بهینه‌سازی سرعت

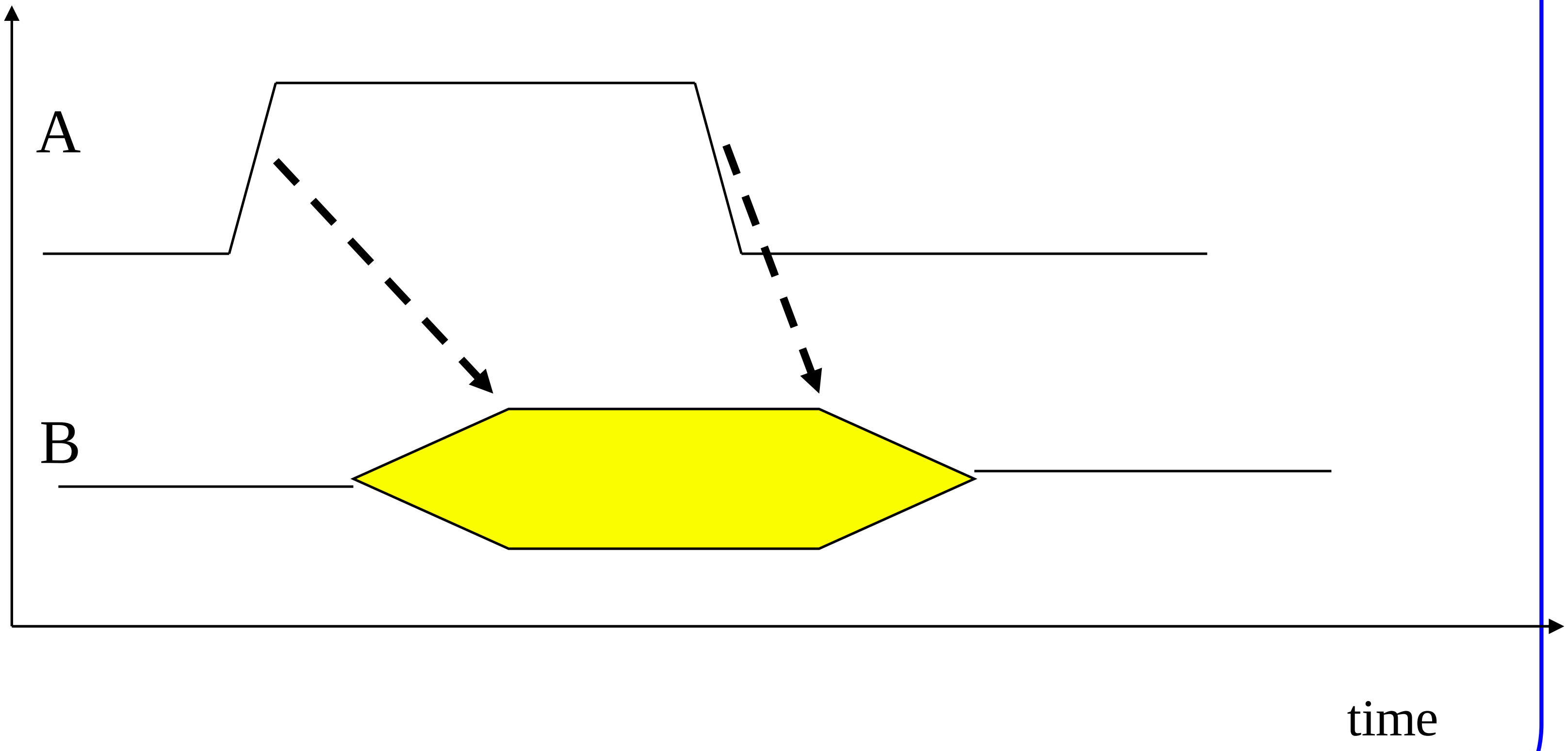
معیارهای بهینه‌سازی

- تحلیل مدارهای ترکیبی
- تحلیل مدارهای ترتیبی

Delay characteristics

- Measured from change in inputs to change in outputs.
- Data-dependent:
  - Some block delays depend on the value/waveform at the input ($t_{pHL} \neq t_{pLH}$) ($t_r \neq t_f$)
- May need to observe different paths through the network.

Timing diagram

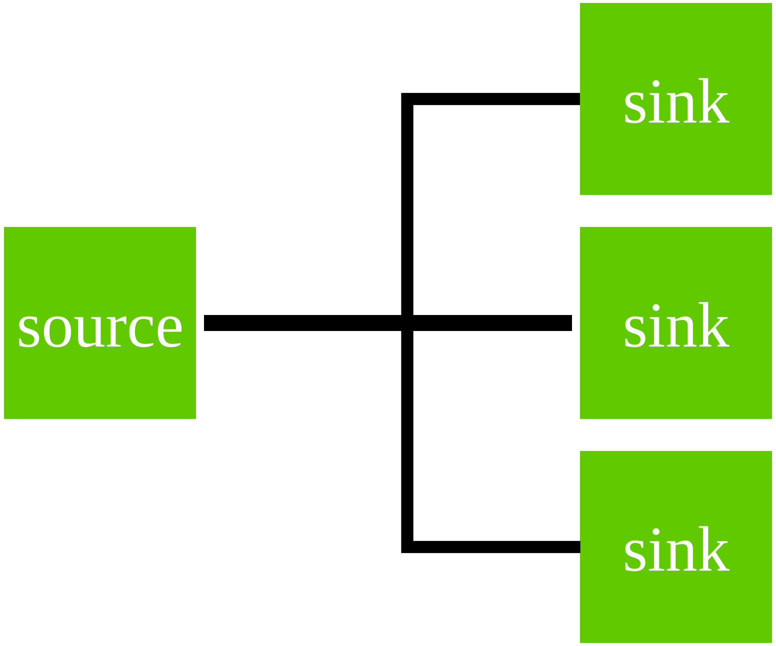


Sources of Delay

- **Gate delay:**
  - ❑ Little we can do about it
    - E.g. select another FPGA with faster logic blocks (LBs) or
    - Minimize the number of LBs in the critical path
- **Wire delay:**
  - ❑ Much we can do
    - E.g. select the proper path of the wire or
    - Select buffered paths.

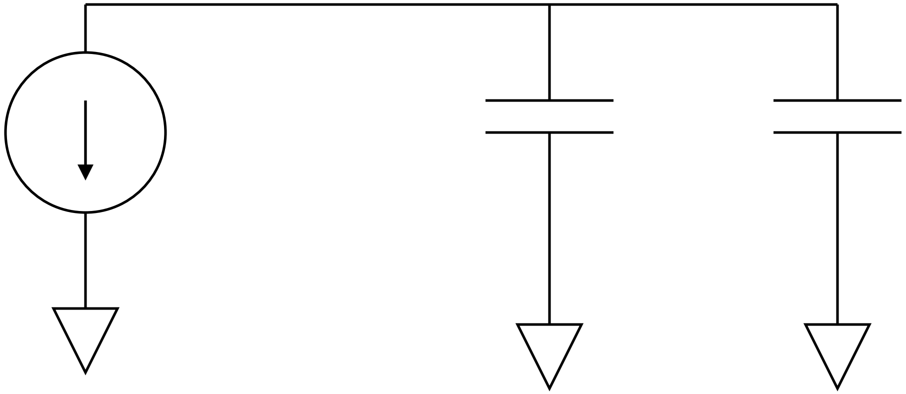
Fanout

- ❑ Fanout adds capacitance.



Driving fanout

- Adding gates adds capacitance:



Path delay

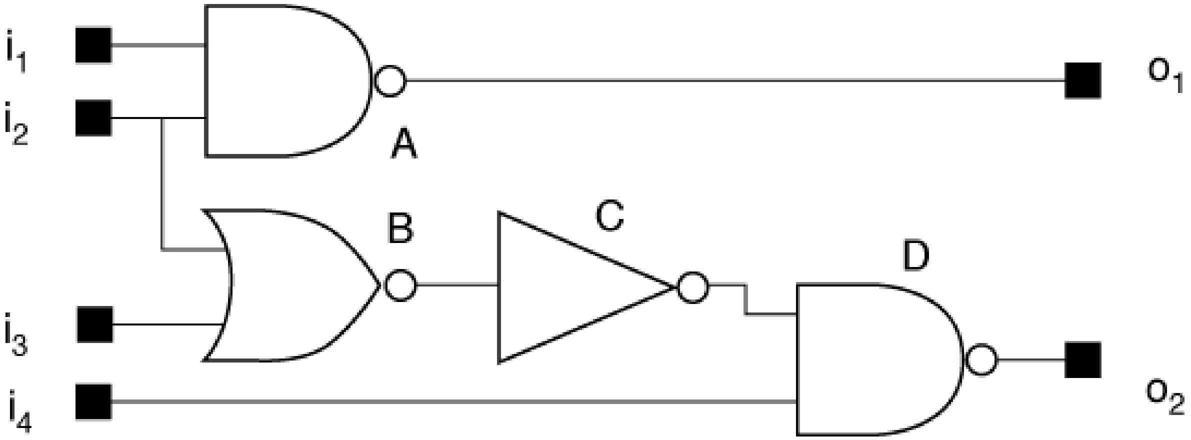
- **Combinational network delay:**
    - Measured over paths through network.
    - Can trace a causality chain from inputs to worst-case output.

Delay model

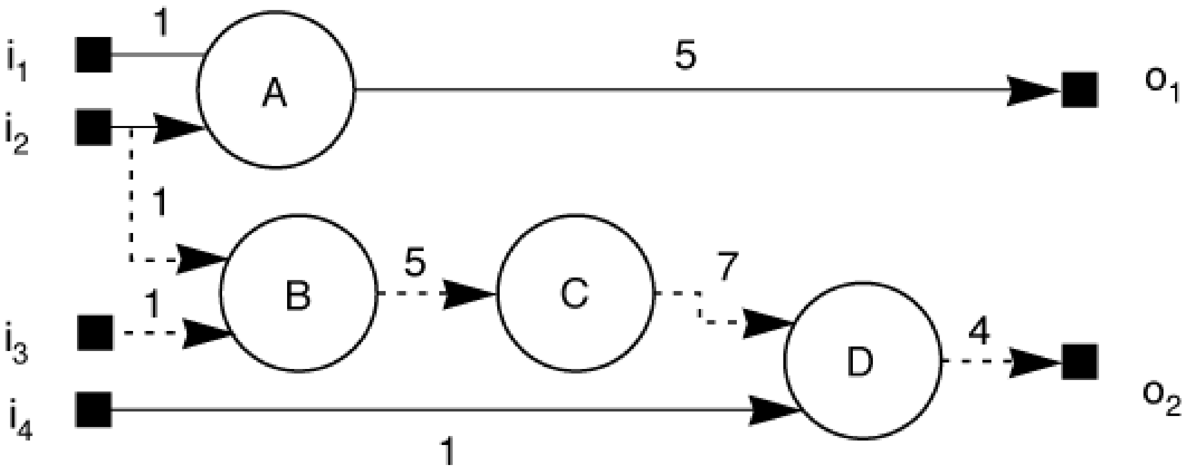
- ❑ Nodes represent gates.
- ❑ Assign delays to edges:
  - Signal may have different delay to different sinks.
- ❑ Lump gate and wire delay into a single value.

