EE382M
VLSI-II

Static Timing Analysis

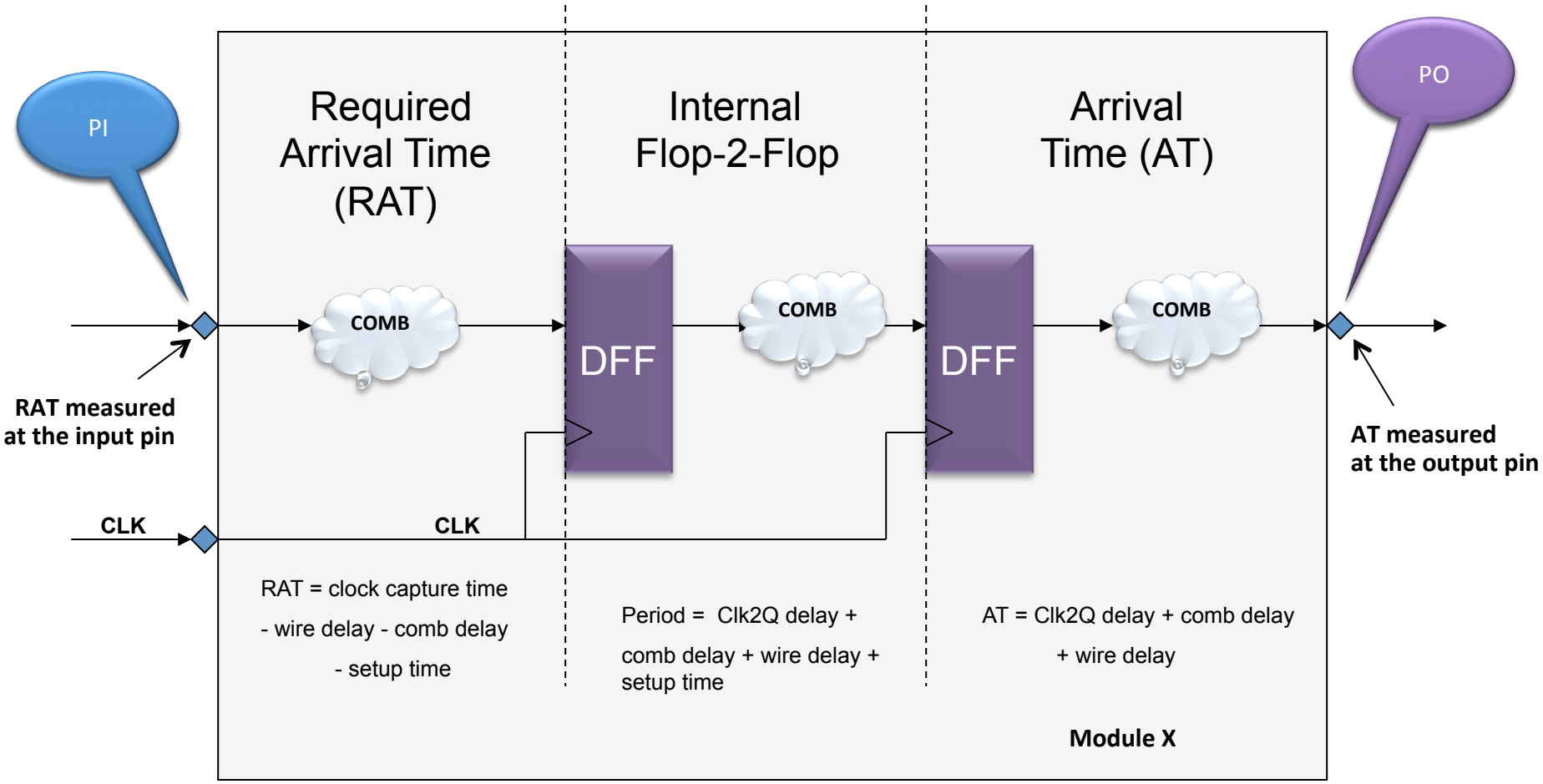
Spring 2017

Mark McDermott
Matthew J. Amatangelo

Agenda

- Basics of Static Timing Analysis & Statistical Timing Analysis
- The Timing Graph – the Timer's internal map
- Timing graph attributes: ATs, RATs, and Slacks
- Clock considerations
- Analyses performed
  - Late mode ("max mode")
  - Early mode ("min mode")

Basics of Timing: AT, RAT, Cycle time



Timing Analysis Techniques

- **There are 3 basic timing analysis techniques**

1. **Spice for entire networks and/or macro cross-sections.**

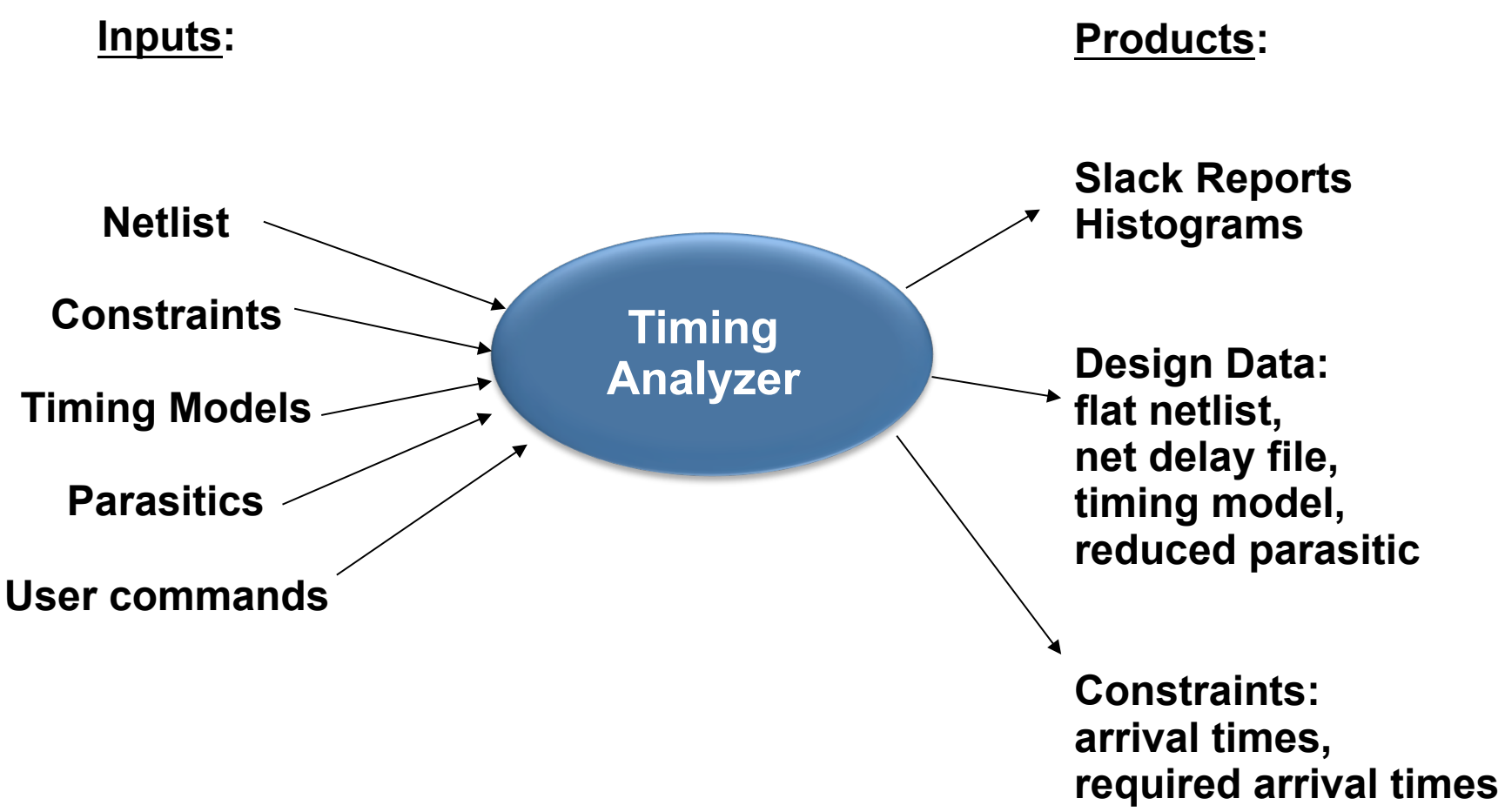
2. **Dynamic Simulation**
   - Coverage dependent on quality of the set of input vectors.
     - A typical microprocessor will require >1M years to run all timing simulations
   - Examination of logic failures not comprehensive in analyzing problems
     - root cause difficult to determine
   - Determines whether an event will occur.
   - Advantage - Does not time non-functional paths.
   - Disadvantage - How do you know all functional paths were timed?

3. **Static Timing Analysis**
   - Input vector pattern independent
     - traverses "all" paths between endpoints
   - Every source of data launch is checked at destinations or sinks
     - min and max delay values saved for each arc
   - Determines the worst possible time an event will occur
   - Advantage - Comprehensive, guarantees all paths are analyzed.
   - Disadvantage - Does not distinguish between functional & non-functional paths.

Key points of Static Timing Analysis (STA)

- Determines best/worst case arrival time of signals at all pins of design elements
- Accuracy is only as good as cell timing models
- Does not test functionality
- Uses ATs and RATs to determine path timing violations
  - Path comprised of launching and capturing components
  - Clocks and sequential elements define RATs
  - Any combinatorial element with clock input is sequential
  - Sorted by capturing clock. Clock phase is important.
- Uses timing graph of delay "arcs" and "checks" to represent the design
  - Particular information is stored at each node
  - Long and short path analysis is performed between source and sink points of graph after ATs and RATs have been propagated through graph
- Underlying assumptions enable STA to produce results
  - Reduced accuracy, ignored connections and effects must be managed
  - Beware of implied synchronicity of cross domain paths, especially where noise is a concern
- Sanity check required – expected results versus actual ones

Timing Analysis Environment



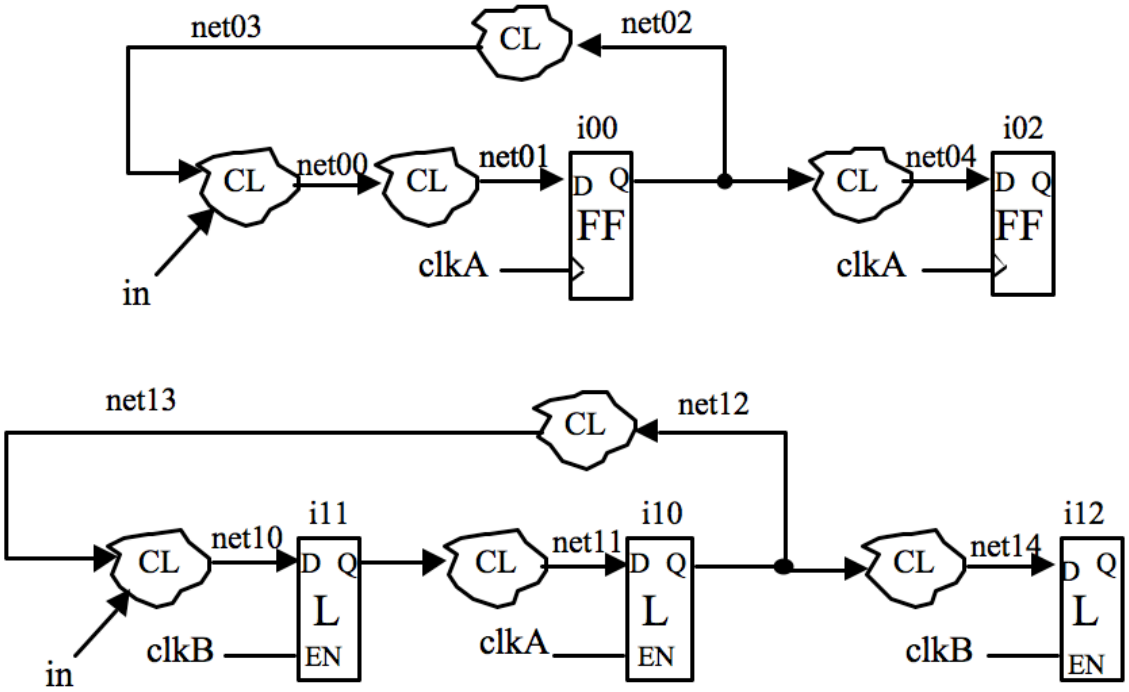
What is Static Timing Analysis?

- **What it does:**
  - Calculates latest and earliest possible switching times for each node in the design.
  - Determine the arrival time of signals for the worst case (latest or earliest) of all possible paths leading to a given node in the design
  - Compares calculated signal arrival times with expected (required) arrival times at storage elements.

- **What it is does not do:**
  - It does not perform a delay based functional simulation.
  - It does not consider the logical functioning of the circuits.
  - It does not analyze individual paths.
  - It doesn't search or analyze all possible paths (Path Pruning).

How a Static Timer Works (cont.)

- **For timing analysis, an acyclic graph is required.**
  - Timing analysers snip all loops of timing arcs in the graph.
    - In transparent latch designs there are frequently many more loops involving multiple latches in the path.
  - All events occur within one cycle of the defined simulation clock
    - adjustments made to arrival times to maintain single cycle context



How a Static Timer Works (cont.)

- The acyclic graph is then traversed from PIs and latch points (and other places loops were snipped) to POs and latch points in a single pass.
  - ATs and RATs are accumulated at every node during this traversal of the graph.
    - Slews are also calculated.
    - These times are calculated for all signals, clock & data.

- The timing analysis occurs when the tests that are resident at each node are applied to the AT/RAT information collected from the graph traversal.
  - This is then analysed, sorted, and written out as reports.

Static Timing Paths

- **Paths begin at source pins**
  - primary inputs (this may be a clock input)
- **Paths end at sink pins**
  - primary outputs
  - wherever timing checks are performed
    - storage elements/timing elements
    - timing elements include clock gates
  - where timer breaks loop paths
- **Paths represented as a series of segments in a timing graph**
  - logical constraints essentially omitted

A Note about STA Assumptions

- When loops are broken, there is an implied assumption by the timer
  - You must check to confirm that it is valid?
- Asynchronous paths in STA
  - No paths are considered asynchronous unless specifically delineated as such e.g., false synchronicity
    - Clock definitions are considered absolute and a rational number multiplier will be computed and used in slack calculations
    - Undermines coupling analysis by assuming aggressors can only affect victims in synchronously derived windows
- You must read and understand all timer log messages to fix any implied assumptions!
  - Examples:
    - Couldn't compute delay, so assumed 0
    - Found non-unate logic in clock trace, assuming even logic (r->r)
    - Object is not a valid endpoint, so not tested
    - Cell timing model characterization usually has assumed particular sensitizations that don't occur in reality

Agenda

- Basics of Timing Analysis
- **The Timing Graph – the Timer's internal map**
- Timing graph attributes: ATs, RATs, and Slacks
- Clock considerations
- Analyses performed
  - Late mode ("max mode")
  - Early mode ("min mode")

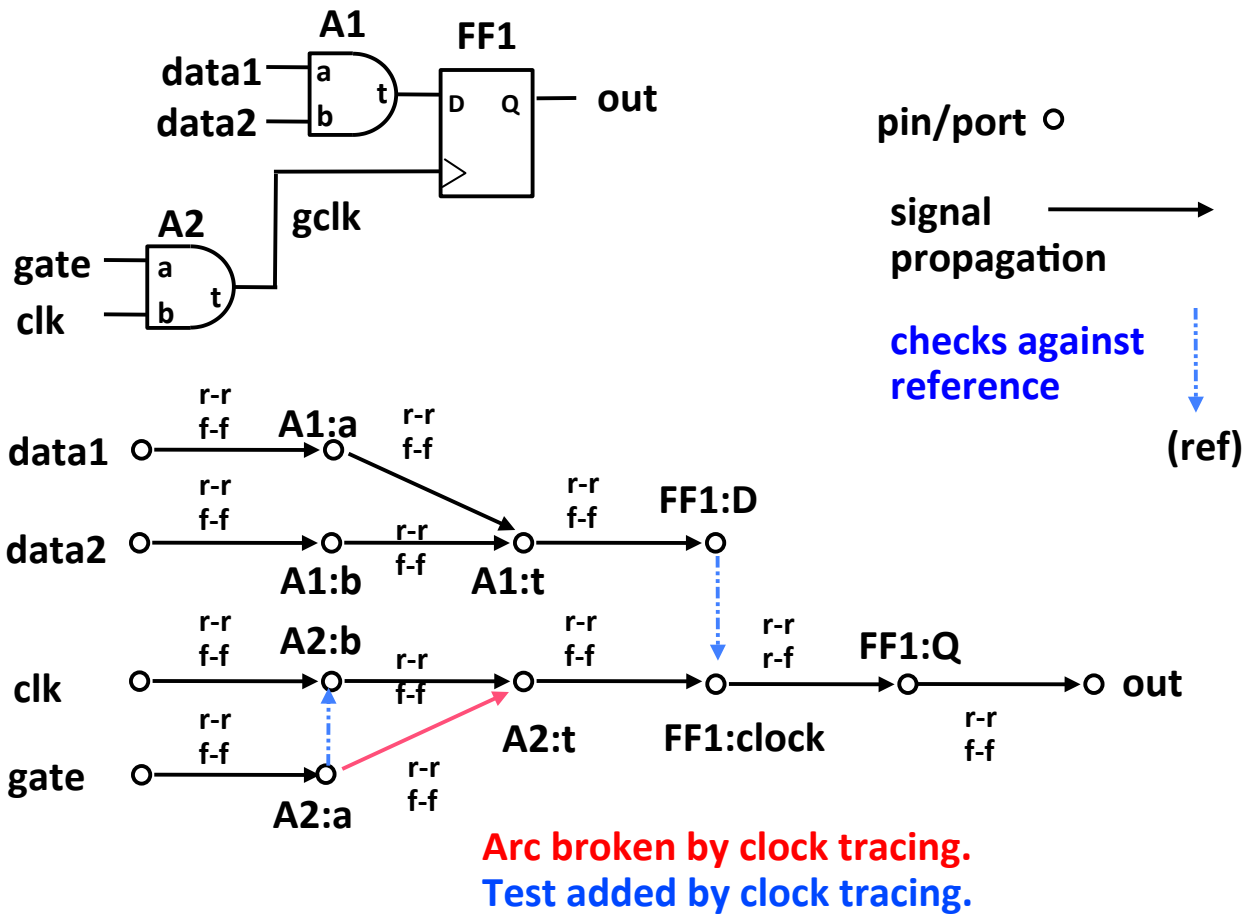
The Timing Graph

- **Arcs**
  - represent the min and max input to output propagation delays and appropriate slews for wires and logic blocks
    - provided in timing model per logic cell
    - calculated for wire and to consider input slew, load, noise – normally from 50% input voltage level to 50% of output
  - exist for each appropriate edge
  - uncontrolled or unconditional arcs
    - input signal influences output response directly regardless of any other signal's state
      - example: clock -> Q of latch
      - example: in -> out of unclocked combinatorial circuit
  - controlled or conditional arcs
    - input signal can only influence the output if the control signal is in a given state
      - example: data -> Q of latch; control signal “enable”

The Timing Graph (cont.)

- **Checks (tests)**
  - Exist whenever a signal arrival time is constrained by another signal's arrival (usually a clock)
  - Where they come from:
    - timing model (explicit)
    - timing checks are implied when clock and data signals meet
      - gate timer may provide checks when not defined in model
        - e.g., clock gates
  - Circuit functionality determines the checks – these are identified in the graph as timing elements
    - clock gate
    - latch
    - flip flop
    - domino
    - Etc…

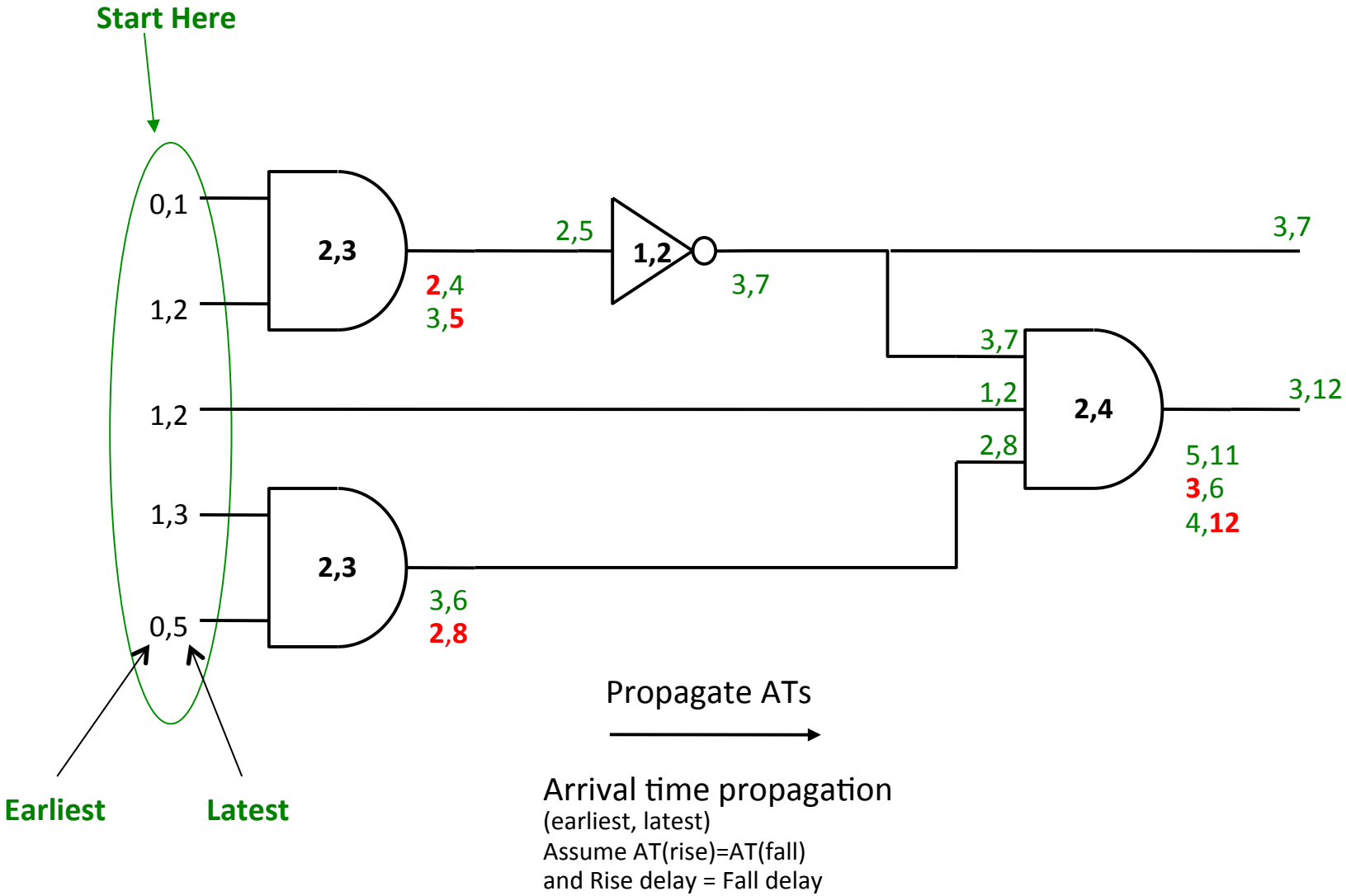
Example Timing Graph



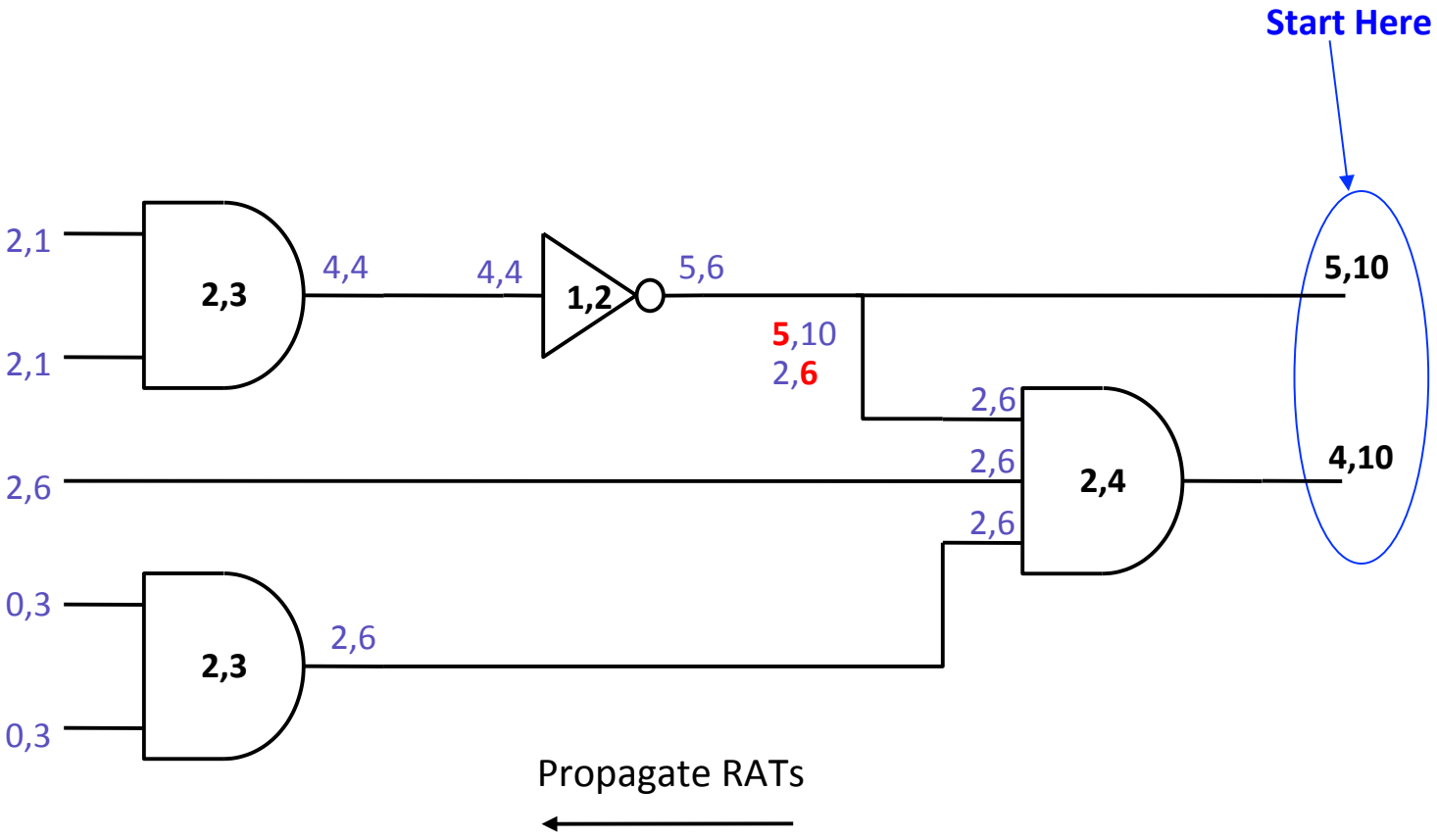
Agenda

- Basics of Timing Analysis
- The Timing Graph – the Timer's internal map
- **Timing graph attributes: ATs, RATs, and Slacks**
- Clock considerations
- Analyses performed
  - Late mode ("max mode")
  - Early mode ("min mode")

STA Example: Propagate ATs

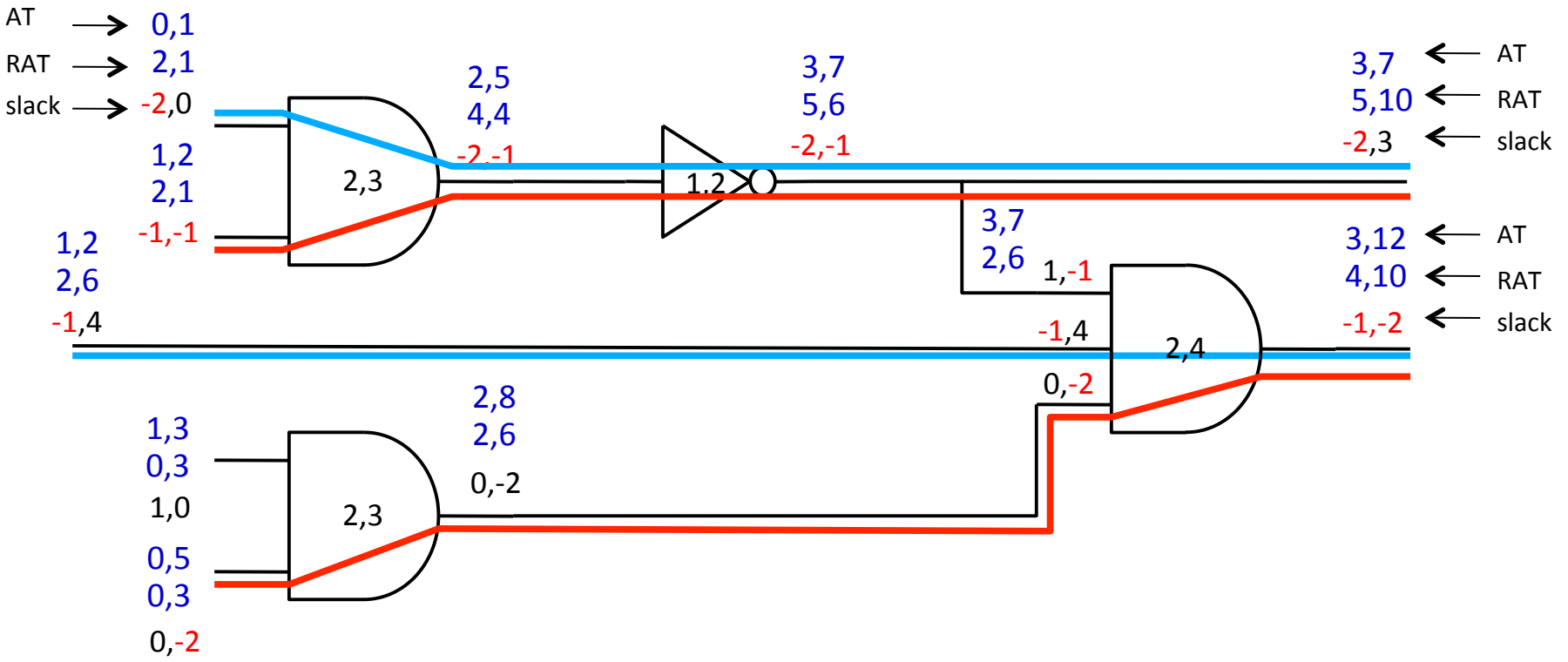


STA Example: Propagate RATs



Required time propagation
(earliest, latest)
Assume RAT(rise)=RAT(fall)
and Rise delay = Fall delay

STA Example: Slacks and Critical Paths



Slacks and critical paths
(earliest, latest)
slack(rise)=slack(fall)

Agenda

- Basics of Timing Analysis
- The Timing Graph – the Timer's internal map
- Timing graph attributes: ATs, RATs, and Slacks
- **Clock considerations**
- Analyses performed
  - Late mode ("max mode")
  - Early mode ("min mode")

Pre-Clock-Discussion

- **Paths**
  - Collection of a series of arcs
  - Created from recursive search of AT and RAT propagation of the timing graph
    - Non-extreme ATs and RATs are pruned
- **Arcs**
  - Delay and slew are attributes
  - Delays and slews are calculated for wires and gates
- **Nodes**
  - Contain ATs, RATs, and slack per edge (rise and fall)

Clocks are Special Signals

- **Clocks are propagated from the primary inputs defined with clock waveforms**
  - Clock "flag" propagates through leaf cells unless the entering pin is determined to be a clock pin
- **Implicate RATs**
  - Every gate where clock meets data, a timing check is performed
  - Combinatorial gates with a clock and data inputs normally become clock gates – their output is also a clock
- ~~**Much of the clock network is not seen by the static timer (e.g., PLL, major trunks) until after PNR**~~
  - ~~**Detailed circuit analysis will provide set of arrival times, slews, uncertainty as "givens" to the logic design**~~
  - ~~**Clock team would provide final clock tree implementation**~~

Clock Phases in STA

- **Each timing event is associated with a clock phase**
  - Clock phase(s) of input boundary signals are set by the user in arrival constraints
  - A clock phase is propagated with each timing event
  - Clock phases are used to perform "cycle accounting": all timing events are "modulo" the cycle time of the clock, accounting is based on phase time.
  - Data pins may contain arrival times from more than one clock domain.

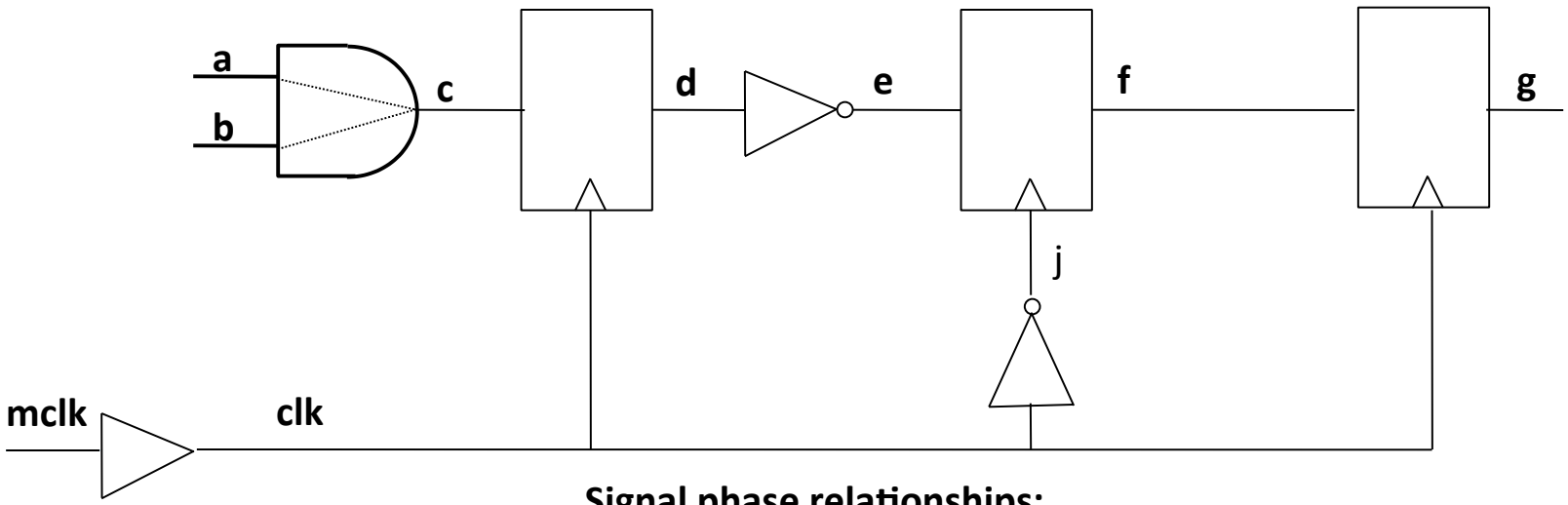
Clock Phase Propagation

- **Combinatorial non-clock arcs**
  - Input phases propagated to output
  - Different phases do not get combined

- **Sequential arcs**
  - Output phase determined by timing event at clock pin
  - How the timing event affects the output phase is built into the model
  - Event time is adjusted as necessary when changing phase, based on cycle accounting rules

Example: Clock Phase Propagation

master clock waveform: mclk

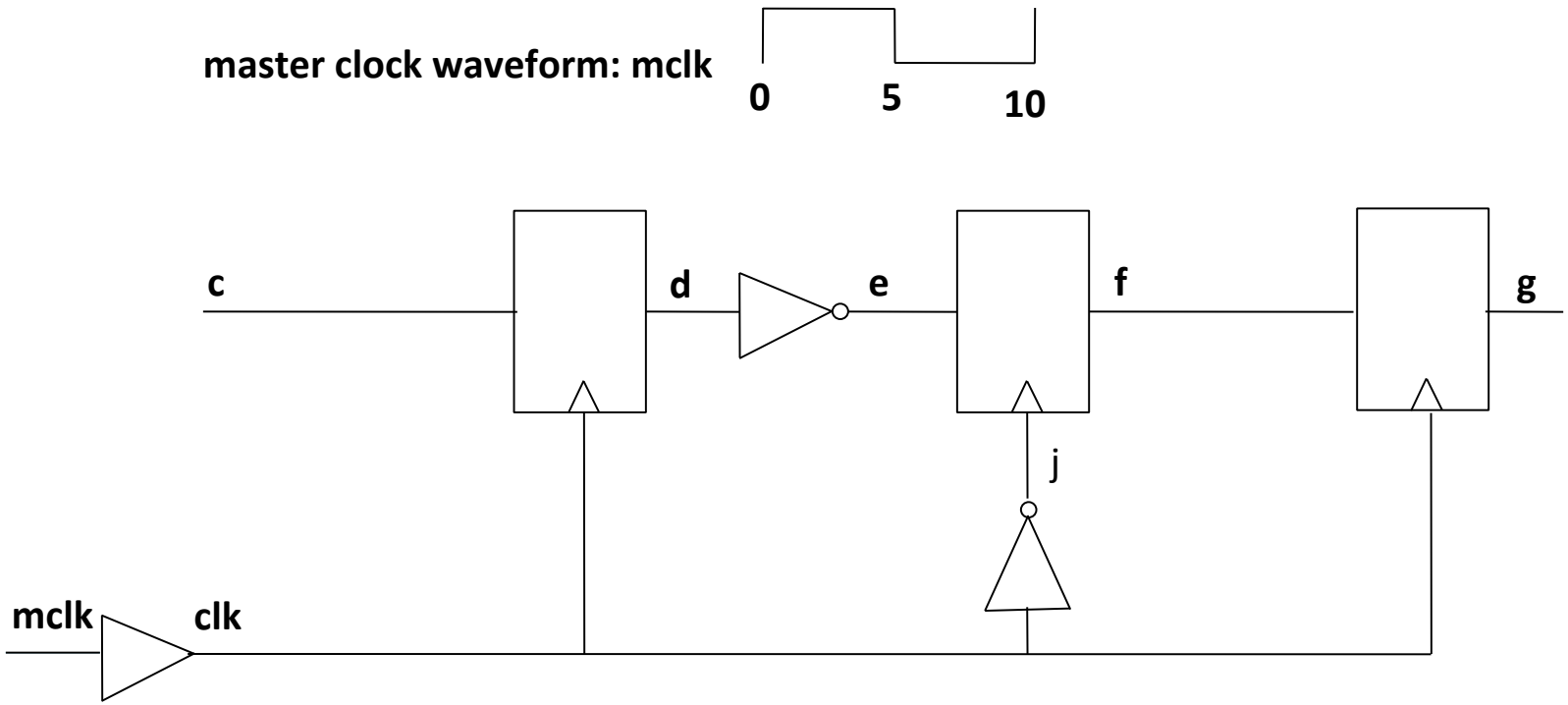


Boundary constraints:
clk is clock port for mclk
a arrival phase mclk ⇑
b arrival phase mclk ⇓

Signal phase relationships:

a ⇑, a ⇓ :	mclk ⇑
b ⇑, b ⇓ :	mclk ⇓
clk ⇑ :	mclk ⇑
clk ⇓ :	mclk ⇓
c ⇑ :	mclk ⇑, mclk ⇓
c ⇓ :	mclk ⇑, mclk ⇓
d ⇑, d ⇓ :	mclk ⇑
e ⇑, e ⇓ :	mclk ⇑
f ⇑, f ⇓ :	mclk ⇓
g ⇑, g ⇓ :	mclk ⇑
j ⇑ :	mclk ⇓
j ⇓ :	mclk ⇑

Cycle Adjustments for Tests

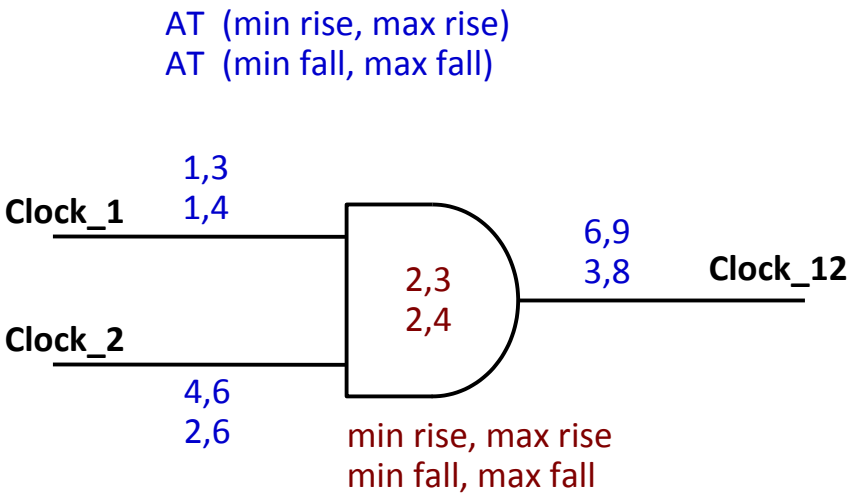


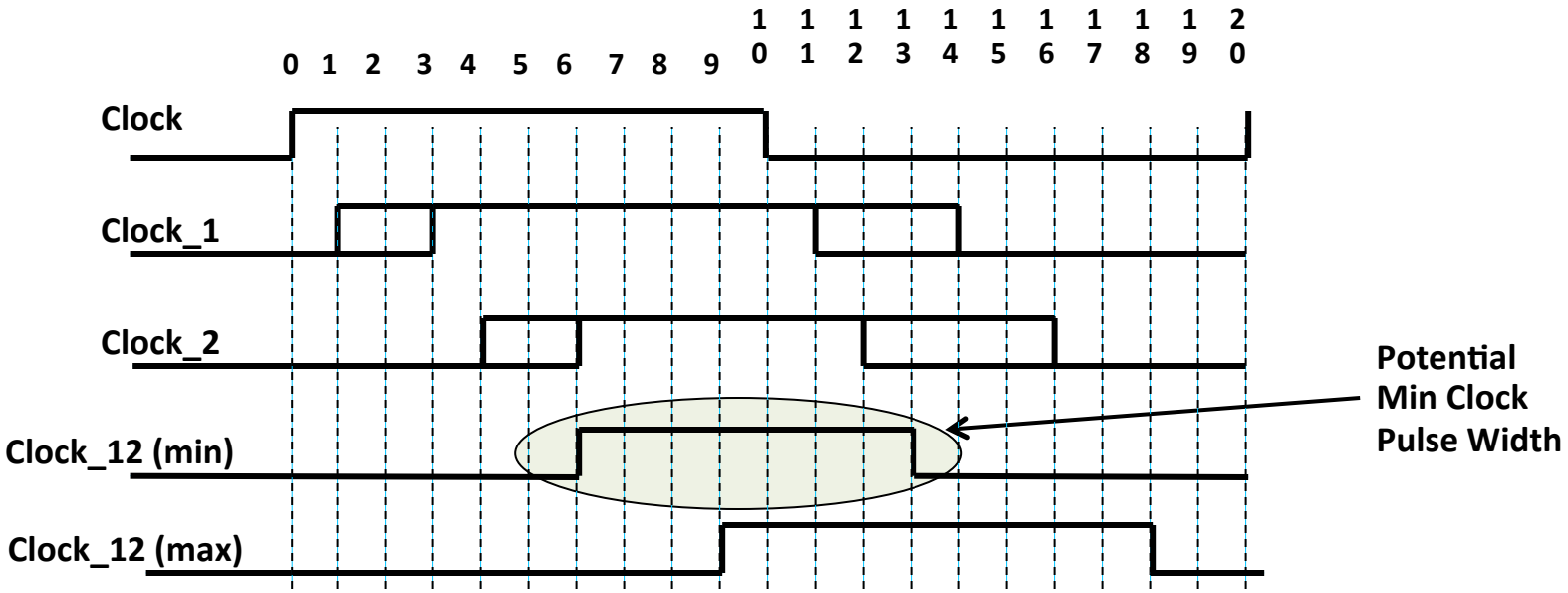
Boundary constraints:
clk is clock port for mclk
c arrival phase mclk ⇑

Setup check cycle adjusts:
c: mclk ⇑ against mclk ⇑ : +1
e: mclk ⇑ against mclk ⇓ : +1/2
f: mclk ⇓ against mclk ⇑ : +1/2

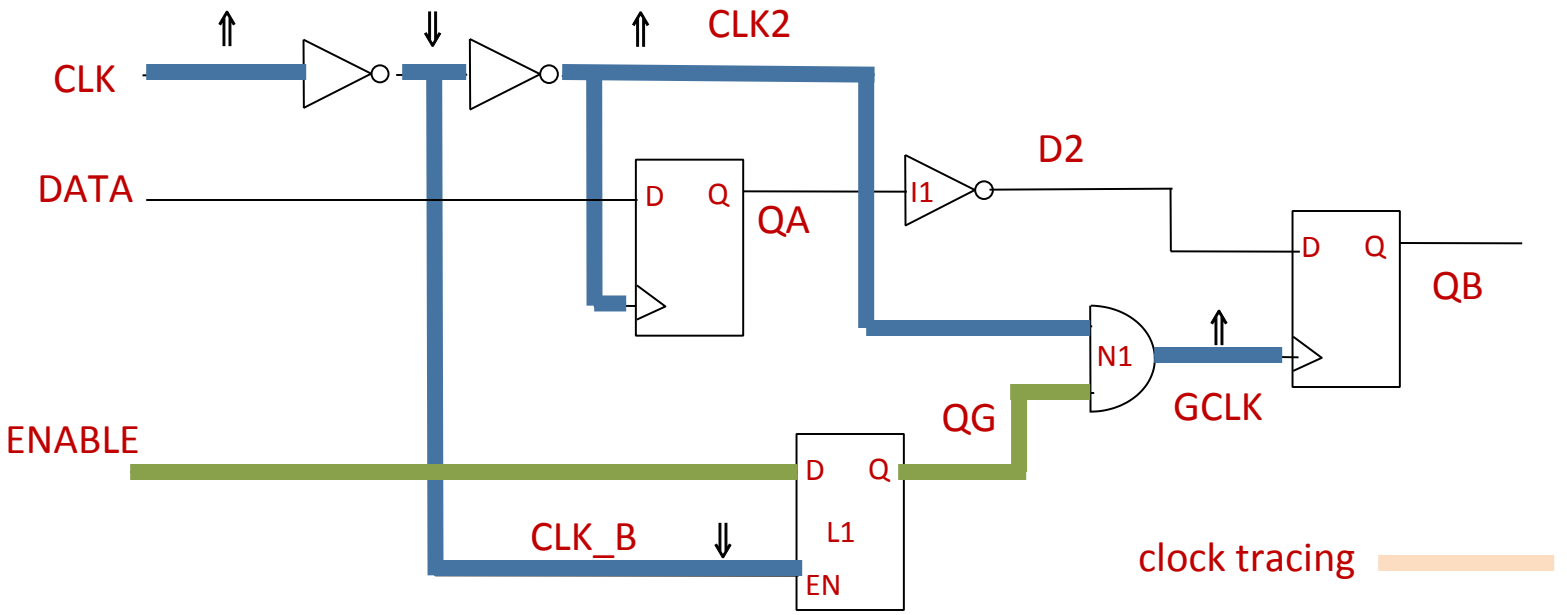
Hold check cycle adjusts:
c: mclk ⇑ against mclk ⇑ : 0
e: mclk ⇑ against mclk ⇓ : 0
f: mclk ⇓ against mclk ⇑ : 0

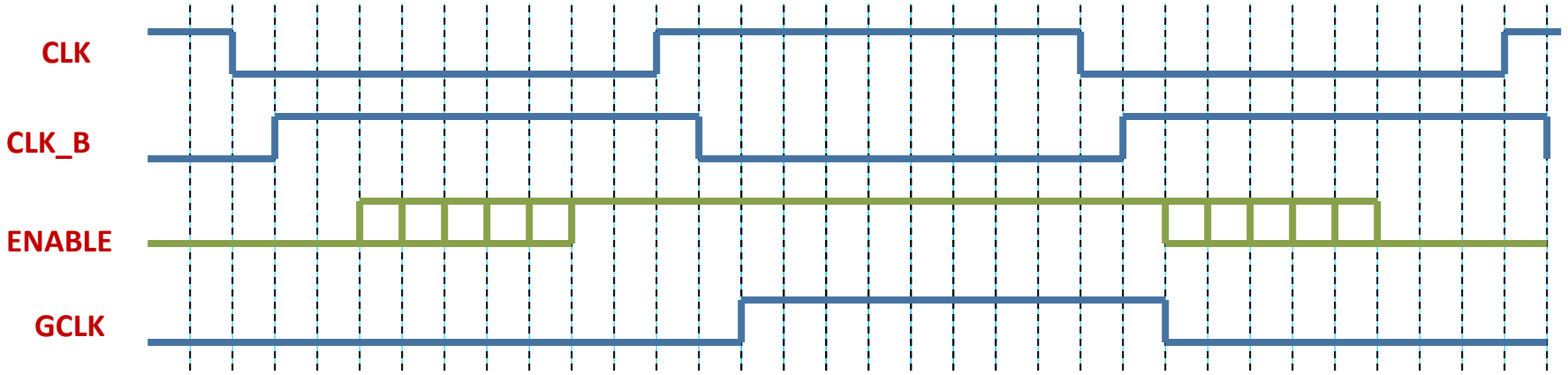
Propagation of Clocks thru Combinational Logic





Gated Clock Propagation





Summary of Clock Considerations

- Presence of a clock means events must occur in a particular order
- Most clock nets are automatically identified through clock tracing
  - Clocks generated by state machines usually need to be manually identified
- When these clocks arrive at gates with data, timing checks are performed to verify proper data arrival time
- Combine clocks at your own peril
- Each arrival time entry of a node has a clock phase associated with it
- In the slack calculation the RAT is determined in part by the clocks arrival at the test point

Agenda

- Basics of Timing Analysis
- The Timing Graph – the Timer's internal map
- Timing graph attributes: ATs, RATs, and Slacks
- Clock considerations
- **Analyses performed**
  - **Late mode ("max mode")**
  - **Early mode ("min mode")**

Late Mode (Max Mode) Analysis

- **Slow path analysis**
  - Verify signals don't arrive too late
    - Compares latest possible data arrival versus earliest required time
    - Answers: What is the max frequency of the design?
  - "set up time" or "slow path" or "late mode" or "max mode analysis"
  - Often conducted at typical process corner under worst PVT conditions for yield considerations

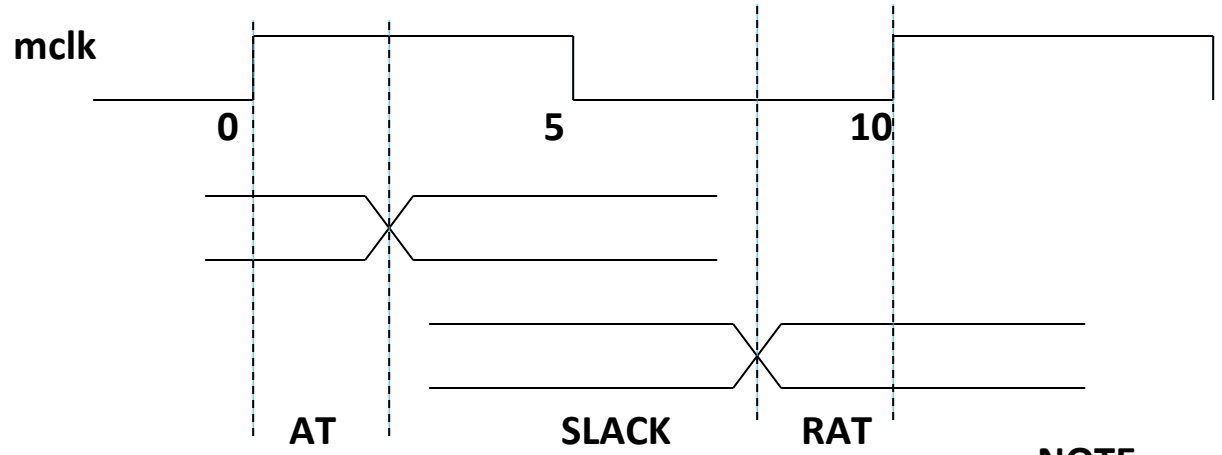
- **Validate registration**
  - Checks that data are set up before the required time
    - Usually the required time is determined by a clock
  - Launch and capture events are derived from different real time edges of the master clock

NOTE: slowing cycle time remedies these violations

Late Mode Slack Calculation

- **Late mode slack = RAT - AT**
  - AT is the latest time the data event can occur
  - RAT is the earliest requirement for the event to occur
    - RAT = AT(early reference) - u - su + adjust (if the requirement is clock derived)
- **Required element setup time (su)**
  - determined from spice simulations (more discussion later)
- **Clock uncertainty (u)**
  - takes into consideration clock skew and other effects that are not being simulated during static timing analysis
- **Cycle time adjustments**
  - adjusts are performed by the static timer in order to relate the correct data with the appropriate clock event; static timers analyse in "modulo" master clock cycle mode
    - e.g,. latches: when the same master clock event causes both the launch of data and its capture, then the adjust = 1 clock cycle
    - e.g., clock gates: when launch happens after capture, adjust = 1 clock cycle

Late Mode Analysis (Full cycle Path)



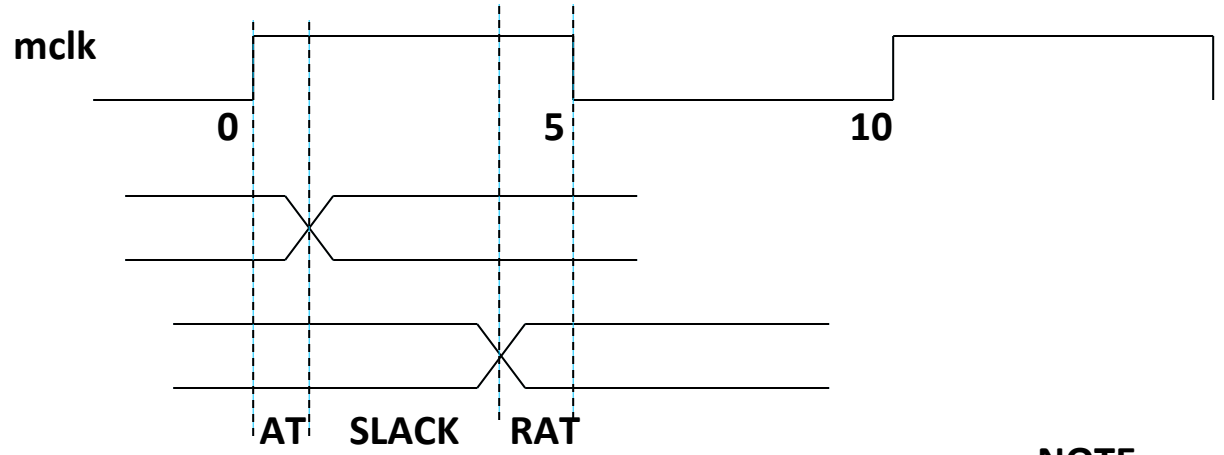
NOTE:
RAT is a negative number WRT to mclk ⇑

SLACK = RAT – AT + CYCLE_ADJUST

Example:
AT = 2
RAT = -3

SLACK = -3 -2 + 10 = 5

Late Mode Analysis (Half cycle Path)



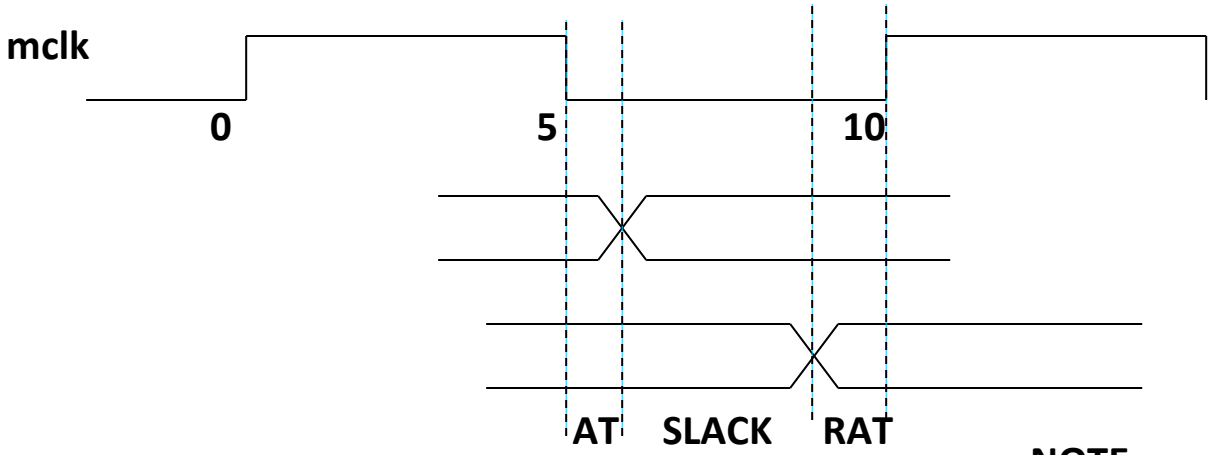
SLACK = RAT – AT + CYCLE_ADJUST/2

Example:
AT = 1
RAT = -1

SLACK = -1 -1 + 5 = 3

NOTE:
AT is a positive number WRT to mck ⇑
RAT is a negative number WRT to mclk ⇓

Late Mode Analysis (Half cycle Path)



NOTE:
AT is a positive number WRT to mck ⇓
RAT is a negative number WRT to mclk ⇑

SLACK = RAT – AT + CYCLE_ADJUST/2

Example:
AT = 1
RAT = -1

SLACK = -1 -1 + 5 = 3

Early Mode (Min Mode) Analysis

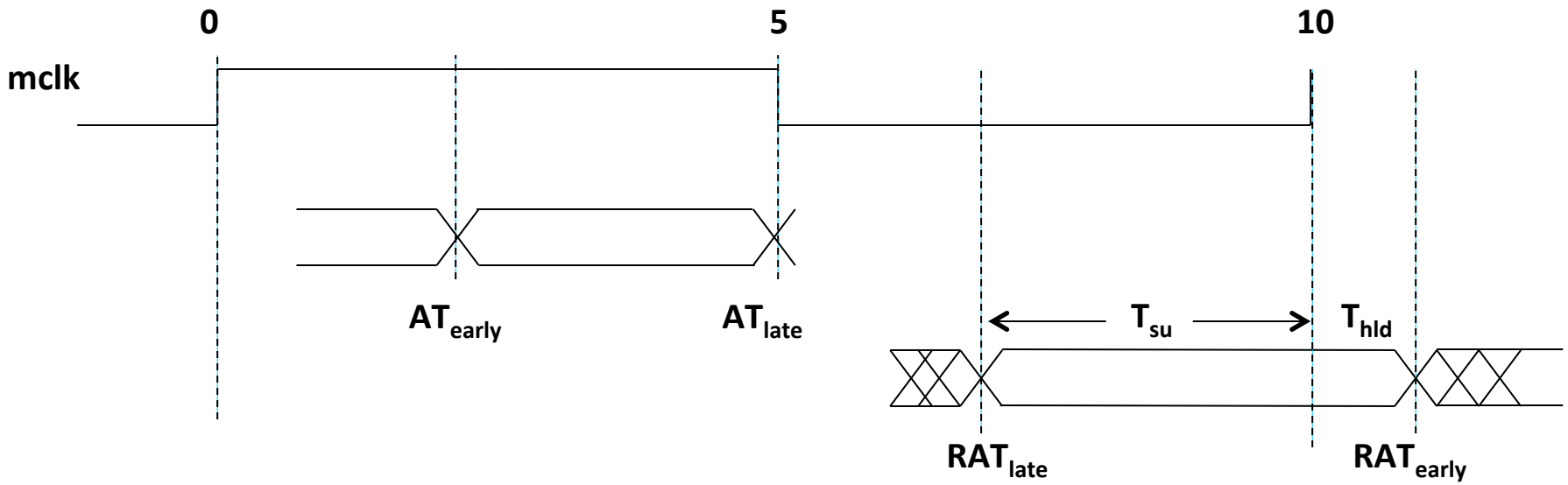
- **Fast path(s) analysis**
  - Verify signals don't arrive too early
    - Compares earliest possible data arrival against latest required time
    - Identifies potential race hazards in a design, usually between clock and data signals.
  - "Hold time" or "Fast Path" or "Early Mode" or "min mode analysis"
  - