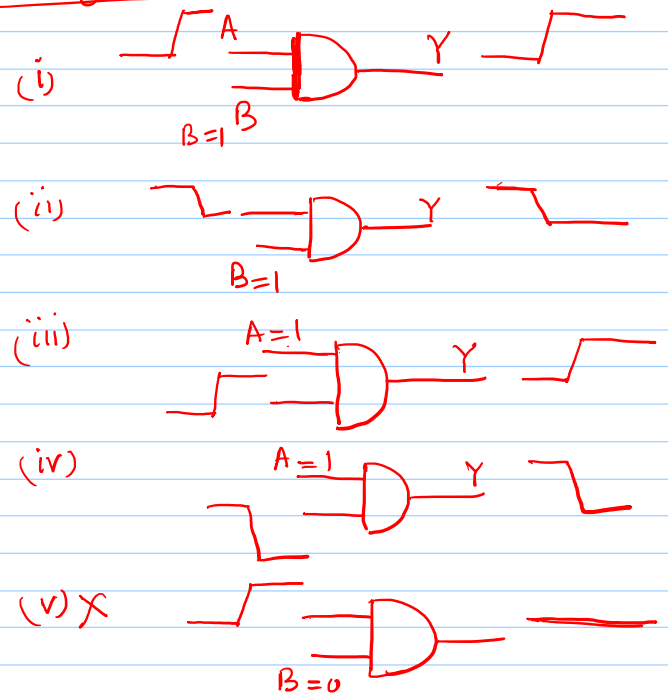


(ii)



AND gate

AND gate has 4 timing Arcs

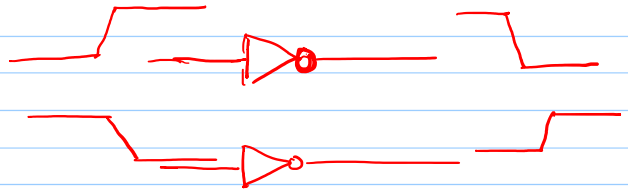


(2) Negative Unate

input signal is "rising" → output signal is "falling" or no change

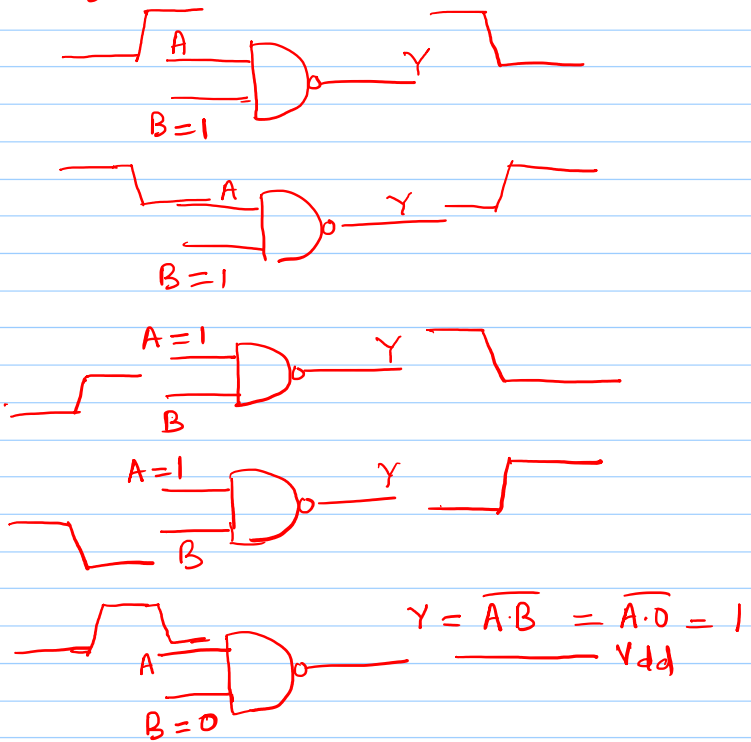
input " " "falling" → " " is "rising" or no change