Path delay example

network



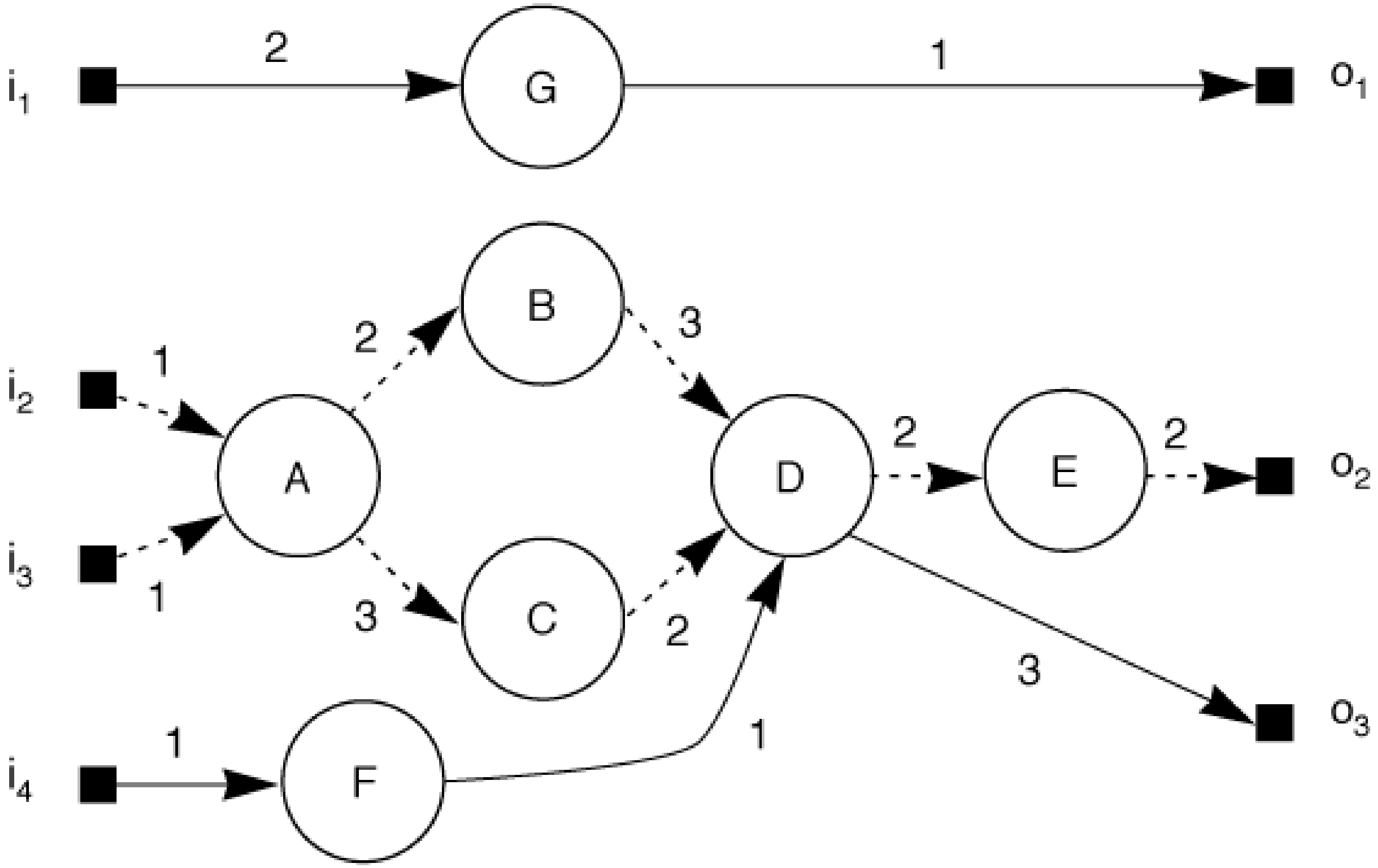
graph model



Critical path

- ❑ Critical path = path which creates longest delay.
- ❑ Can trace transitions which cause delays that are elements of the critical delay path.

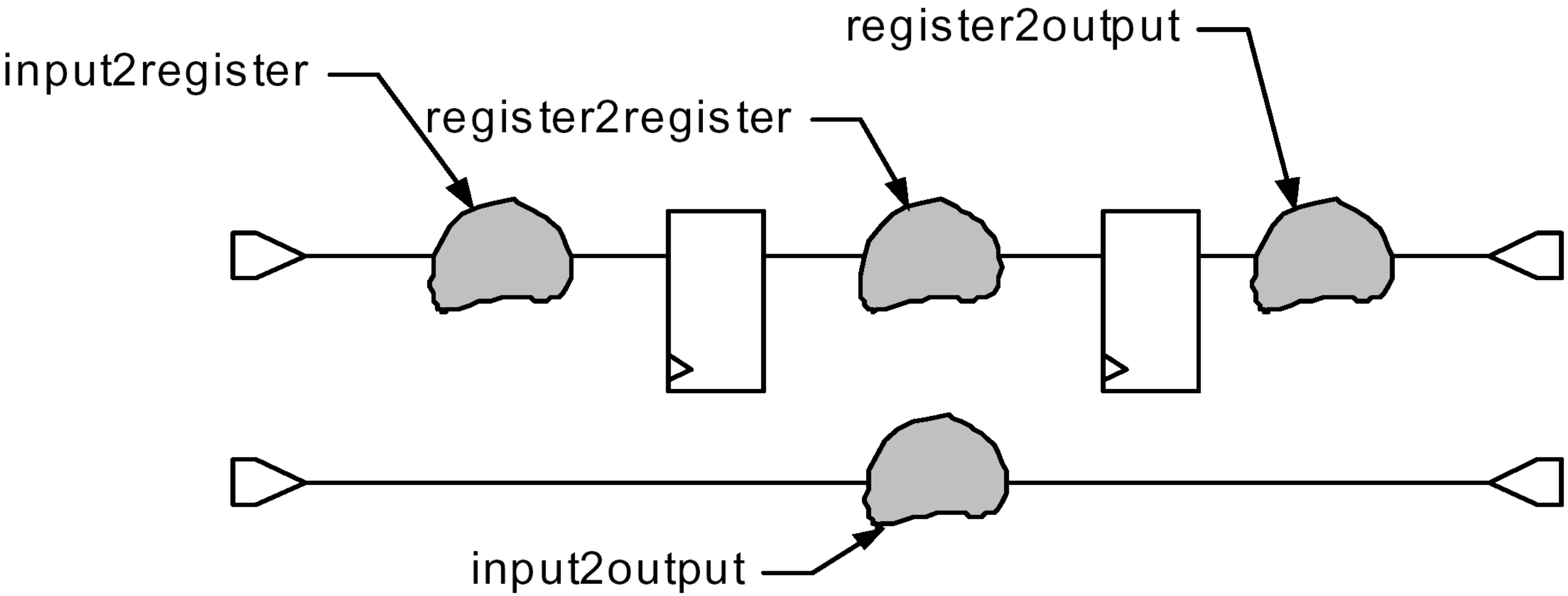
Critical path through delay graph



Reducing critical path length

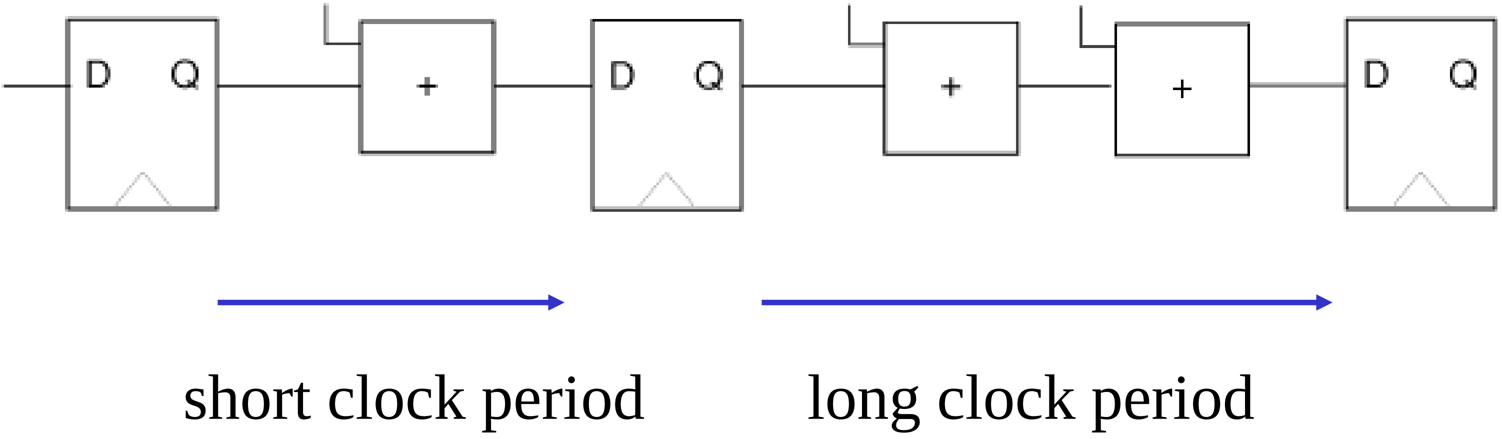
- **Must speed up the critical path**
  - ❑ Reducing delay off the path doesn't help.
- **There may be more than one path of the same delay.**
  - ❑ → Must speed up all equivalent paths to speed up circuit.
- **Cutset:**
  - ❑ A set of edges that when removed, break the graph into two unconnected paths. (e.g. {(C,D), (B,D)} or {(D,E})
    - – → Must speed up cutset through critical path.

Delay Paths in a design



Unbalanced delays

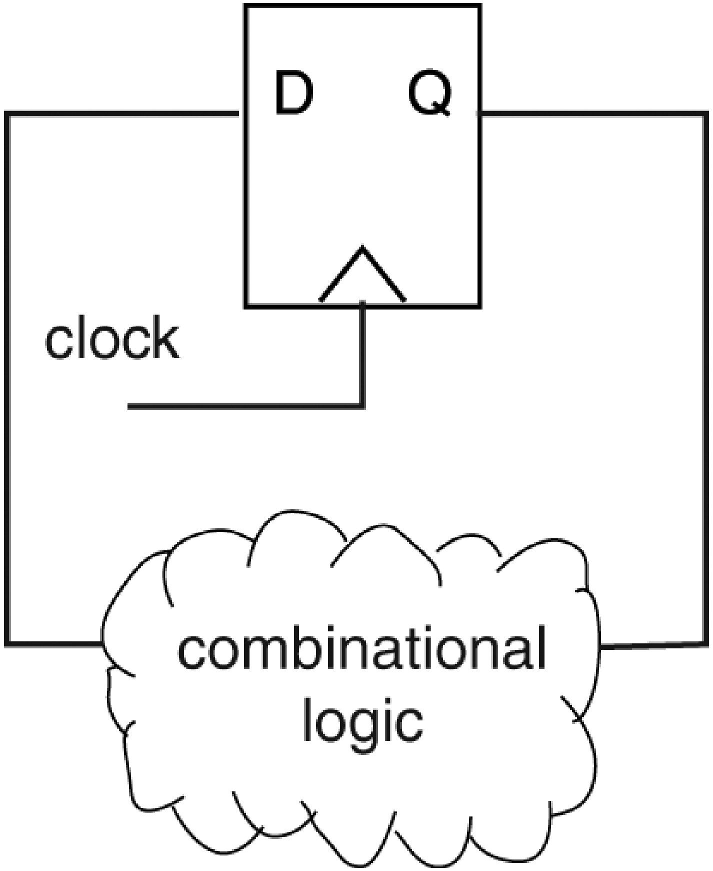
- Logic with unbalanced delays leads to inefficient use of logic:



