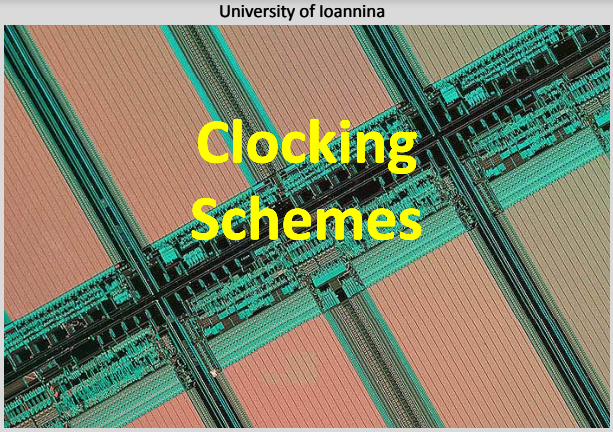
CMOS INTEGRATED CIRCUIT DESIGN TECHNIQUES

University of Ioannina



Dept. of Computer Science and Engineering

Y. Tsiatouhas

CMOS Integrated Circuit Design Techniques

Overview

1. ***Jitter-Skew – Throughput-Latency***
2. ***Pipeline structures***
3. ***Clocking schemes***
4. ***Skew tolerant design***
5. ***Slack Borrowing***
6. ***Time stealing***

VLSI Systems
and Computer Architecture Lab

Clock Jitter – Clock Skew

Clock Jitter

Clock jitter is a temporal variation (uncertainty) of the clock period at a given point in the chip. The clock period can reduce or expand on a cycle-by-cycle basis. Clock jitter is the inherent inaccuracy of the clock generation circuitry (e.g. PLLs).

Clock Skew

Clock skew is a variation on the arrival time of a clock signal transition due to static mismatches and process variations in the clock paths and differences in the clock load. Clock skew is the clock inaccuracy introduced by the clock distribution network.
Clock skew is constant from cycle to cycle.

Throughput – Latency

Metrics for the performance evaluation of circuits / systems.

Throughput

Throughput (παραγωγικότητα) is defined as the processing rate of the input data by the circuit / system.
Equivalently, it is the data transfer rate inside the circuit.
Throughput is related to the clock frequency (clock cycle).

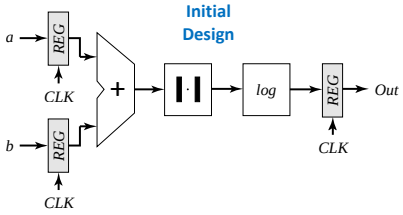
Latency

Latency (λανθάνων χρόνος) is defined as the time required by the circuit / system to complete a computation.
In case that the required computation time is available inside a clock cycle, then latency and throughput are conversely proportional between each other.

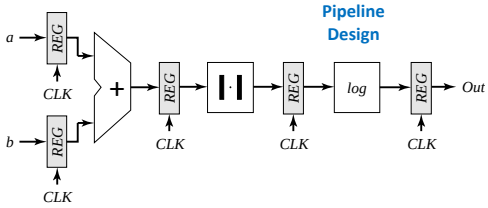
Pipeline Structures

Pipeline Structures I

Initial Design



Pipeline Design



$$T_{min,org} = t_{c\to q} + \left(t_{p\_add} + t_{p\_abs} + t_{p\_log}\right) + t_{su}$$

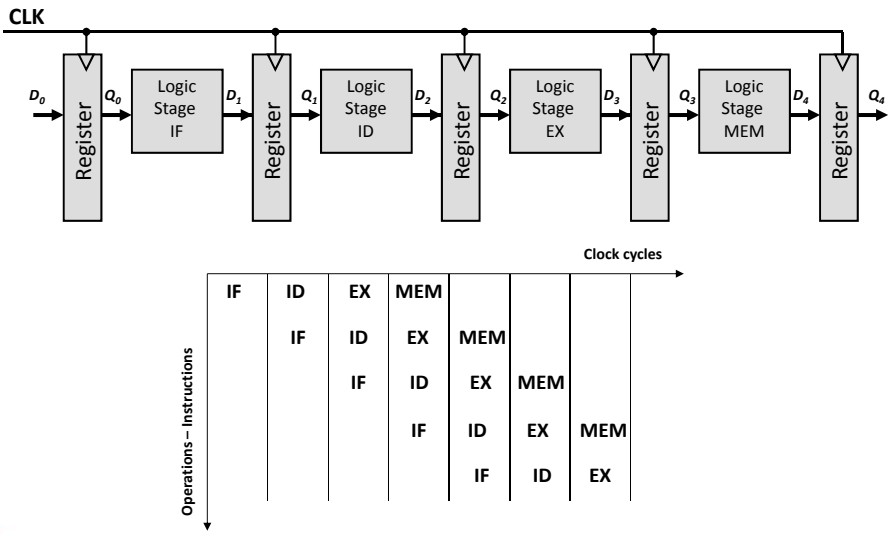
$$T_{min,pipe} = t_{c\to q} + \max\left(t_{p\_add}, t_{p\_abs}, t_{p\_log}\right) + t_{su}$$