(i) Inverter



2 timing Arcs

(ii) 2-input NAND gate



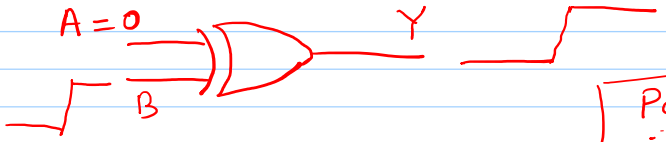
4 timing Arcs

$Y = \overline{A \cdot B} = \overline{A \cdot 0} = 1$

V_{dd}

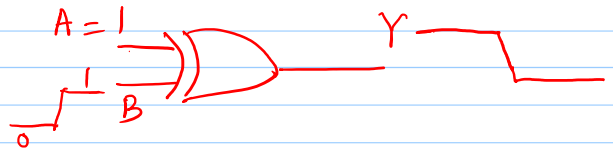
Non-Unate gate

XOR gate



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

Positive unate if A = 0



Negative unate if A = 1

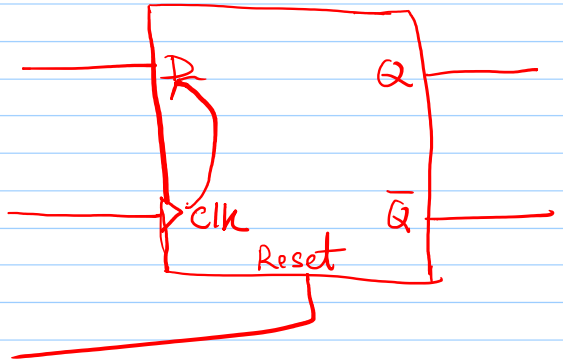
```
pin (OUT) {
  max_transition : 1.0;
  timing() {
    related_pin : "INP";
    timing_sense : negative_unate;
    rise_transition(delay_template_3x3) {
      index_1 ("0.1, 0.3, 0.7"); /* Input transition */
      index_2 ("0.16, 0.35, 1.43"); /* Output capacitance */
      values ( /*   0.16      0.35     1.43 */ \
        /* 0.1 */  "0.0417,  0.1337,  0.4680", \
        /* 0.3 */  "0.0718,  0.1827,  0.5676", \
        /* 0.7 */  "0.1034,  0.2173,  0.6452");
    }
    fall_transition(delay_template_3x3) {
      index_1 ("0.1, 0.3, 0.7"); /* Input transition */
      index_2 ("0.16, 0.35, 1.43"); /* Output capacitance */
      values ( /*    0.16      0.35     1.43 */ \
        /* 0.1 */   "0.0817,  0.1937,  0.7280", \
        /* 0.3 */   "0.1018,  0.2327,  0.7676", \
        /* 0.7 */   "0.1334,  0.2973,  0.8452");
    }
```

Source : Static Timing Analysis For Nanometer Designs: A Practical Approach by J. Bhasker and Rakesh Chadha

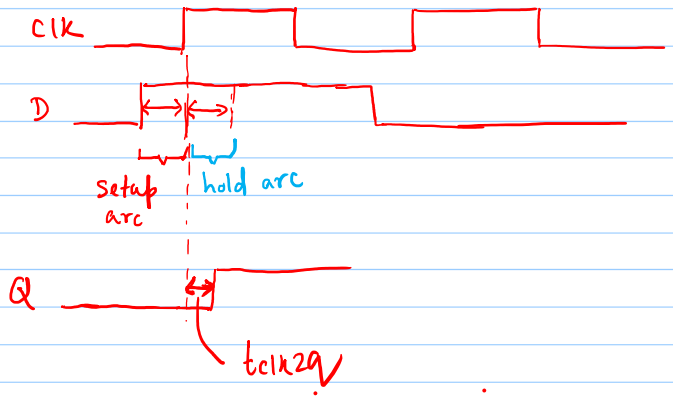
Timing Arcs of Sequential Circuits

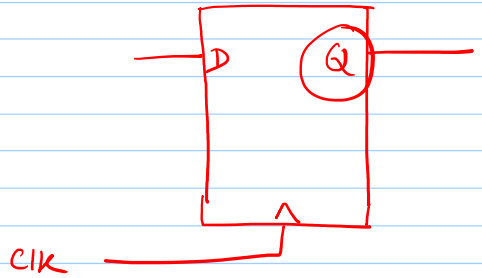
(1) For Synchronous inputs
   (i) setup check Arc (Rising and falling)
   (ii) Hold check Arc (Rising and falling)

(2) For asynchronous inputs (Reset)
   (i) Recovery check arc
   (ii) Removal check arc



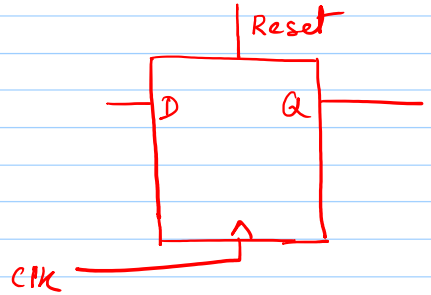
(3) For Synchronous outputs (Q, and $\bar{Q}$)
   (i) Clock-to-output propagation delay (rising and falling)

