Clocking Strategies in VLSI Circuits

Dr. Shubhajit Roy Chowdhury,

Centre for VLSI and Embedded Systems Technology,
IIIT Hyderabad, India

Email: src.vlsi@iiit.ac.in

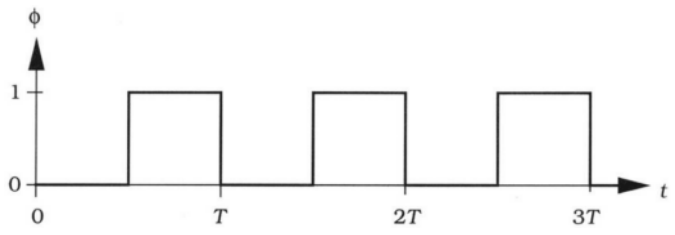
Clocked Flip-flops

- *Synchronous* design employs clocking signals to coordinate the movement of data through the system
- The data bit D is loaded into the DFF only on a rising clock edge

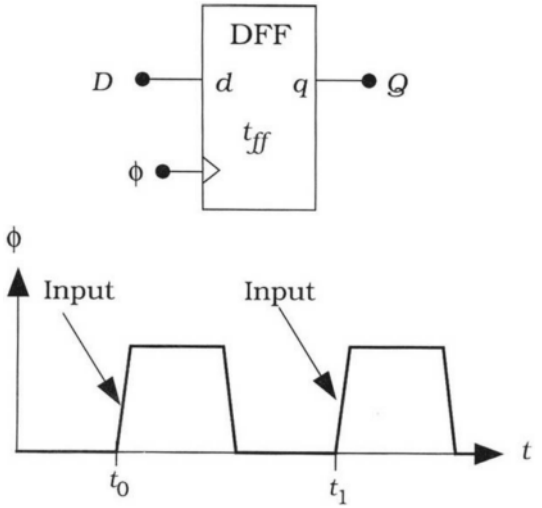
$$Q(t_0 + t_{ff}) = D(t_0)$$

Where t_o is the rise edge, and t_{ff} is the time delay when the output has this value

- In high speed design, the limiting circuit factor is the DFF delay time t_{ff} that is determined by the electronics and the load
  - Decreasing t_{ff} allows for a higher frequency clock
  - The tradeoff of speed and power



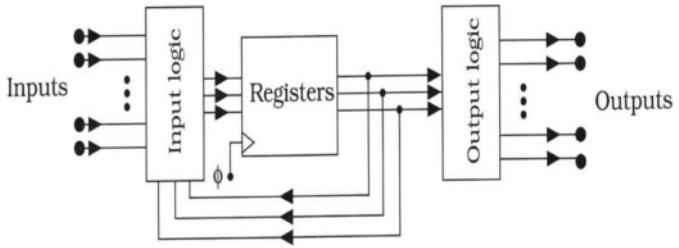
Ideal clocking signal



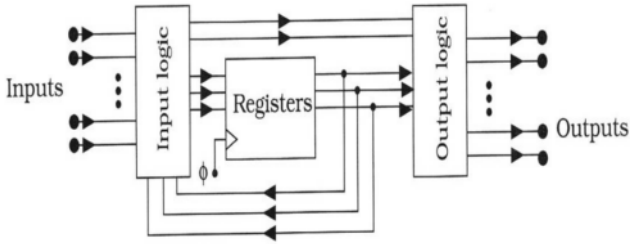
Timing in a DFF

Classical State Machines

- Two models for state machines that use single-clock timing are shown
- ❑ Moore machine and Mealy machine
- ❑ Huffman model: contains both the Moore and Mealy models