short clock period

long clock period

Flip-flop-based system performance analysis

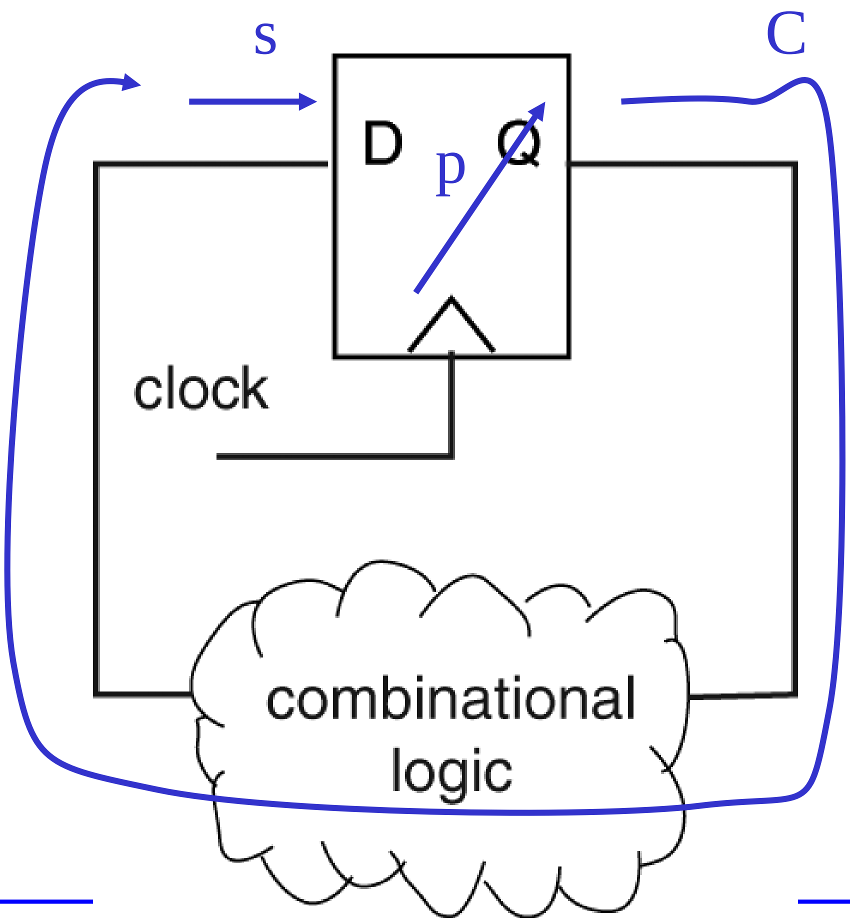


Flip-flop-based system model

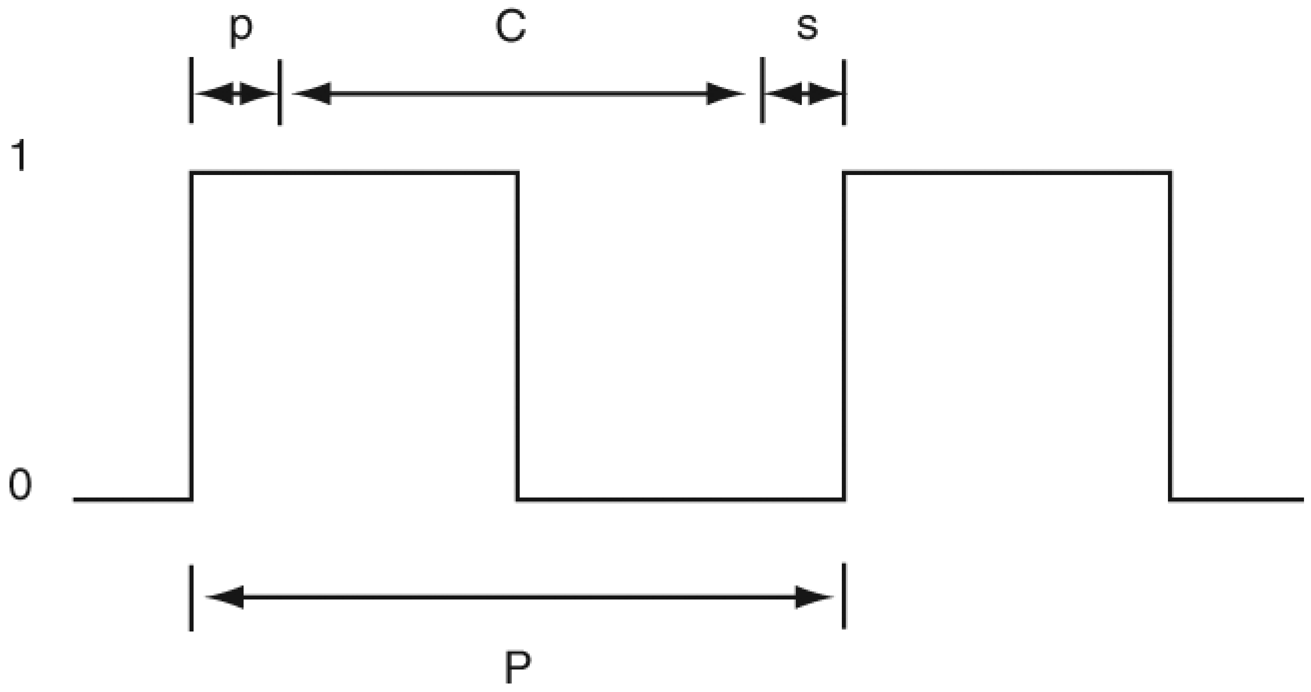
- **Assumptions:**
  - Clock signal is perfect (no rise/fall), period P
  - Clock event on rising edge
  - Setup time: s
    - Time required for the FF input to be stable before active edge of clock
  - Propagation time p
    - Time for value to go from input to output (t_{co})
  - Worst-case combinational delay C
    - Time from output of FF to input of this/other FFs

Clock period constraint

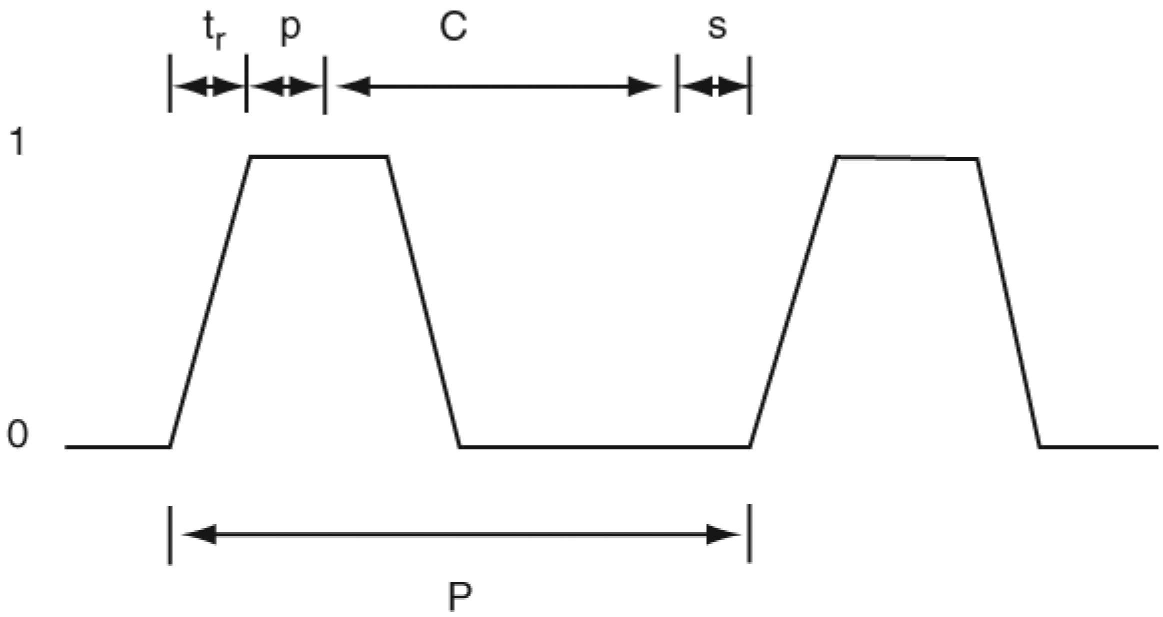
P >= p + C + s.



Clock parameters



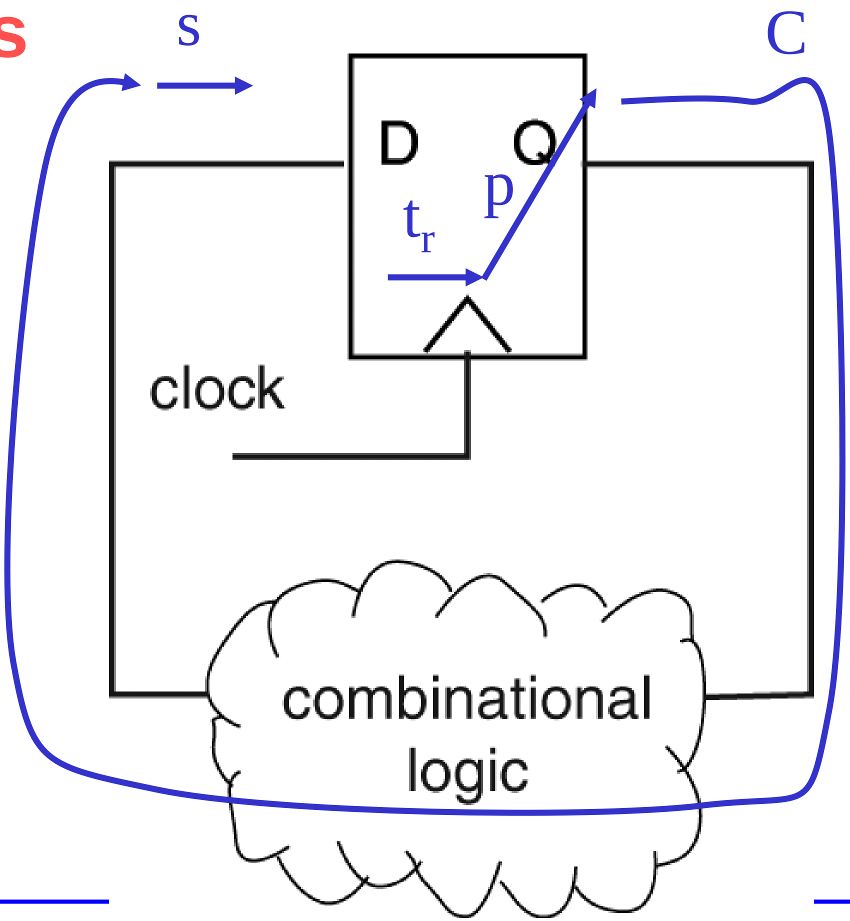
Clock with rise/fall



- t_r **is large because the clock wire is long and has high capacitance.**

Rise/fall clock period constraint

$P >= t_r + p + C + s$

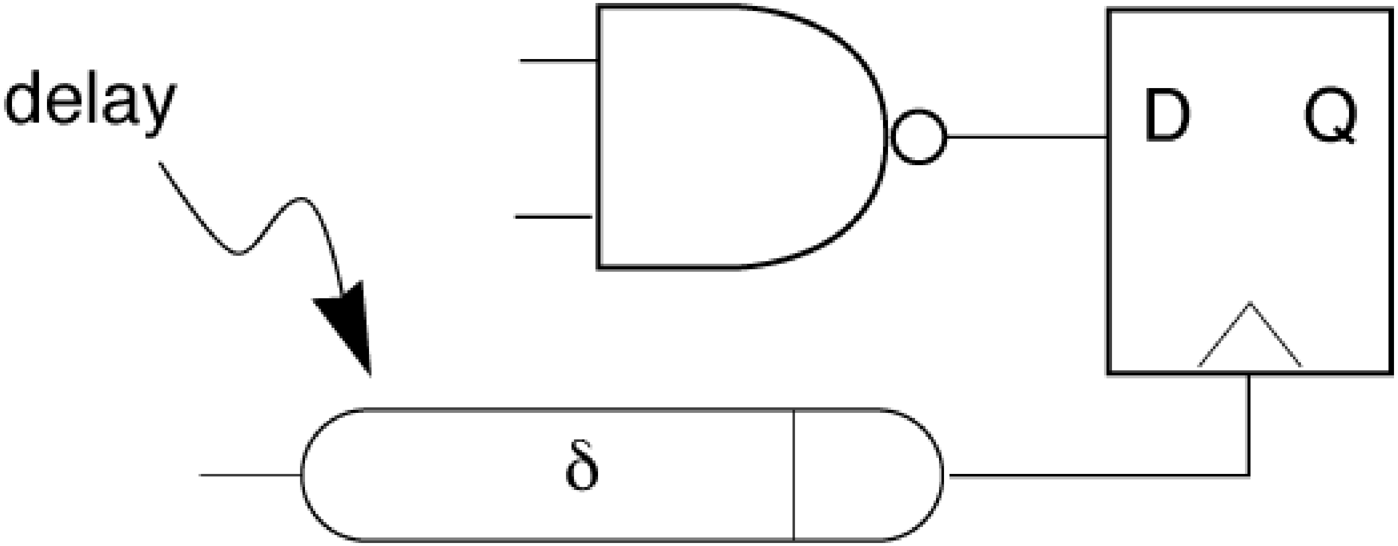


Skew

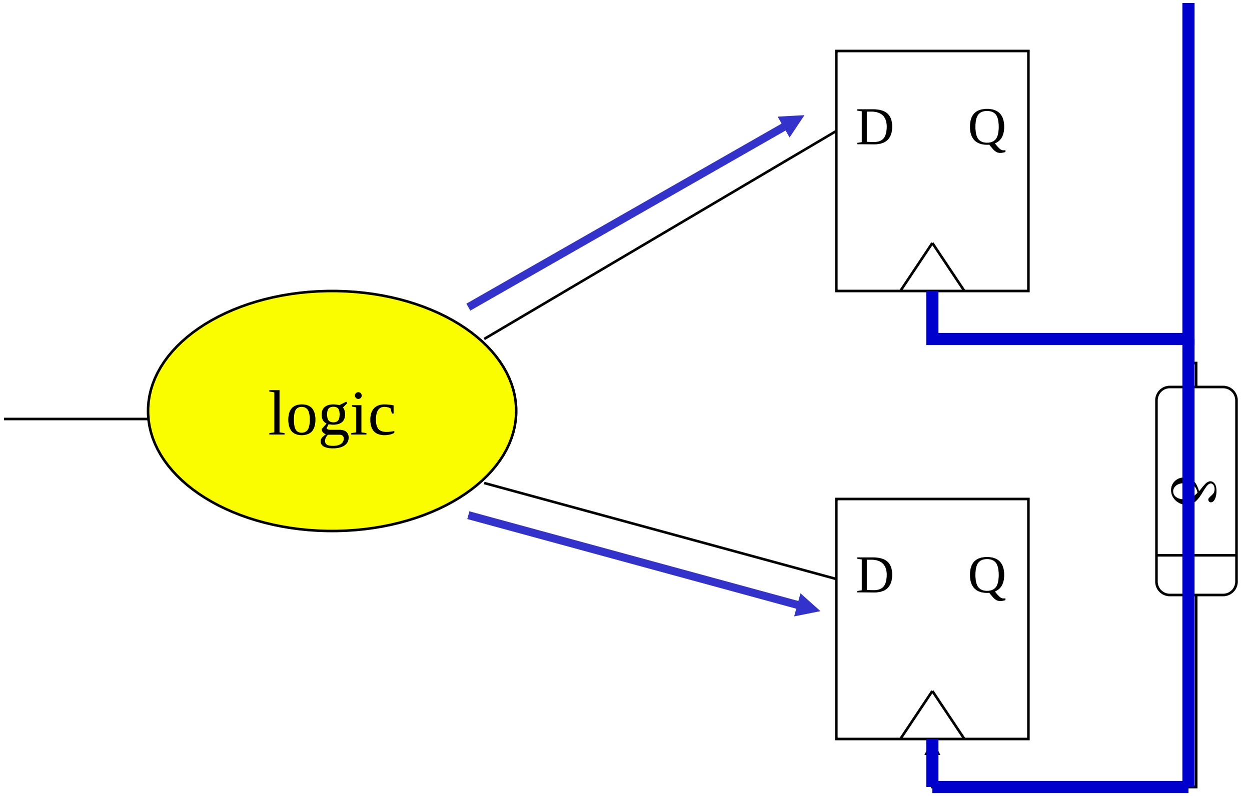
- **Skew**
  - relative delay between events
- **Clock skew**
  - can harm any sequential system.