in case that $t_{p\_add} = t_{p\_abs} = t_{p\_log}$

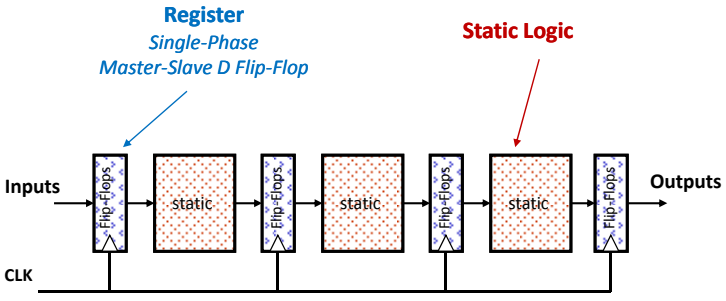
then $T_{min,pipe} \cong T_{min,org}/3$

Clock Period	Adder	Absolute Value	Logarithm
1	$a_1 + b_1$		
2	$a_2 + b_2$	$\lvert a_1 + b_1 \rvert$	
3	$a_3 + b_3$	$\lvert a_2 + b_2 \rvert$	$\log(\lvert a_1 + b_1 \rvert)$
4	$a_4 + b_4$	$\lvert a_3 + b_3 \rvert$	$\log(\lvert a_2 + b_2 \rvert)$
5	$a_5 + b_5$	$\lvert a_4 + b_4 \rvert$	$\log(\lvert a_3 + b_3 \rvert)$

Pipeline Structures II

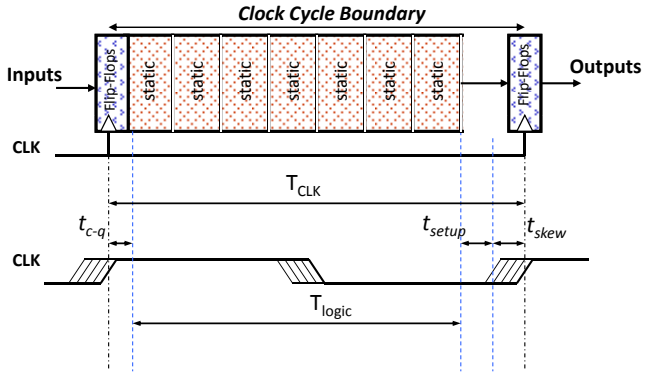


Single-Phase Master-Slave Clocking I



Positive edge triggered D Flip-Flops

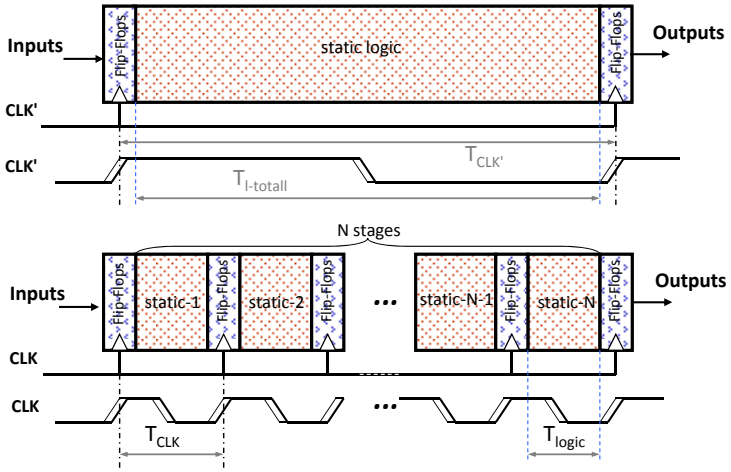
Single-Phase Master-Slave Clocking II



Available Time for Logic Evaluation:

$$T_{logic} = T_{CLK} - t_{c-q} - t_{setup} - t_{skew} = T_{CLK} - T_{overhead}$$