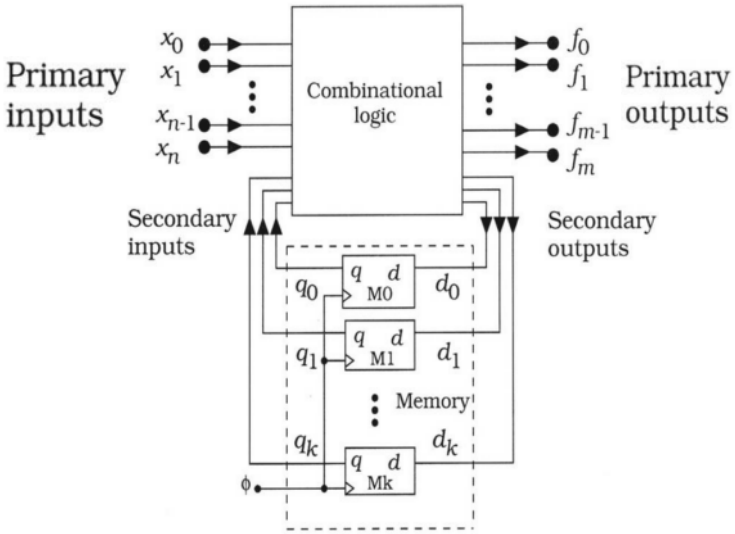


(a) Moore machine



(b) Mealy machine

Moore and mealy state machines



Huffman model of a state machine

Classical State Machines

- FPGA design are also heavily based on classical state machine theory
  - For examples, in combinational logic, Individual gates, PLAs, Programming Logic Device (PLD), and groups of multiplexors
  - Programming is achieved with EPROMs, fuses, SRAM arrays, and some contain lookup tables (LUT, e.g. FPGA) to aid in the design

$$\sum_r m_r$$

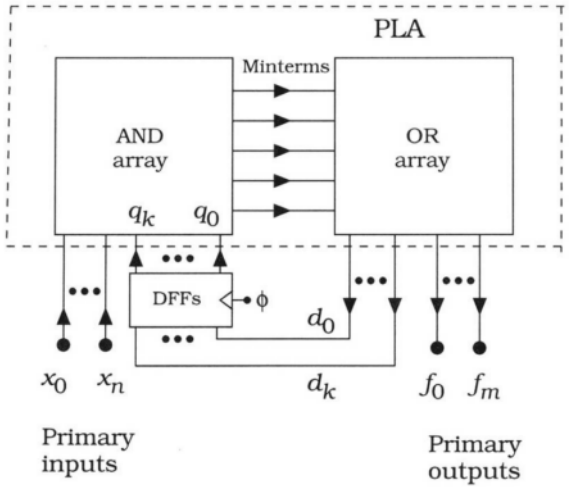
(The AND-plane of the PLA can be programmed to produce minterms m_r)

$$T > t_{ff} + t_d + t_{su} + t_h$$

(The clock T must be large enough to allow completion)

Where,

- t_{ff} is the delay time form input to output of the flip-flop
- t_d is the logic delay time through the PLA
- t_{su} is the "setup time" of the flip-flop
- T_h is the "hold time" of the flip flop



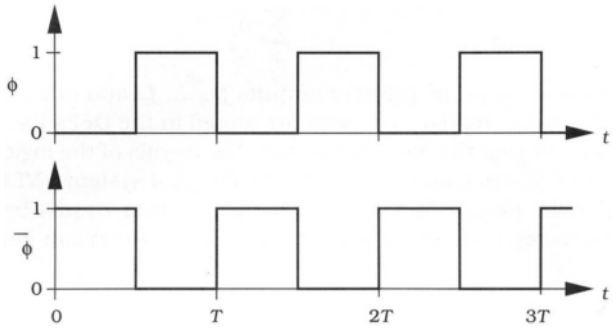
Huffman state machine using PLA logic

Clocked Logic Cascades

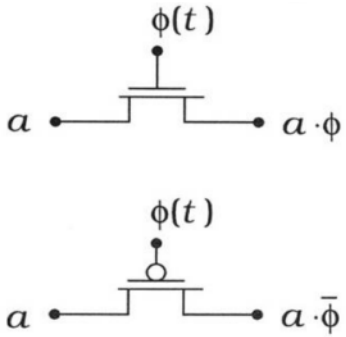
- Ideal waveforms for the complimentary signal