Clock skew

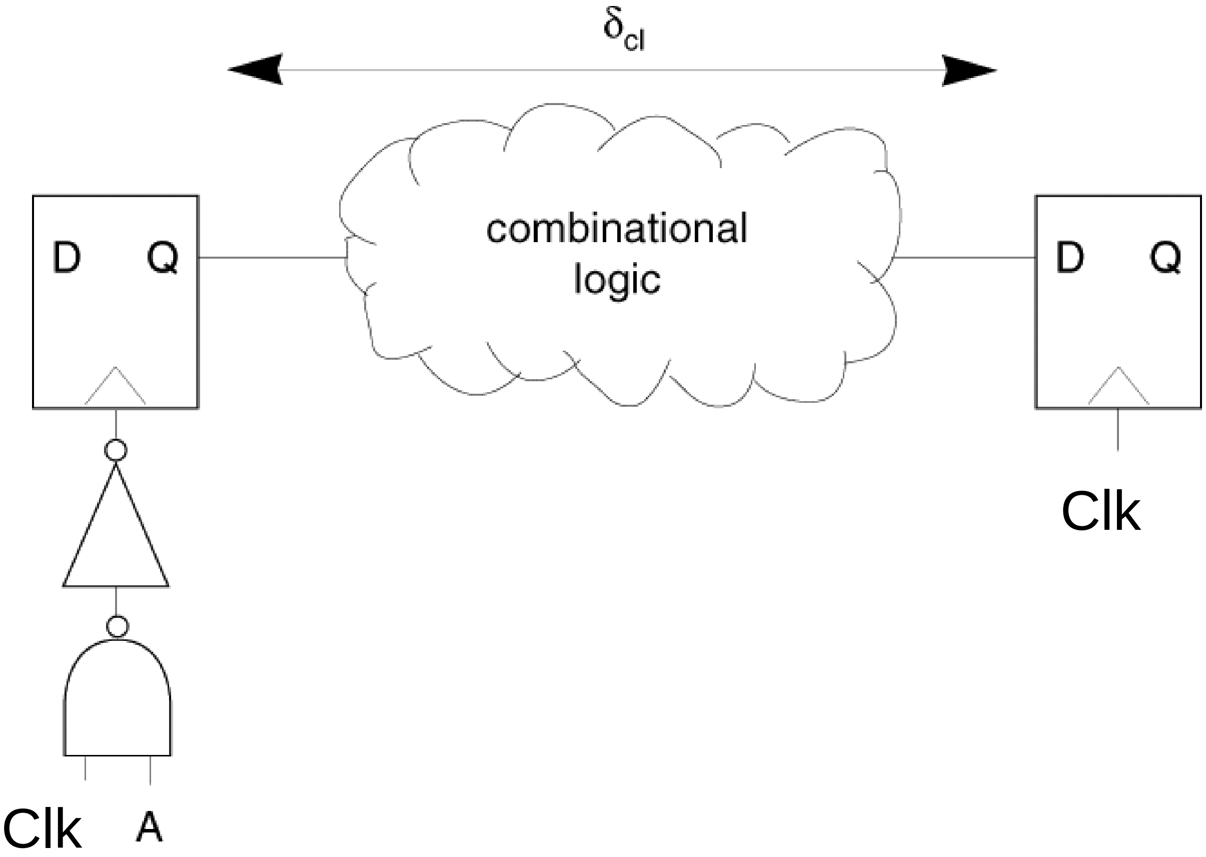
Clock must arrive at all memory elements in time to load data.



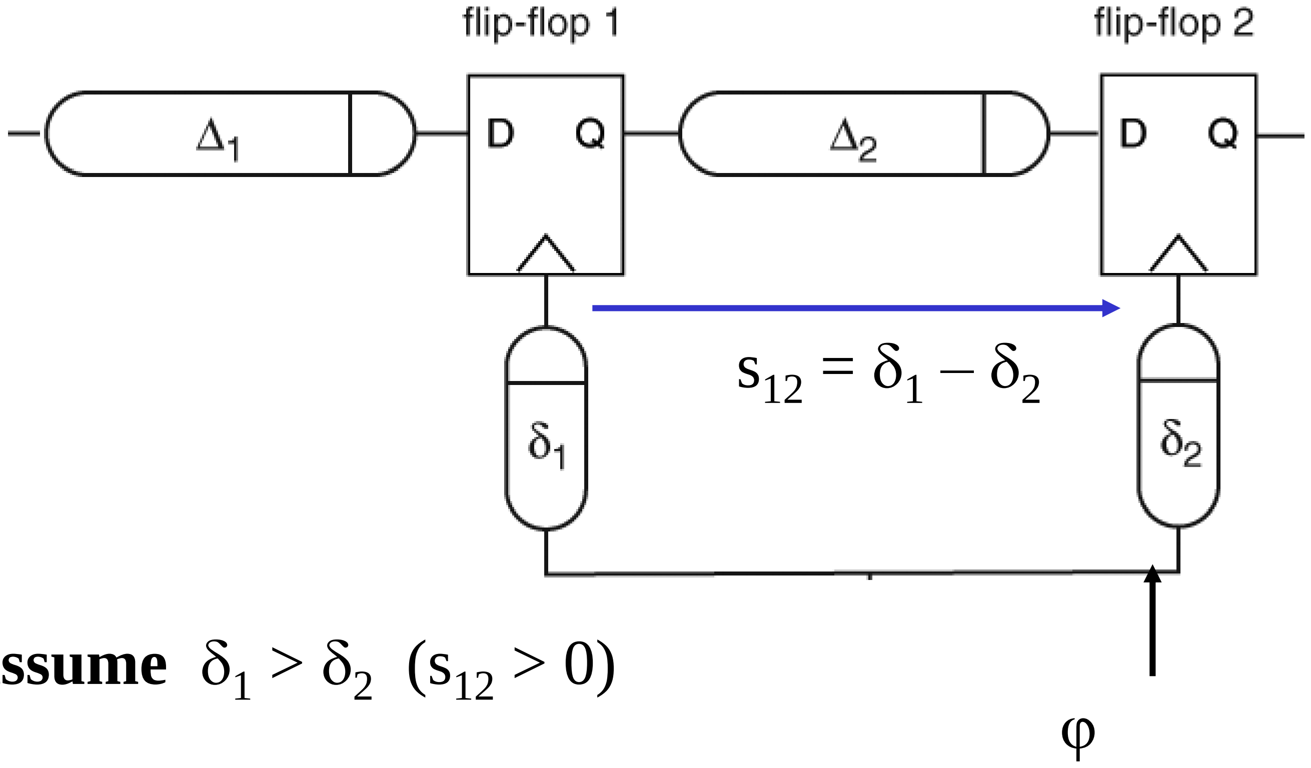
Clock skew in system



Clock skew and gated clocks



Clock skew analysis model



- **Assume** $\delta_1 > \delta_2$ $(s_{12} > 0)$

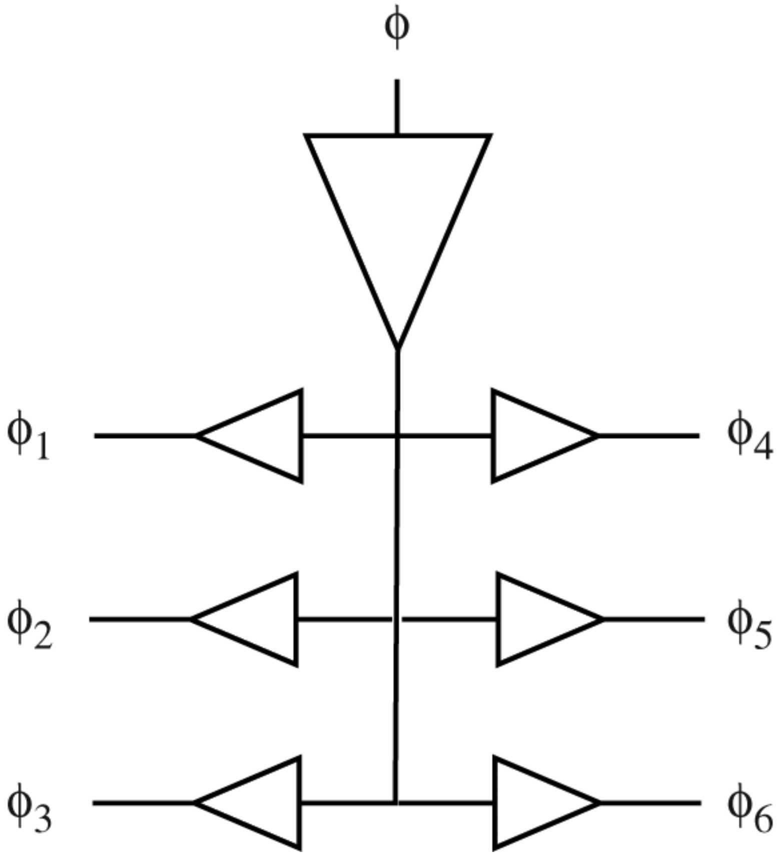
Skew and clock period

- Assume: each flip-flop operates instantaneously
- If clock arrives at FF1 after FF2, then there is less time for the signal to propagate through the combinational logic.
- Determine clock period:

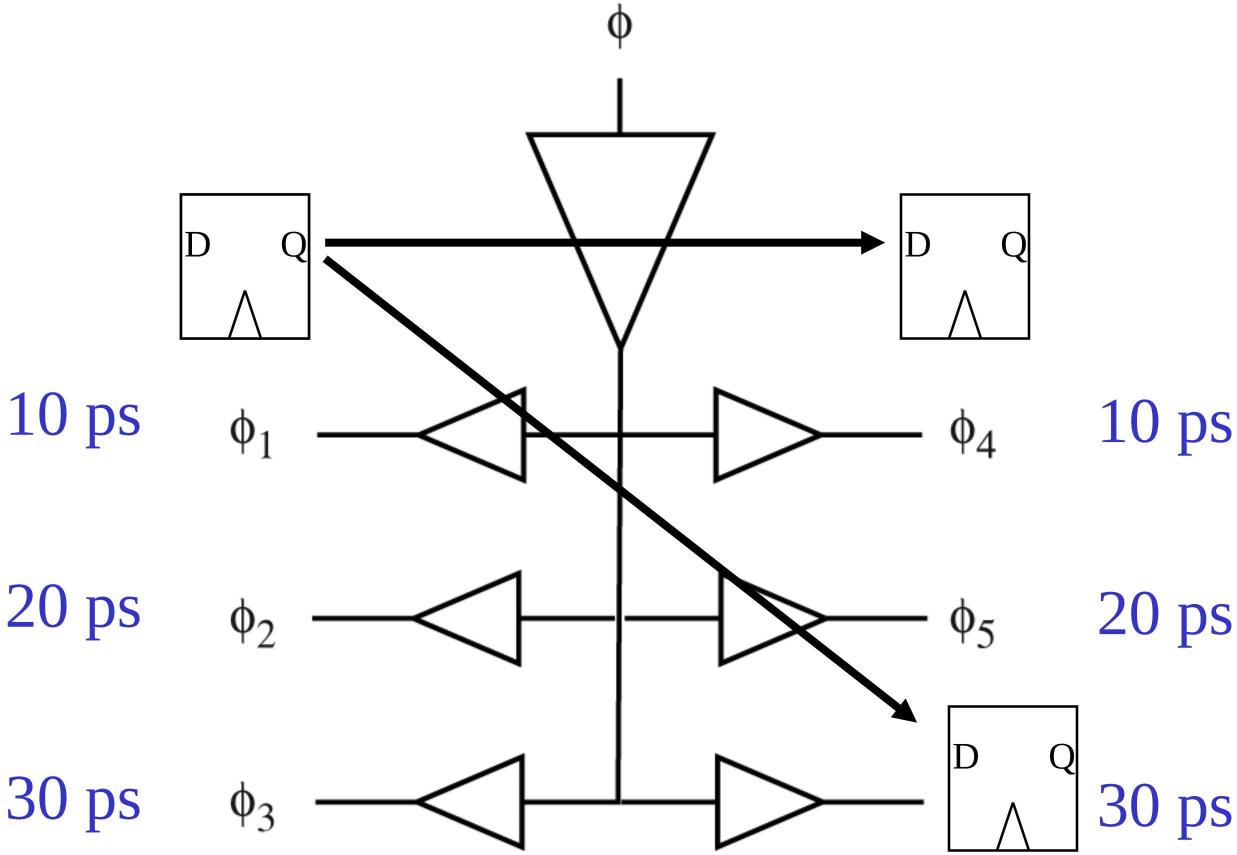
$$P >= \Delta_2 + s_{12}$$

Clock distribution

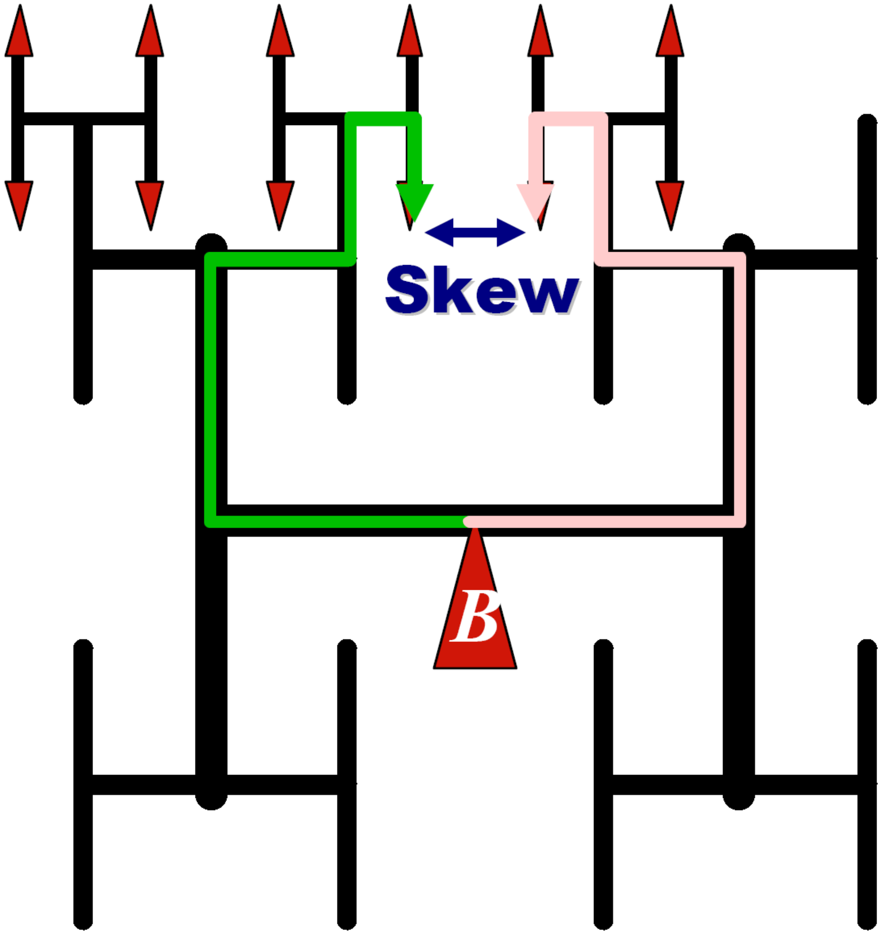
- **Clock design:**
  - ❑ Fast edges
  - ❑ Minimum skew



Clock skew example



Clock H-Tree



False paths

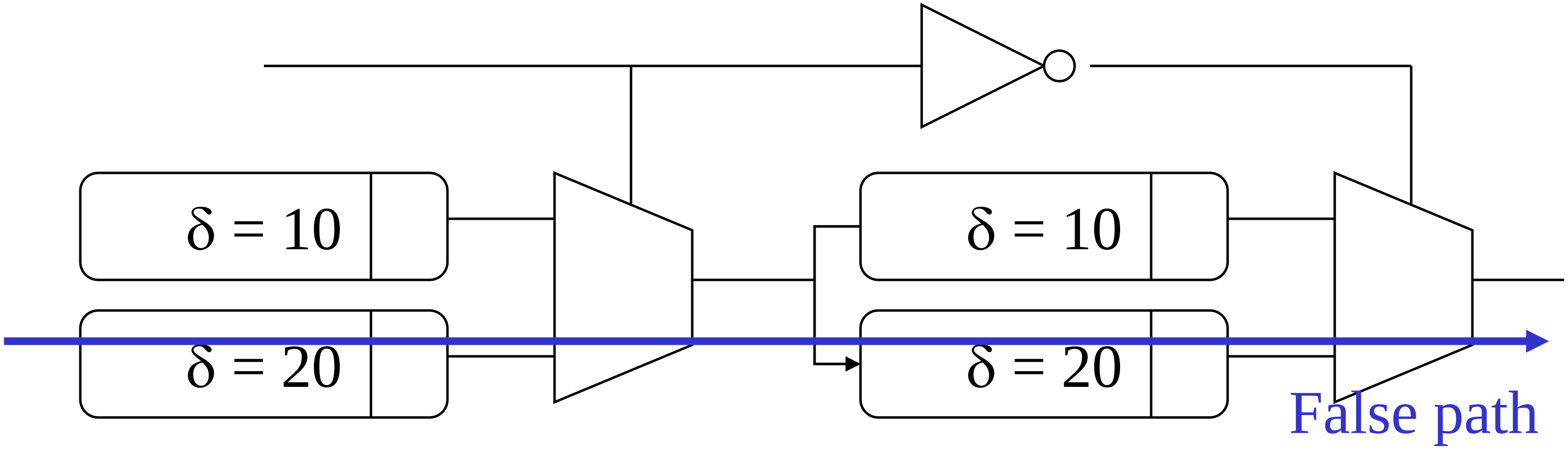
❑ Some input changes don’t cause output changes.

- **False path:**

❑ A path which never happens due to Boolean gate conditions.

❑ Cause pessimistic delay estimates.

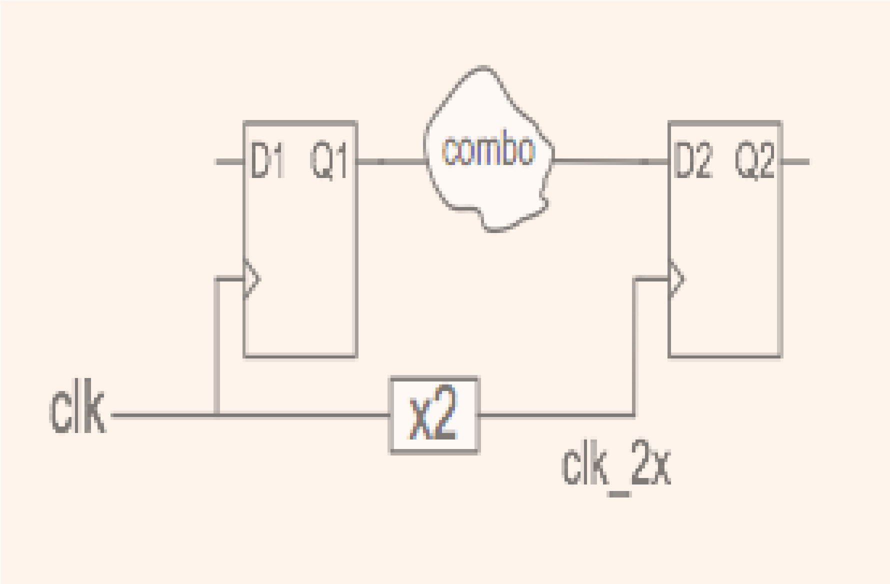
False path example



Multi-Cycle paths

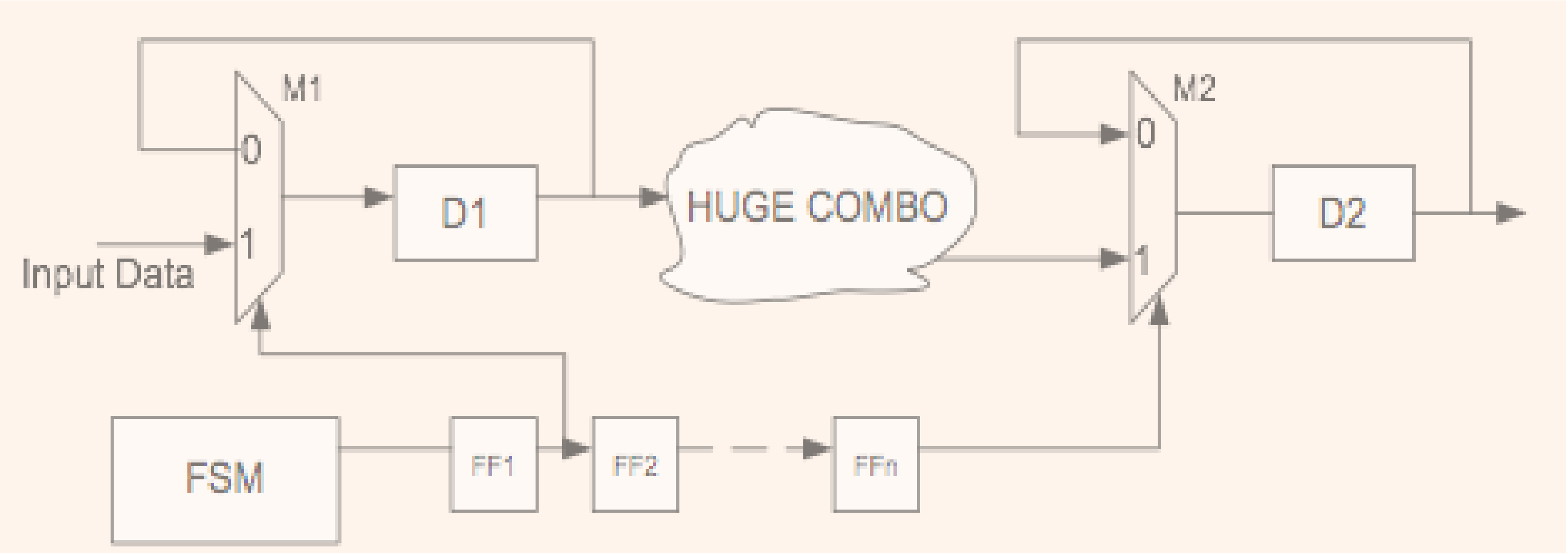
- ❑ Some combinational paths from a FF to another have more time than a clock cycle.