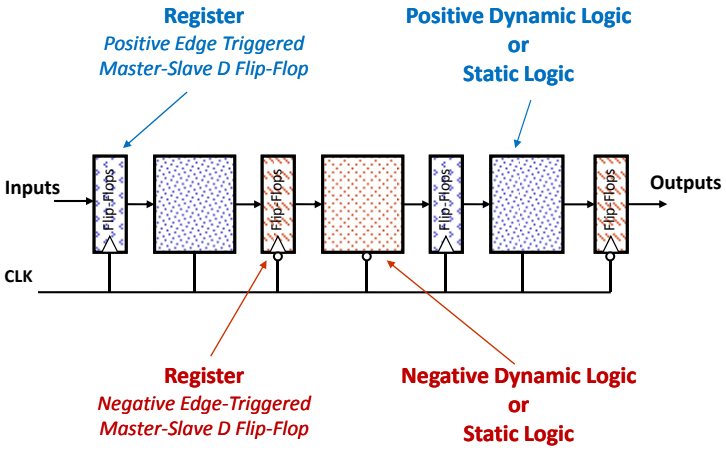
Overhead Impact in Pipelines



$$T_{logic} = \frac{T_{I-totall}}{N} \Rightarrow \quad T_{CLK} = T_{logic} + T_{overhead} \Rightarrow \quad T_{latency} = N \cdot T_{CLK} = T_{I-totall} + N \cdot T_{overhead}$$

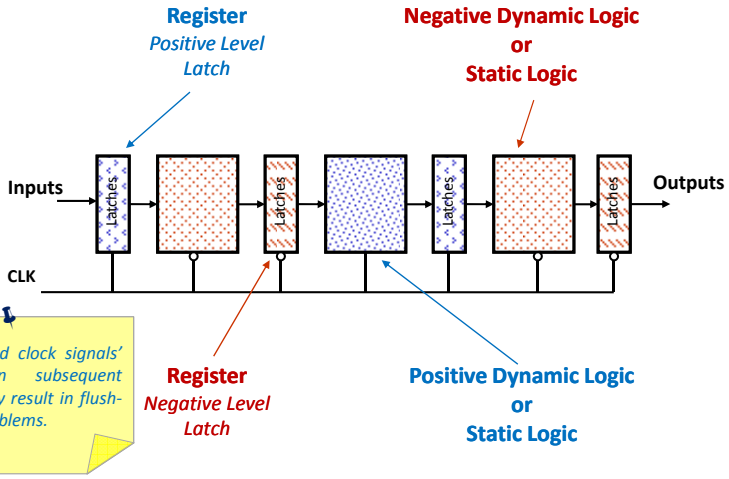
Single-Phase Double Edge Clocking

Concurrent Register and Subsequent Logic Activation



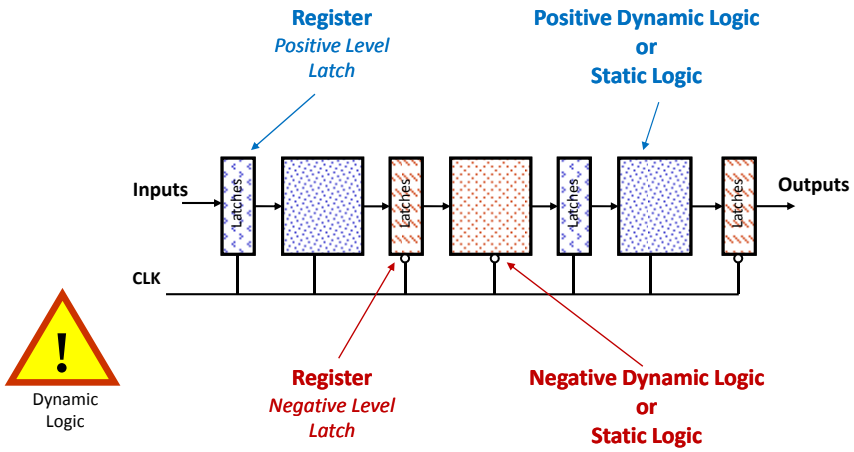
Single-Phase Two-Level Clocking

Concurrent Logic and Subsequent Register Activation



Single-Phase Two-Level Clocking

Concurrent Register and Subsequent Logic Activation



Multi-Phase Clocking

Two-Phase Clocking

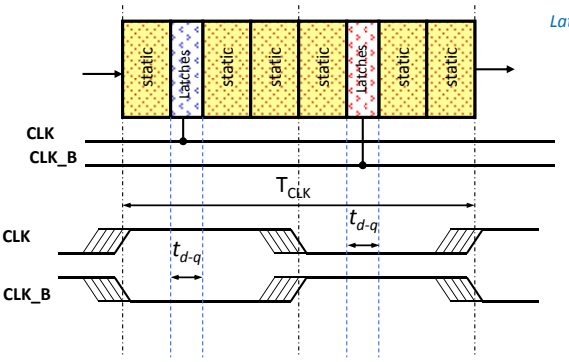
In two-phase clocking systems two discrete clock phases are utilized, which are generated by the main clock signal at the last level of the clock distribution network.

Overlapping clock signals can be used or not. In the first case higher speeds can be achieved at the risk of increased signal integrity problems due to skew related issues in the clock distribution.

Four-Phase Clocking

Four-phase clocking systems utilize four discrete clock phases. In general, design techniques with more than two phases are not very common in system development.

Skew Tolerant Static Circuits



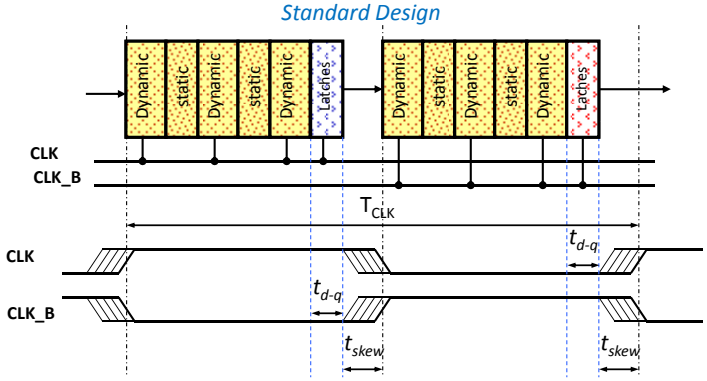
Latch memory mode @ low!

Available Time for Logic Evaluation:

$$T_{logic} = T_{CLK} - 2t_{d-q}$$

Skew Tolerant Domino Circuits I

Standard Design



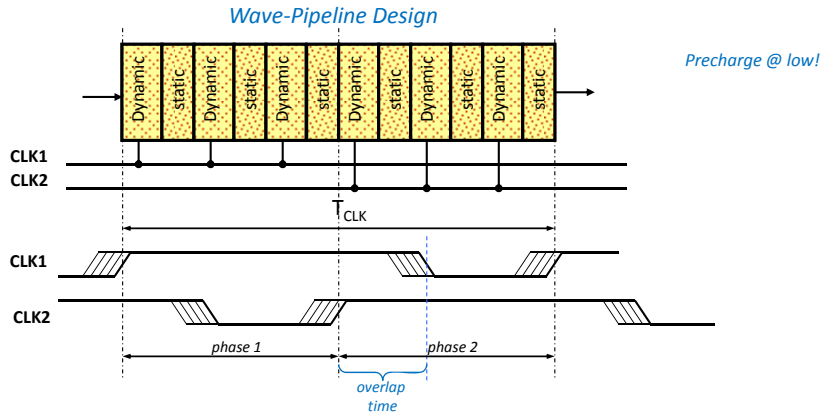
Memory @ low!

Precharge @ low!

Available Time for Logic Evaluation:

$$T_{logic} = T_{CLK} - 2t_{d-q} - 2t_{skew}$$

Skew Tolerant Domino Circuits II



Available Time for Logic Evaluation:

$$T_{logic} = T_{CLK} \quad !$$

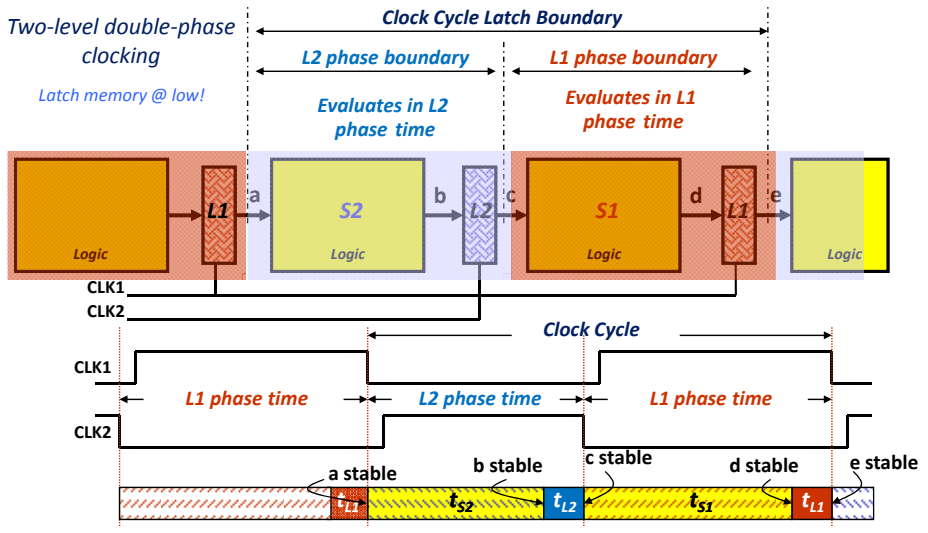
Slack Borrowing

In the *slack borrowing* technique, a logic partition utilizes time left over (slack) by the previous partition.
By definition this additional time is automatically (voluntarily) surrendered without circuitry and/or clock arrival time adjustments.

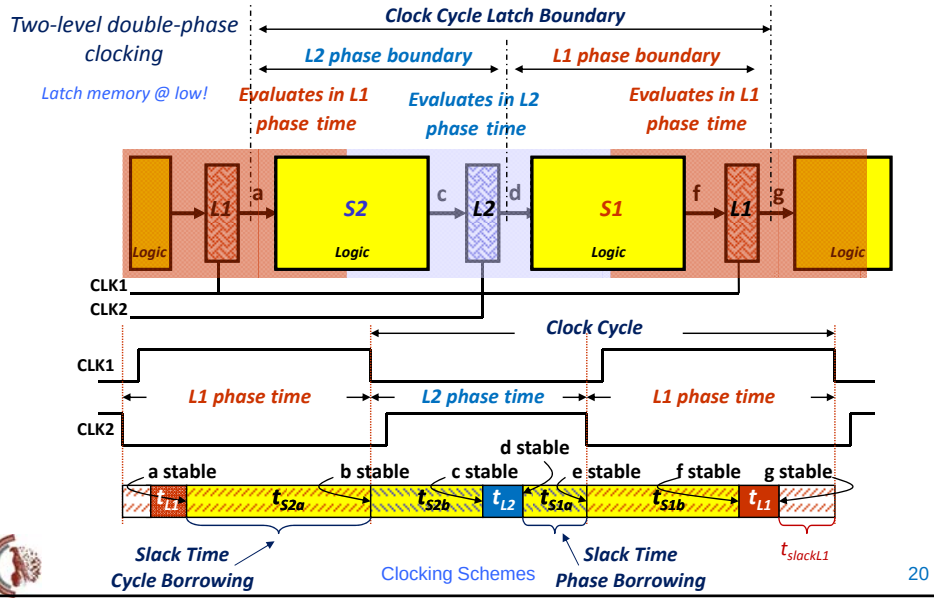
This technique is suitable for use in static logic with two phase, two-level clocking (latch-based) designs.

- ***Cycle slack borrowing***: permits logic to use more than one cycle time and still fit within a single clock cycle boundary while maintaining the overall machine cycle time. The time used for logic evaluation exceeds one cycle and the machine still works at speed. The slack time is borrowed from preceding cycle(s).
- ***Phase slack borrowing***: permits logic to use more than one phase time and still maintain the overall machine cycle time. The time used for logic evaluation in a clock phase exceeds the clock phase time and the machine still works at speed. The slack time is borrowed from preceding phase(s).

Typical Operation



Slack Borrowing



Slack Borrowing: Clocking Issues (I)

Cycle Slack Borrowing

The logic propagation delay within a *clock cycle latch boundary* is:

$$t_{dcycle} = t_{S2a} + t_{S2b} + t_{L2} + t_{S1a} + t_{S1b} + t_{L1}$$

The *clock cycle time* is:

$$t_{cycle} = t_{S2b} + t_{L2} + t_{S1a} + t_{S1b} + t_{L1} + t_{slackL1}$$

The time difference between the clock cycle latch boundary and the clock cycle time, is:

$$t_{dcycle} - t_{cycle} = t_{S2a} - t_{slackL1}$$

$t_{S2a(max)} = 0.5 \cdot t_{cycle} \Rightarrow$

$t_{dcycle(max)} = 1.5 \cdot t_{cycle}$

Slack Borrowing: Clocking Issues (II)

Phase Slack Borrowing

The logic propagation delay within an L2 *phase latch boundary* is:

$$t_{dL2phase} = t_{S2a} + t_{S2b} + t_{L2}$$

The L2 *phase cycle time* is:

$$t_{pL2} = t_{S2b} + t_{L2} + t_{S1a}$$

The time difference between the L2 phase latch boundary and the L2 phase cycle time, is:

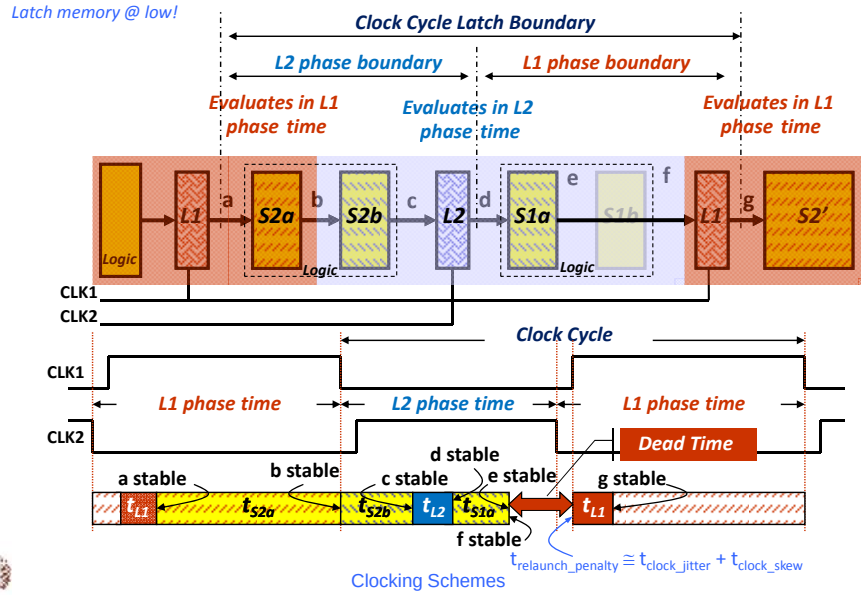
$$t_{dL2phase} - t_{pL2} = t_{S2a} - t_{S1a}$$

$t_{S1a(max)} = 0.5 \cdot t_{cycle}$

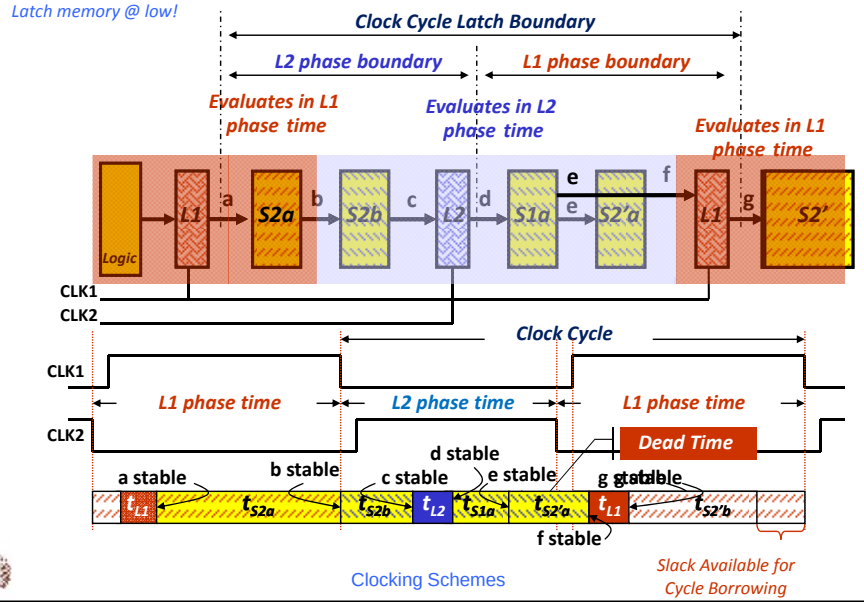
$t_{S2a(max)} = 0.5 \cdot t_{cycle} \Rightarrow$

$t_{dL2phase(max)} = 1.0 \cdot t_{cycle}$

Dead Time and Latch Relaunch Penalty



Phase Partitioning



Time Stealing

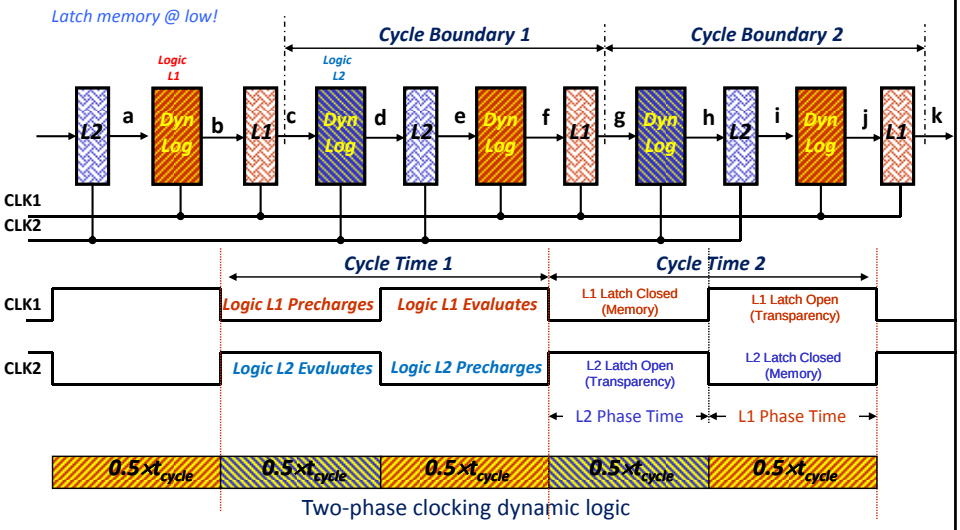
In the *time stealing* technique, a logic partition gains evaluation time by taking (stealing) it from the next clock cycle.

It is suitable for use in dynamic logic with two-phase clocking or in static logic with single phase master-slave clocking.

The additional time is involuntarily surrendered and it is obtained by adjusting the clock edges arrival times.

- ***Cycle time stealing***: permits logic to use more than one cycle time and still fit within a single clock cycle boundary while maintaining the overall machine cycle time. The time used for logic evaluation exceeds one cycle and the machine still works at speed. The time is stolen from the subsequent cycle.
- ***Phase time stealing***: permits logic to use more than one phase time and still maintain the overall machine cycle time. The time used for logic evaluation in a clock phase exceeds the clock phase time and the machine still works at speed. The time is stolen from the subsequent phase.

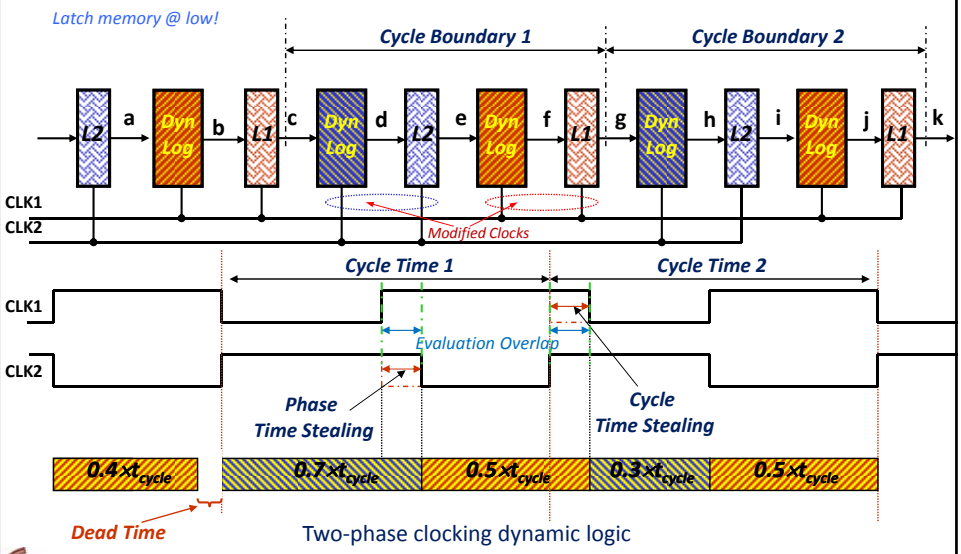
Two-Phase Typical Dynamic Logic



Two-phase clocking dynamic logic

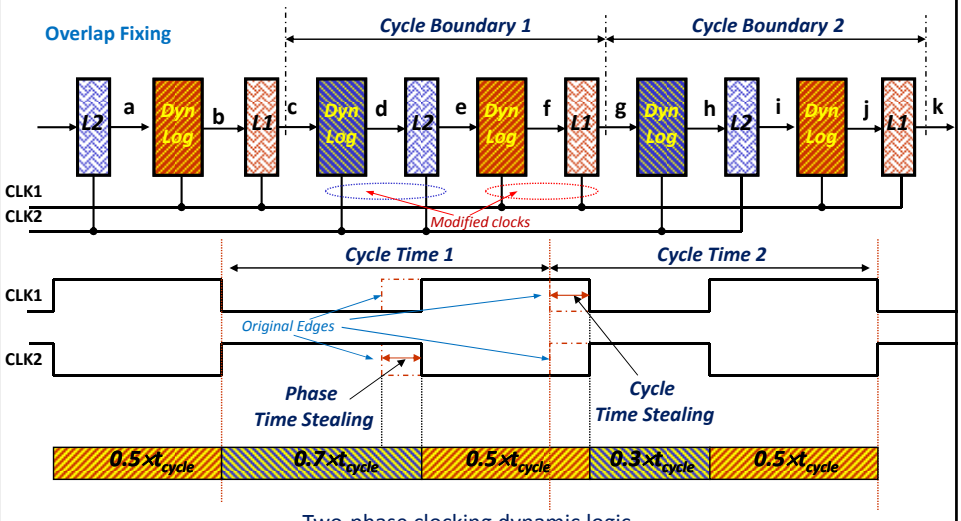
(non-overlapping clocks)

Time Stealing – Dynamic Logic I



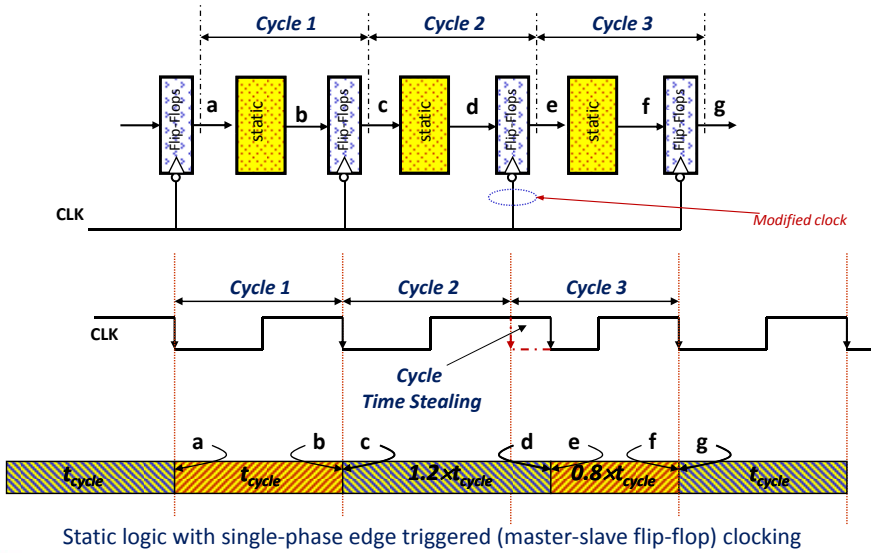
Two-phase clocking dynamic logic

Time Stealing – Dynamic Logic II



Two-phase clocking dynamic logic

Time Stealing – Master-Slave Flip-Flops



Static logic with single-phase edge triggered (master-slave flip-flop) clocking

References

- "*High Speed CMOS Design Styles*," K. Bernstein et al, Kluwer, 1999.
- "*Skew-Tolerant Circuit Design*," D. Harris, Morgan Kaufmann Pub., 2001.
- "*Digital Integrated Circuits: A Design Perspective*," J.M. Rabaey, A. Chandrakasan and B. Nikolic, Prentice Hall, 2003.
- "*CMOS VLSI Design: A Circuits and Systems Perspective*," N. Weste and D. Harris, Addison Wesley, 2011.