$$\phi \cdot \bar{\phi} = 0$$

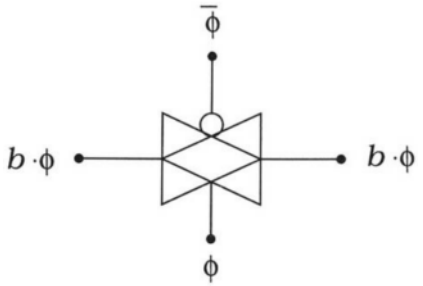
$$V_{\max} = V_{DD} - V_{Tn}$$



Complementary clocks

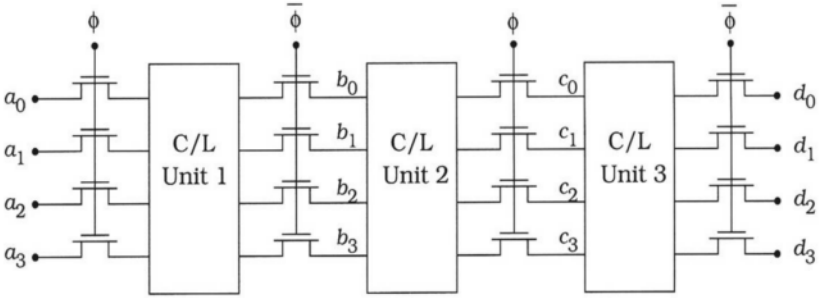


(a) FETs



(b) Transmission gate

Clock-controlled transistors

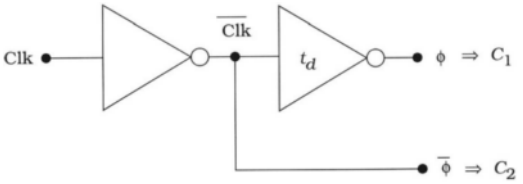


A clocked cascade

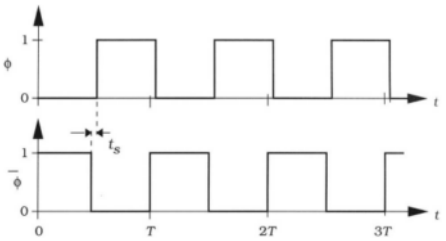
Timing Circles and Clock Skew

- *Timing circles*: are simple constructs that can be useful for visualizing data transfer
  - this defines what is known as a 50% duty cycle

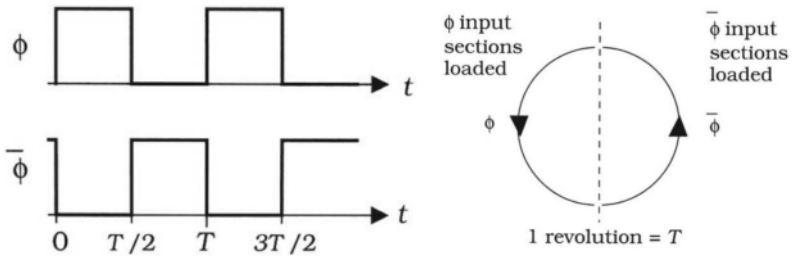
- *Clock skew*: the timing of a clock is out of phase with the system reference



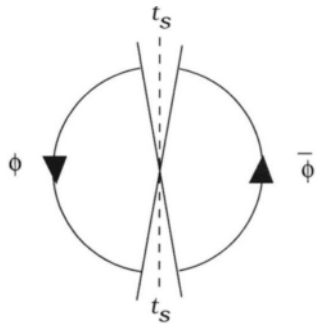
Clock generation circuit



Clock skew



Timing circle for a single-clock, dual-phase cascade



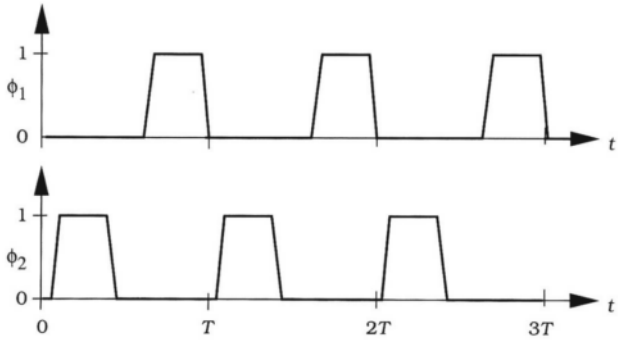
Timing circle with clock skew

Dual Non-overlapping Clocks

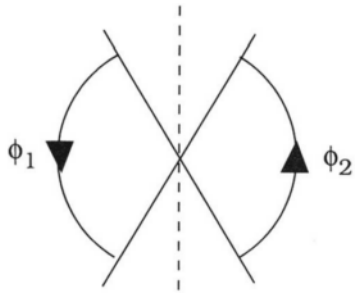
- In this technique, two distinct non-overlapping clocks ψ_1 and ψ_2 are used such that

$$\phi_1(t) \cdot \phi_2(t) = 0$$

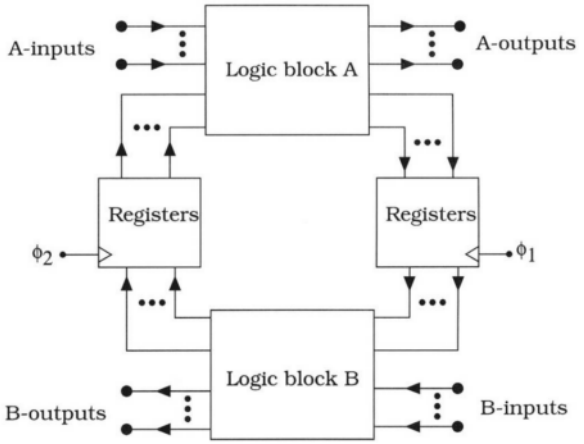
- Finite-state machines that are based on dual-clock schemes can provide powerful interactive capabilities



Dual non-overlapping clocks



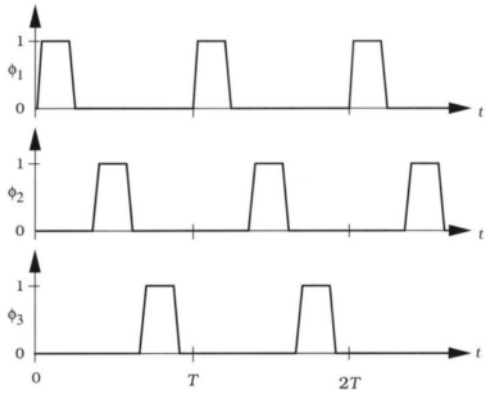
Timing circuit for a 2-clock network



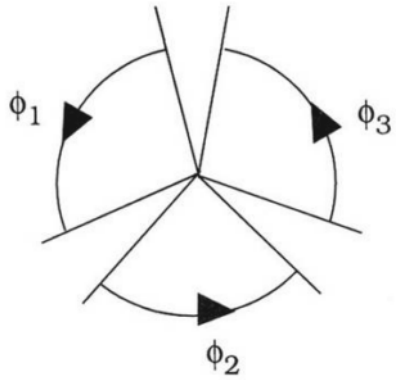
A dual-clock finite-state machine design

Other Multiple-clock Schemes

- It is possible to create different mutliple-clock schemes to control clocked logic cascades and state machine
  - For example, a triple, non-overlapping clock set
- However, in modern high-speed VLSI, complicated clocking schemes introduce too many problems
  - Solution: speed gains are accomplished by improved *circuit design*, *processing*, and *architectural modifications*
  - The most popular approach is to use a *single-clock, dual-phase system*