- **Example:**
  - ❑ Two-clock system



Multi-Cycle paths

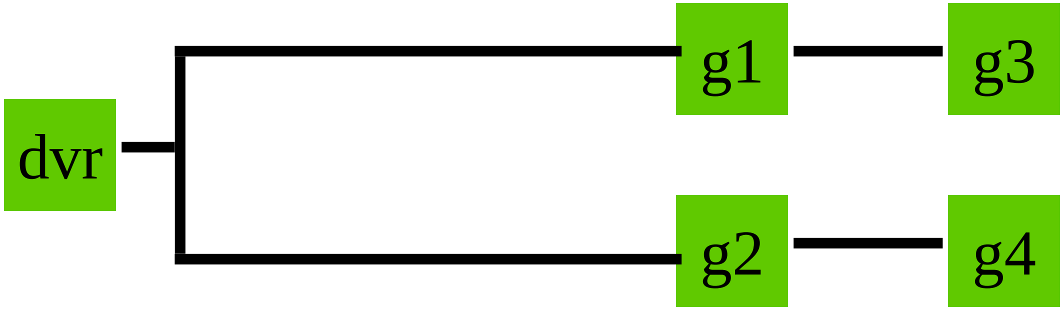
- **Example:**
  - ❑ Single-clock system

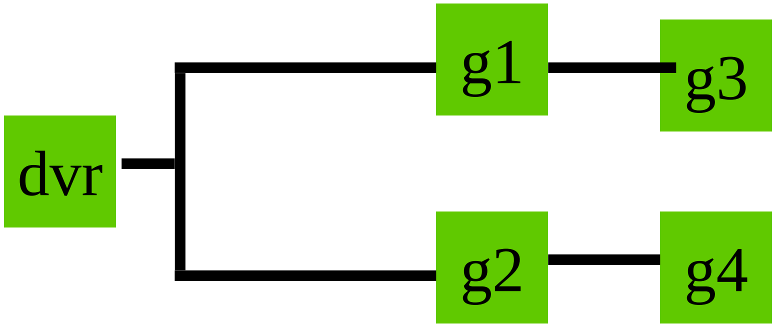


Placement and delay

- ❑ Placement helps determine gate distances.
- ❑ Gate distances determine routing.
- ❑ Routing determines wire length.
- ❑ Wire length determines capacitive load.
- ❑ Capacitive load determines delay.

Placement and wire capacitance





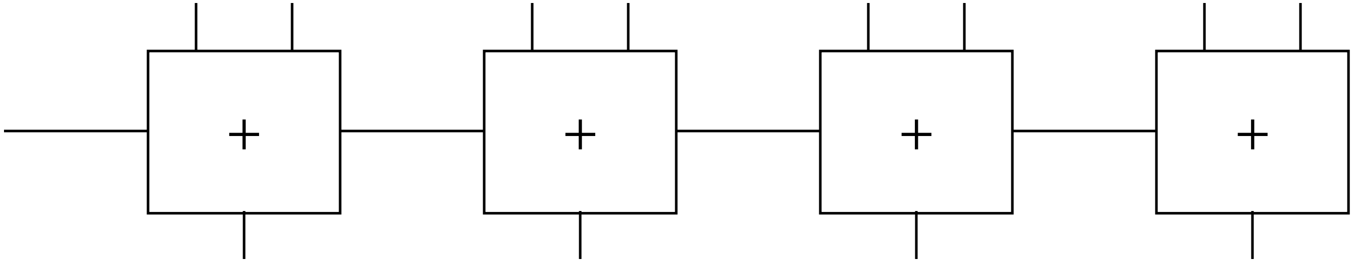
Optimizing network delay

- ❑ Identify the longest path(s).
- ❑ Improve delay along the longest path(s):
  - Driver delay
  - Wire delay
  - Logic restructuring

```
NET "data_out_1_OBUF"
ROUTE="{3;1;6slx25csg324;477afbc1!-1;8040;6064;S!0;-845;-504!1;0;344!1;" "-9743;1431!2;845;144;L!3;-16261;1!5;-22484;-4!6;-
17991;3!7;-12477;5681!8;"
"0;12800!9;0;12800!10;0;12800!11;0;13872!12;0;12800!13;0;12800!14;0;12800!"
"15;0;13872!16;305;7589!17;0;3200!18;1855;1675!19;686;18!20;80;20!21;" "-
1490;2207!22;-1311;251;L!}";
NET "data_out_1" LOC=D6;
INST "data_out_1_OBUF" LOC=SLICE_X29Y41;
INST "data_out_1" LOC=SLICE_X29Y41;
```

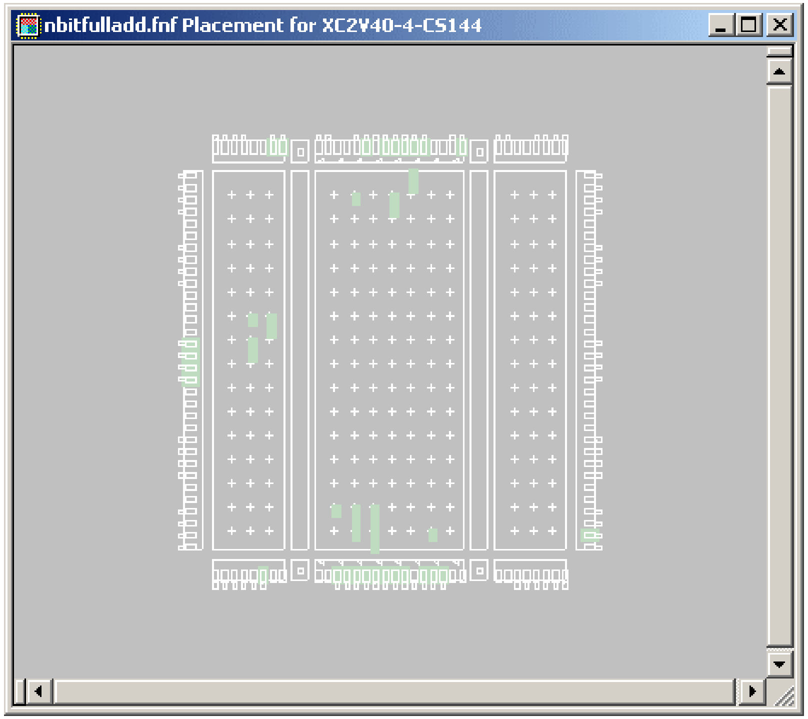
Example: Adder placement and delay

- N-bit adder: (optimal placement)

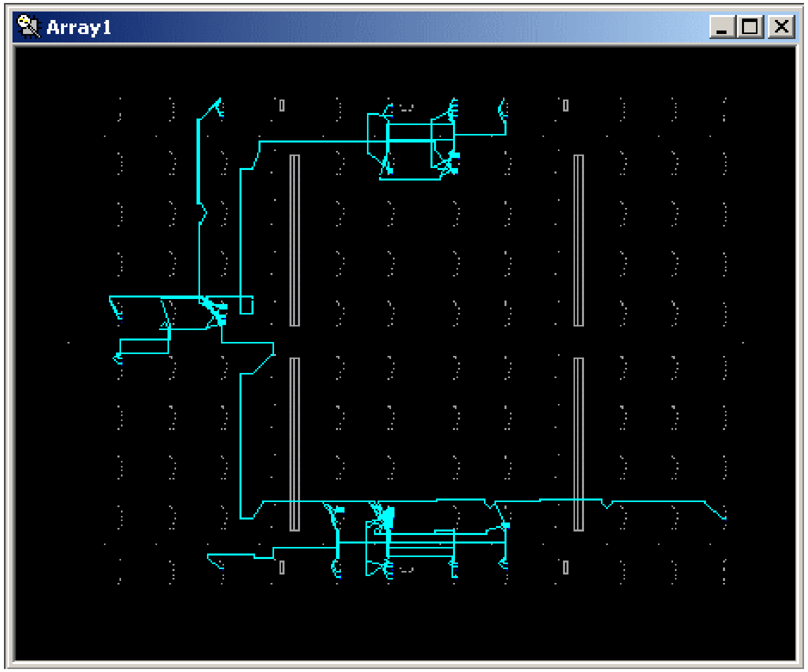


Bad placement and routing

With no delay constraints.



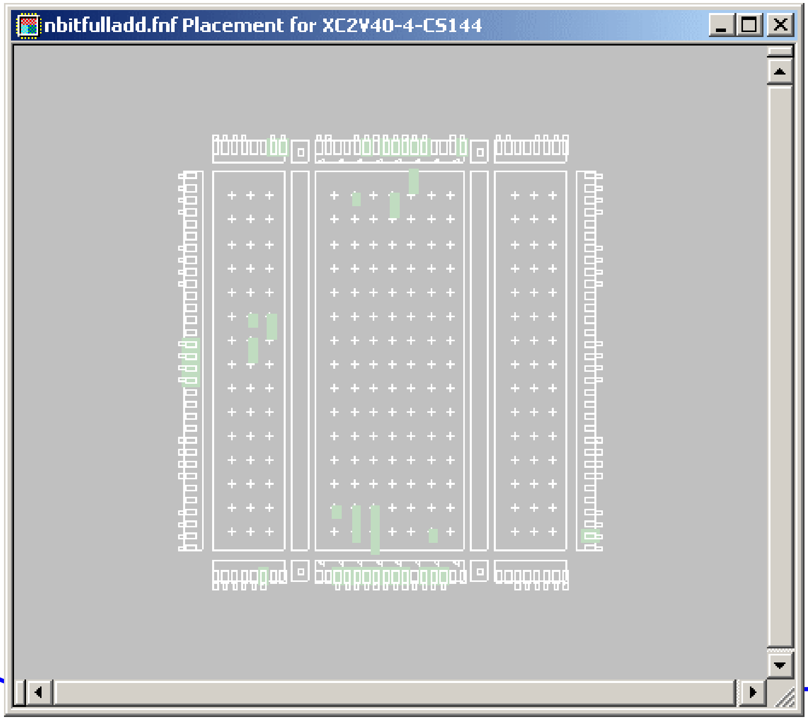
placement

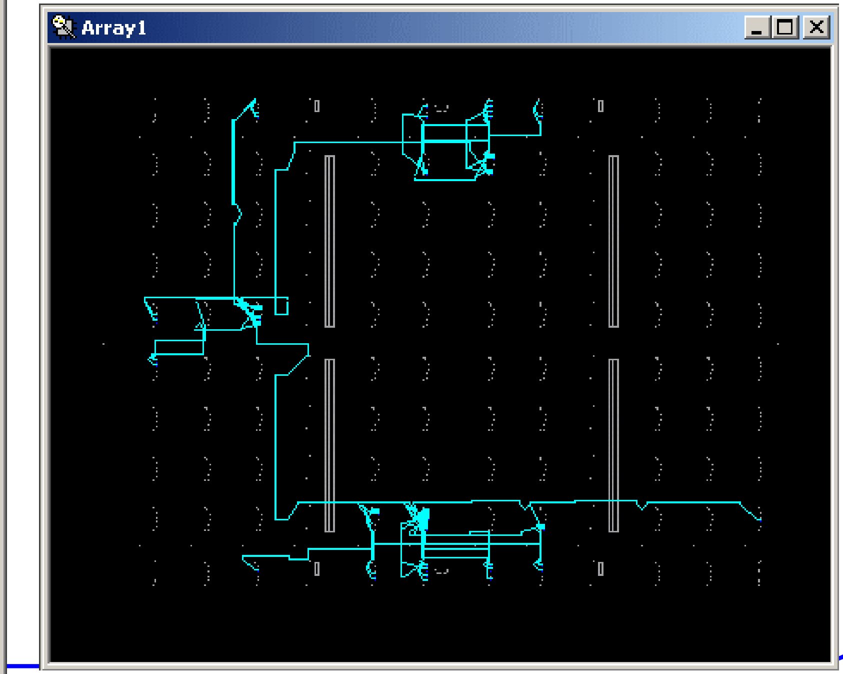


routing

Bad placement and routing

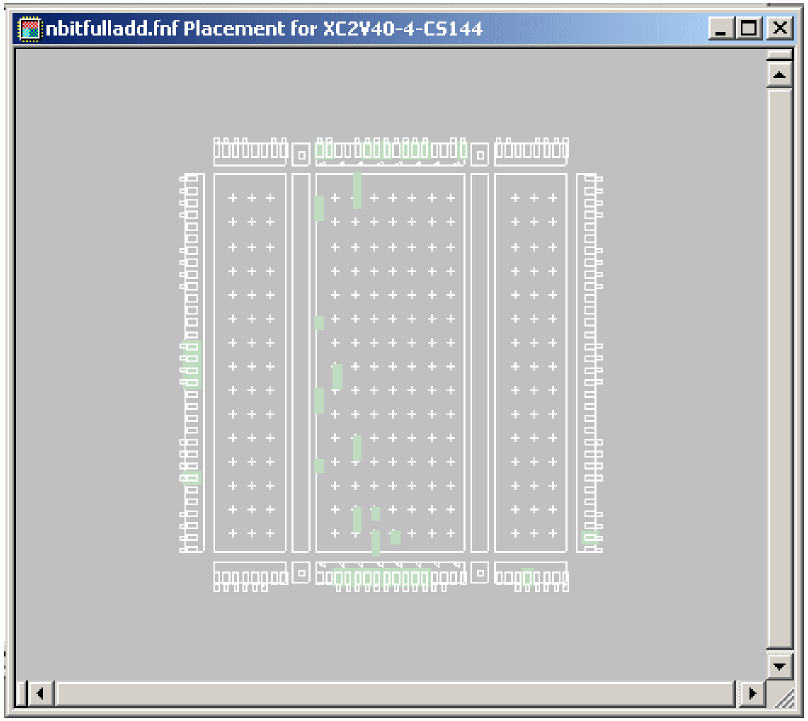
- Adder has been distributed throughout the FPGA.
- I/O pins have been spread around the chip.
- → P&R algorithms do not catch on regularity.

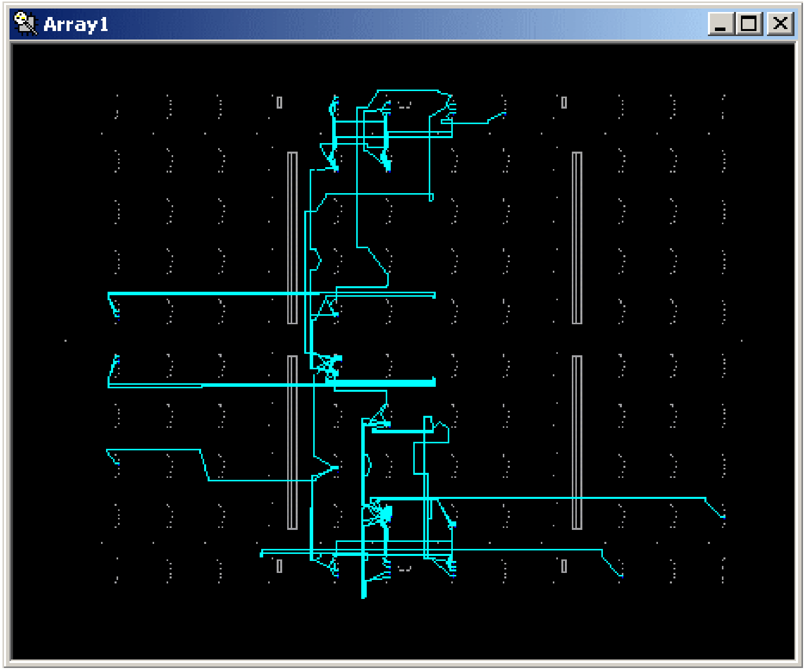




Better placement and routing

With delay constraints.





- Better but far from optimal (less spread out horizontally but spread out vertically)

How to improve?

- ❑ Use macros (optimized)
- ❑ Use proper constraints
  - – Objective
  - – Proper pin assignment
  - – Floorplan
    - – Put constraints on the placement of objects
  - – Hand place objects

Design experiments

- **Synthesize with no constraints.**
- **Synthesize with timing constraint.**
  - ❑ Tighten timing constraint.
- **Synthesize with placement constraints.**
- **Power:**
  - ❑ Many tools don’t allow us to directly specify power consumption
  - ❑ Some tools allow us to specify power as an objective
  - ❑ May need to rewrite our HDL code for better power consumption.

Commercial Tools

- **XST “-power” option**
  - ❑ reduces dynamic power consumption.
- **Xilinx MAP and PAR“-power” option**
  - ❑ reduces dynamic power
    - – But increases runtime and decreases design performance.
- **Quartus-II**
  - ❑ has Power-Driven Synthesis and Place & Route.

Mapping report

- **Case study:**
  - 16 x 16 multiplier
    - Combinational circuit
  - ❑ No timing or area constraints

```
Design Summary
--------------
Number of errors:      0
Number of warnings:    0
Logic Utilization:
  Number of 4 input LUTs:            501 out of   1,024   48%
Logic Distribution:
  Number of occupied Slices:          255 out of     512   49%
  Number of Slices containing only related logic:    255 out of     255  100%
  Number of Slices containing unrelated logic:         0 out of     255    0%
Total Number 4 input LUTs:           501 out of   1,024   48%

  Number of bonded IOBs:              64 out of      92   69%

Total equivalent gate count for design:  3,006
Additional JTAG gate count for IOBs:  3,072
Peak Memory Usage:  64 MB
```

Static timing analysis report

```
Timing constraint: TS_P2P = MAXDELAY FROM TIMEGRP "PADS" TO
    TIMEGRP "PADS" 99.999 uS  ;

 20135312 items analyzed, 0 timing errors detected. (0 setup errors, 0
    hold errors)
 Maximum delay is  20.916ns.
--------------------------------------------------------------------------------
```

After Mapping: → estimated delays
(no information about interconnects)

Static timing report: delays along paths

```
Data Sheet report:
-----------------
All values displayed in nanoseconds (ns)

Pad to Pad
------------------+----------------------+------------+
Source Pad  |Destination Pad|  Delay  |
------------------+----------------------+------------+
x<0>          |p<0>              |    5.824|
x<0>          |p<10>            |   10.675|
x<0>          |p<11>            |   11.214|
x<0>          |p<12>            |   11.753|
```

Routing report

```
Phase 1: 1975 unrouted;       REAL time: 11 secs

Phase 2: 1975 unrouted;       REAL time: 11 secs

Phase 3: 619 unrouted;       REAL time: 12 secs

Phase 4: 619 unrouted; (0)      REAL time: 12 secs

Phase 5: 619 unrouted; (0)      REAL time: 12 secs

Phase 6: 619 unrouted; (0)      REAL time: 12 secs

Phase 7: 0 unrouted; (0)      REAL time: 12 secs

The NUMBER OF SIGNALS NOT COMPLETELY ROUTED for this design is: 0
```

- **REAL time: Routing algorithm run time.**

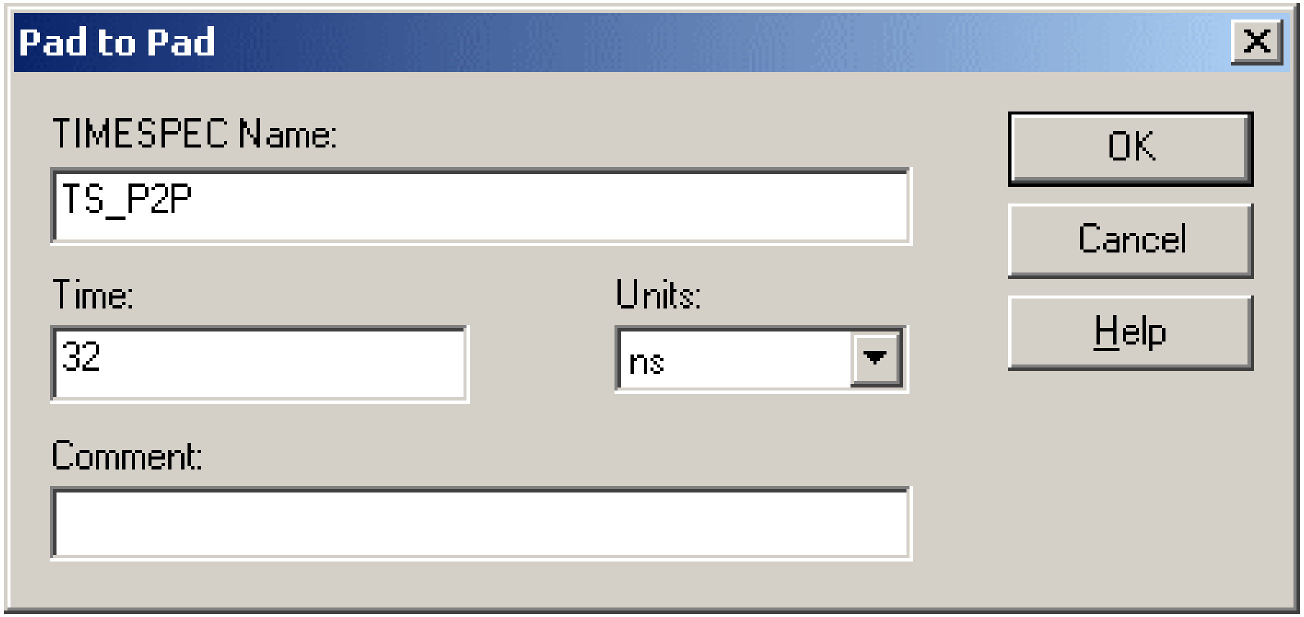
Static timing after routing

```
Timing constraint: TS_P2P = MAXDELAY FROM TIMEGRP
    "PADS" TO TIMEGRP "PADS" 99.999 uS  ;

 20135312 items analyzed, 0 timing errors detected. (0 setup
    errors, 0 hold errors)
 Maximum delay is  38.424ns.
--------------------------------------------------------------------------------
```

- **(vs. 20.916 ns in mapping report) Because of interconnect delays.**

Timing constraint

- Use timing constraint editor:



TIMESPEC Name: TS_P2P

Time: 32

Units: ns

Comment:

Post-map static timing report

Timing constraint: TS_P2P = MAXDELAY FROM TIMEGRP "PADS"
TO TIMEGRP "PADS" 32 nS ;

20135312 items analyzed, 0 timing errors detected. (0 setup errors, 0 hold errors)

Maximum delay is 20.916ns.

Pad to pad

Hasn't changed since this design has limited opportunities for logic synthesis to change delays by restructuring logic.

Post-routing static timing report

Timing constraint: TS_P2P = MAXDELAY FROM TIMEGRP "PADS"
→ TO TIMEGRP "PADS" 32 nS ;

20135312 items analyzed, 0 timing errors detected. (0 setup errors, 0 hold errors)

→ Maximum delay is 31.984ns.

Tools generally try to meet the delay goal as closely as possible to minimize area.

Tighter timing constraints

- Tighten requirement to 25 ns.
- Post-place-route timing report:

```
Timing constraint: TS_P2P = MAXDELAY FROM TIMEGRP
        "PADS" TO TIMEGRP "PADS" 25 nS  ;

 20135312 items analyzed, 11 timing errors detected. (11 setup
        errors, 0 hold errors)
 Maximum delay is  31.128ns.
```

Report on a violated path

```
Slack:                -6.128ns (requirement - data path)
  Source:                y<0> (PAD)
  Destination:             p<30> (PAD)
  Requirement:               25.000ns
  Data Path Delay:            31.128ns (Levels of Logic = 31)
```

Modify the logic and/or physical design to improve the delay.

Power report

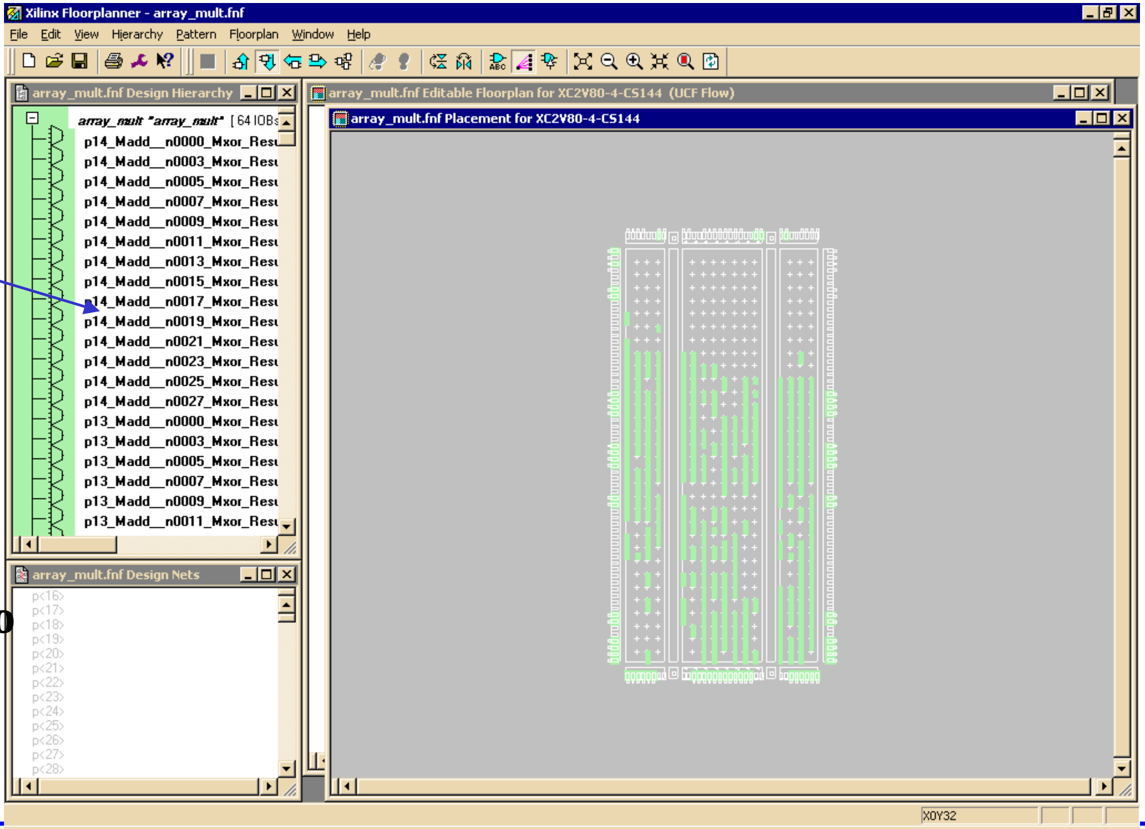
```
Power summary:                         I(mA)    P(mW)
----------------------------------------------------------------
Total estimated power consumption:                   333
                          ---
                  Vccint 1.50V:       0        0
                  Vccaux 3.30V:      100      330
                  Vcco33 3.30V:        1        3
                          ---
                      Inputs:       0        0
                       Logic:       0        0
                     Outputs:
                        Vcco33       0        0
                     Signals:       0        0
                          ---
         Quiescent Vccaux  3.30V:     100      330
         Quiescent Vcco33  3.30V:       1        3

Thermal summary:
----------------------------------------------------------------
  Estimated junction temperature:                 36C
                Ambient temp:  25C
                   Case temp:  35C
                 Theta J-A:  34C/W
```

Helps us determine whether we need additional cooling.

بهبود سرعت با جاسازی

- **Floorplanner window:**
  - ❑ Floorplanner → View/edit placed design

LEs


Xilinx Floorplanner - array_mult.fnf
File Edit View Hierarchy Pattern Floorplan Window Help
array_mult.fnf Design Hierarchy
array_mult.fnf Editable Floorplan for XC2V80-4-CS144 (UCF Flow)
array_mult.fnf Placement for XC2V80-4-CS144
array_mult.fnf Design Nets
X0Y32

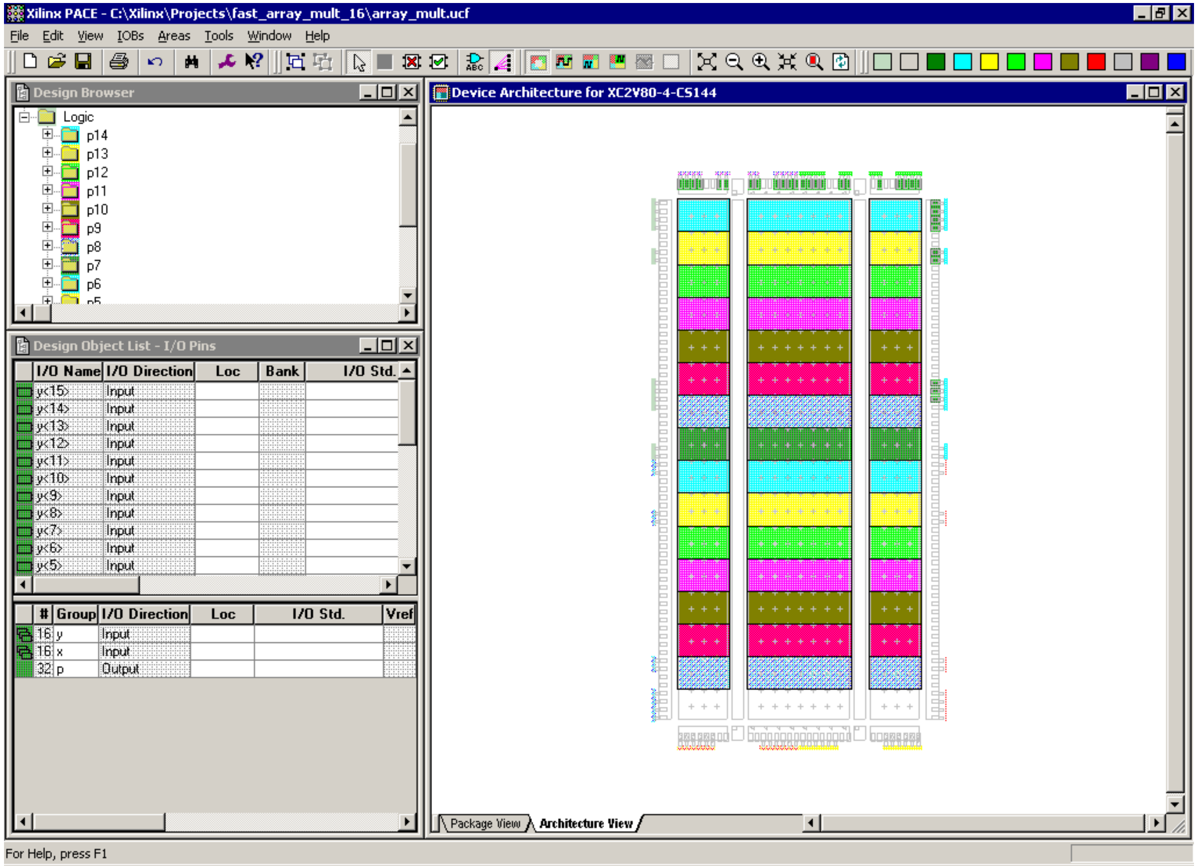
array_mult "array_mult" [64 IOBs
p14_Madd__n0000_Mxor_Resu
p14_Madd__n0003_Mxor_Resu
p14_Madd__n0005_Mxor_Resu
p14_Madd__n0007_Mxor_Resu
p14_Madd__n0009_Mxor_Resu
p14_Madd__n0011_Mxor_Resu
p14_Madd__n0013_Mxor_Resu
p14_Madd__n0015_Mxor_Resu
p14_Madd__n0017_Mxor_Resu
p14_Madd__n0019_Mxor_Resu
p14_Madd__n0021_Mxor_Resu
p14_Madd__n0023_Mxor_Resu
p14_Madd__n0025_Mxor_Resu
p14_Madd__n0027_Mxor_Resu
p13_Madd__n0000_Mxor_Resu
p13_Madd__n0003_Mxor_Resu
p13_Madd__n0005_Mxor_Resu
p13_Madd__n0007_Mxor_Resu
p13_Madd__n0009_Mxor_Resu
p13_Madd__n0011_Mxor_Resu

p<16>
p<17>
p<18>
p<19>
p<20>
p<21>
p<22>
p<23>
p<24>
p<25>
p<26>
p<27>
p<28>

- **Green rectangles: mapped components to CLBs**

جاسازی

- طرح منظم ← ساختار مداری منظم
  - ❑ در طرح مسطح، ابزار فاقد اطلاعات نظم طرح است

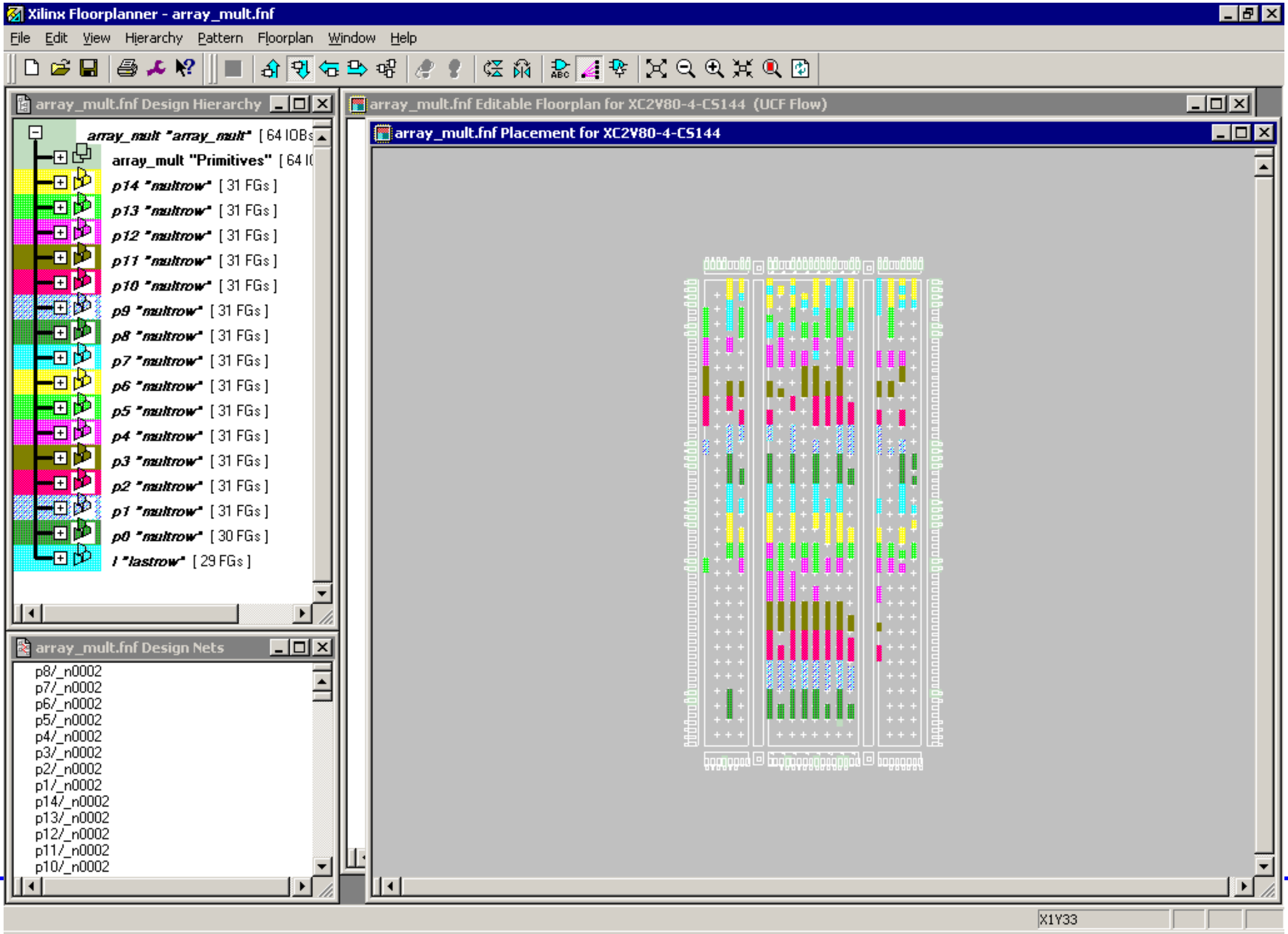

Xilinx PACE - C:\Xilinx\Projects\fast_array_mult_16\array_mult.ucf
File
Edit
View
IOBs
Areas
Tools
Window
Help
Design Browser
Logic
p14
p13
p12
p11
p10
p9
p8
p7
p6
p5
Design Object List - I/O Pins
Device Architecture for XC2V80-4-CS144
Package View
Architecture View
For Help, press F1

	I/O Name	I/O Direction	Loc	Bank	I/O Std.
	y<15>	Input			
	y<14>	Input			
	y<13>	Input			
	y<12>	Input			
	y<11>	Input			
	y<10>	Input			
	y<9>	Input			
	y<8>	Input			
	y<7>	Input			
	y<6>	Input			
	y<5>	Input			

	#	Group	I/O Direction	Loc	I/O Std.	Vref
	16	y	Input			
	16	x	Input			
	32	p	Output			

جاسازی

- نتیجهٔ جایابی


Xilinx Floorplanner - array_mult.fnf
File Edit View Hierarchy Pattern Floorplan Window Help
array_mult.fnf Design Hierarchy
array_mult.fnf Editable Floorplan for XC2V80-4-CS144 (UCF Flow)
array_mult.fnf Placement for XC2V80-4-CS144
array_mult.fnf Design Nets
X1Y33

array_mult "array_mult" [64 IOBs
array_mult "Primitives" [64 I(
p14 "multrow" [31 FGs]
p13 "multrow" [31 FGs]
p12 "multrow" [31 FGs]
p11 "multrow" [31 FGs]
p10 "multrow" [31 FGs]
p9 "multrow" [31 FGs]
p8 "multrow" [31 FGs]
p7 "multrow" [31 FGs]
p6 "multrow" [31 FGs]
p5 "multrow" [31 FGs]
p4 "multrow" [31 FGs]
p3 "multrow" [31 FGs]
p2 "multrow" [31 FGs]
p1 "multrow" [31 FGs]
p0 "multrow" [30 FGs]
l "lastrow" [29 FGs]

p8/_n0002
p7/_n0002
p6/_n0002
p5/_n0002
p4/_n0002
p3/_n0002
p2/_n0002
p1/_n0002
p14/_n0002
p13/_n0002
p12/_n0002
p11/_n0002
p10/_n0002

جاسازی

- گزارش ابزار جایابی و مسیریابی:

```
19742142 items analyzed, 0 timing errors detected. (0 setup
      errors, 0 hold errors)
Maximum delay is  29.934ns.
```

- مقایسه با 31.128 بدون جاسازی