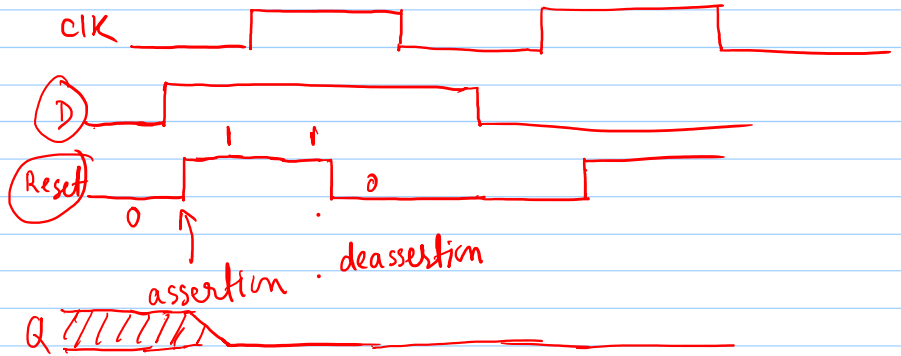

CLK
D
Setup arc
hold arc
Q
tclk2q


D
Q
CLK

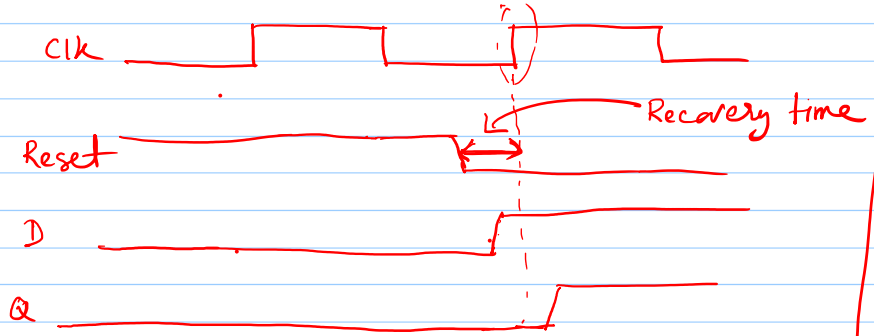
Reset

Reset = 1 $\Rightarrow$ Q = 0

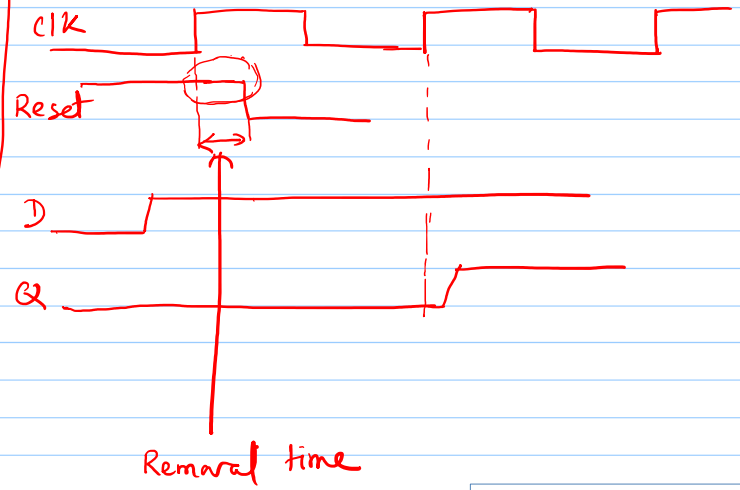




Recovery check Arc



Removal check Arc



```
pin (D) {
  direction : input;
  . . .
  timing () {
    related_pin : "CK";
    timing_type : "setup_rising";
    rise_constraint ("setuphold_template_3x3") {
      index_1("0.4, 0.57, 0.84"); /* Data transition */
      index_2("0.4, 0.57, 0.84"); /* Clock transition */
      values( /*    0.4      0.57     0.84 */ \
        /* 0.4 */   "0.063,  0.093,   0.112", \
        /* 0.57 */  "0.526,  0.644,   0.824", \
        /* 0.84 */  "0.720,  0.839,   0.930");
```

Source : Static Timing Analysis For Nanometer Designs: A Practical Approach by J. Bhasker and Rakesh Chadha

Thank You