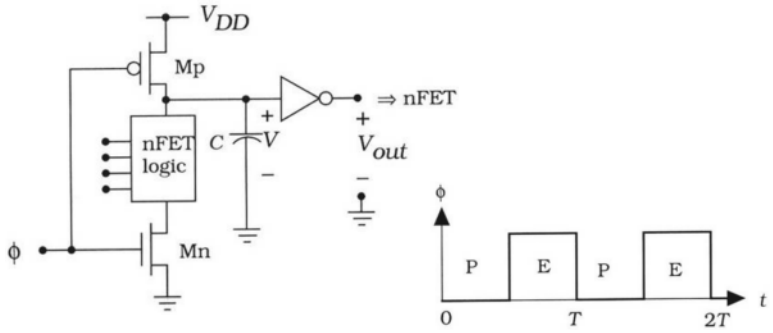
Triple, non-overlapping clock signals



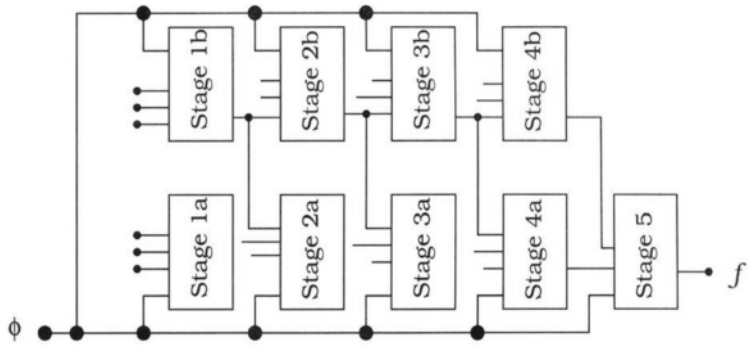
Timing circle for a 3-clock non-overlapping network

Dynamic Logic Cascades

- Dynamic logic circuits achieve synchronized data flow by controlling the internal operational states of the logic gate circuits
  - Typical domino logic state:
  - P: pre-charge phase
  - E: Evaluation phase



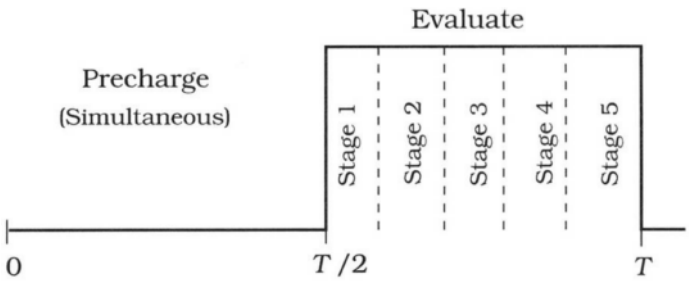
Operation of a domino logic state



A dynamic logic cascade

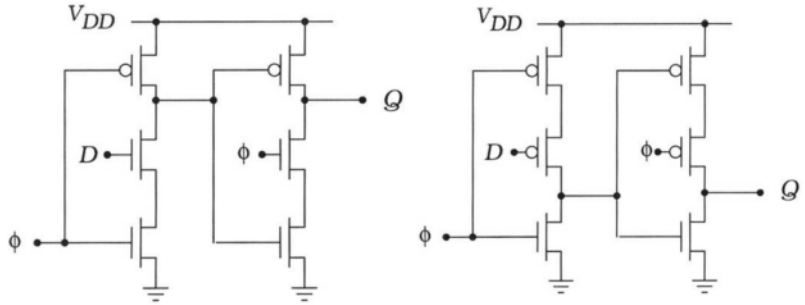
Dynamic Logic Cascades

- The data transfer into and out of a dynamic logic cascade is sequenced with the clock
  - The number of stages that can be included in the chain is determined by the delay for the case where every stage switches



Timing sequence in the domino cascade

- True Single-Phase Clock (TSPC): use only a single clock ψ throughout

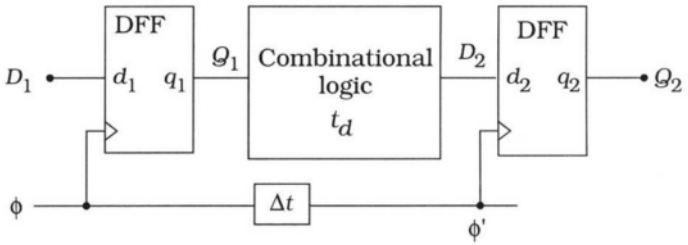


(a) n-block (b) p-block

True single-phase clock latches

Basic Concept of Pipelining

- Pipelining is a tech. that is used to increase the throughput of a sequential set of distinct data inputs through a synchronous logic cascade



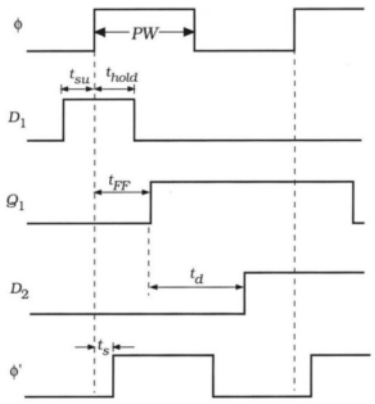
Basic pipelined stage for timing analysis

$$T > t_{ff} + t_d + t_{su} + t_s$$

$$t_{hold} < PW$$ (PW: pulse width of the clock)

$$f < \frac{1}{t_{ff} + t_d + t_{su} + t_s}$$

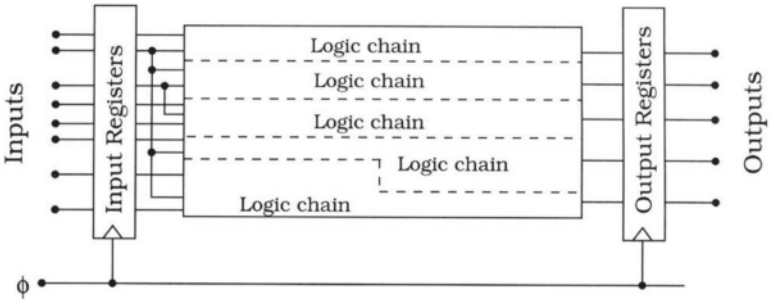
Where,
- t_{ff}: flip-flop delay time
- t_d : logic delay time
- t_{su} : setup time of the flip-flop
- t_{hold} : hold time of the flip-flop



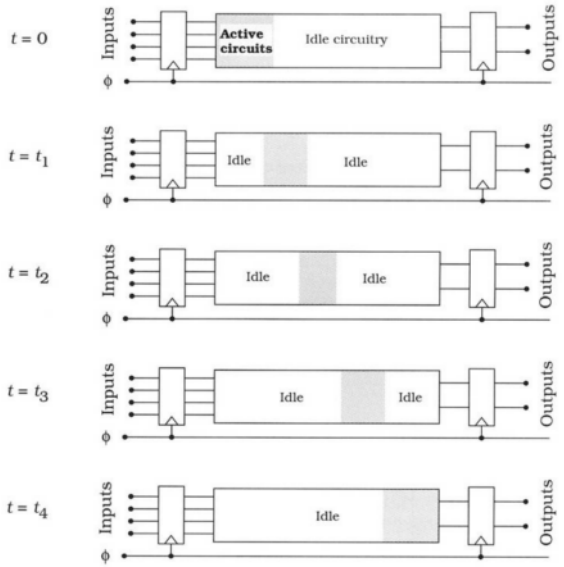
Waveform quantities for timing analysis

Pipelining

- Pipelined systems are designed to increase the overall throughput of a set of sequential input states by dividing the cascade into small visualization of the problem
- Once a circuit completes a calculation and passes the result on to the next stage, it remain idle for the rest of the clock cycle



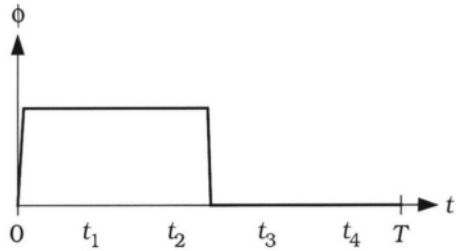
Logic chains in a clocked system



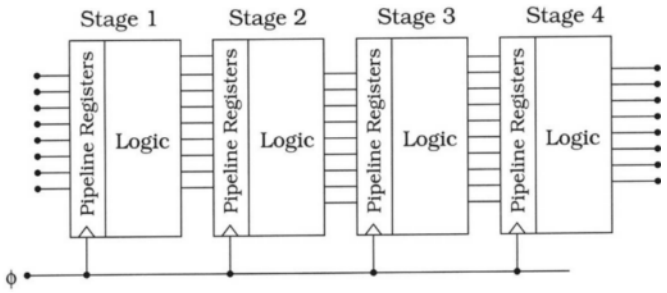
Circuit activity in a logic cascade

Pipelining

- Since the delay through a logic gate varies its complexity and parasitics, the logic propagation rate will not be uniform

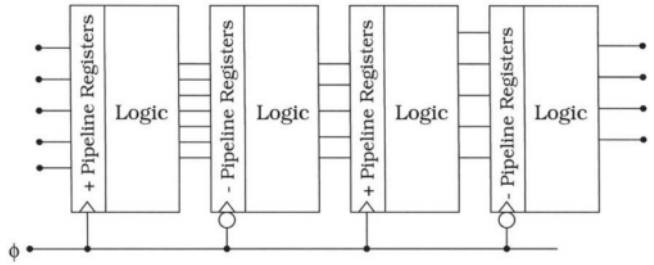


Progression times in the logic cascade



A 4-stage pipeline

- Dividing the long logic into small groups, add registers between the sections, and use a faster clock, then most of the circuits will be active at any given time



Pipeline with positive edge and negative edge-triggering

$$T_i > t_{ff} + t_{su} + t_{d,i} + t_{s,i+1}$$

$$T_{pipe} = \max\{T_1, \ldots, T_m\}$$

Clock Distribution

- When frequencies *f* reach the 1GHz (10^9 Hz) level corresponding to a clock period of

$$T = \frac{1}{f} = 1 \ \ ns$$

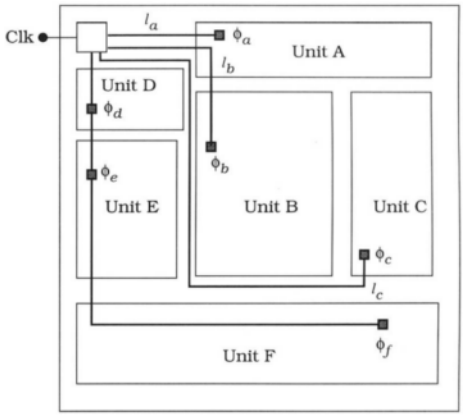
- However, distribution of the clocking signal to various points of the chip is complicated because the intrinsic RC time delay τ increases as the square of the line length *l* according to

$$\tau = Bl^2$$

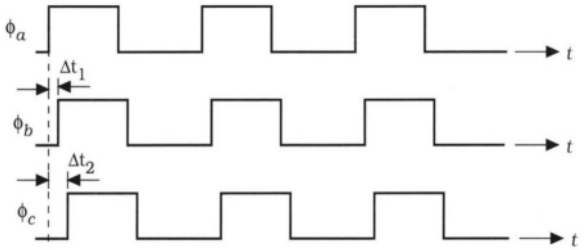
$$\Delta t_1 = B(l_b^{\ 2} - l_a^{\ 2})$$

$$\Delta t_2 = B(l_c^{\ 2} - l_b^{\ 2})$$

- Problems: clock skew and signal distortion will be very difficult to deal with in large chips

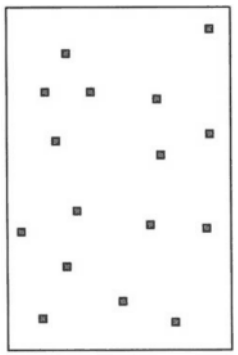


A 4-stage pipeline

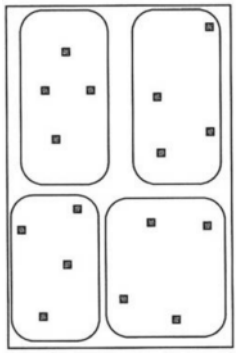


A 4-stage pipeline

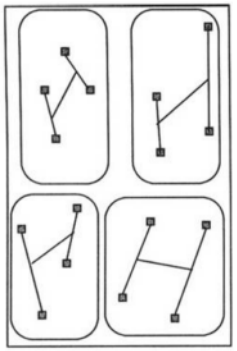
Clock Routing and Driver Trees



(a) Clocking points

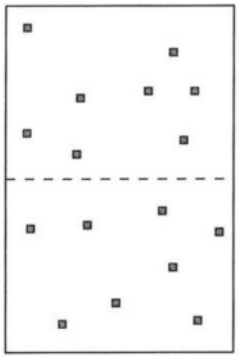


(b) First grouping

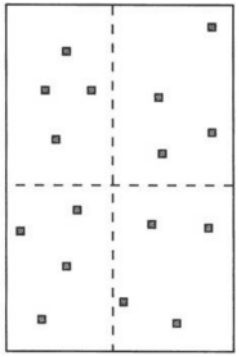


(c) Interior routing

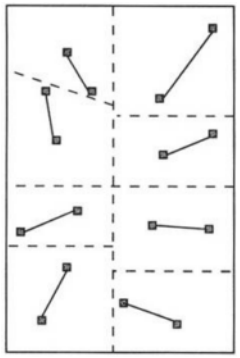
Simplified view of the clock routing problem



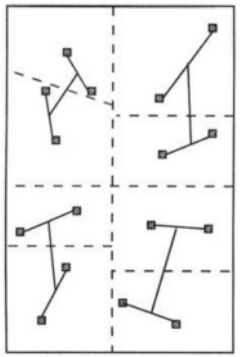
(a) First



(b) Second



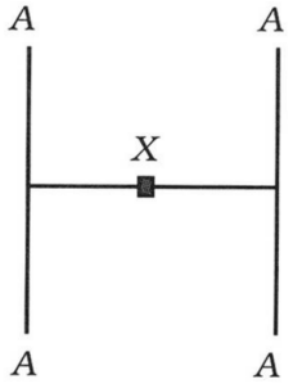
(c) Third



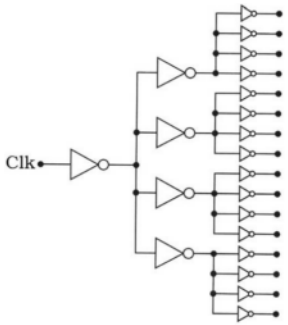
(d) Routing

Partitioning steps for defining clocking groups

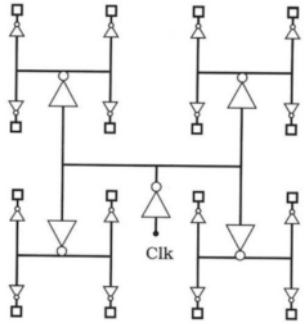
Driver Tree



Geometrical analysis of the letter "H"

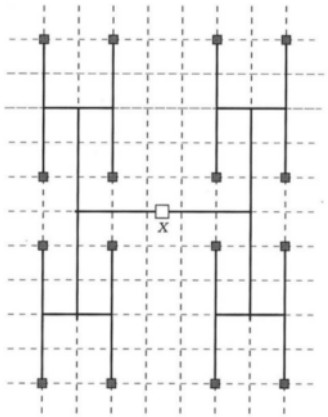


(a) Driver tree

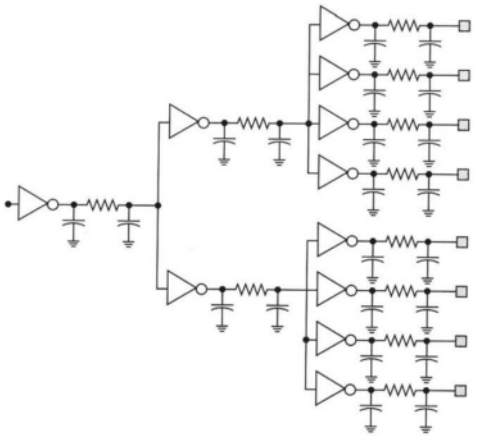


(b) Application to H-tree

Driver tree arrangement



Macro-level H-type distribution tree



Driver tree design with interconnect parasitics

Questions?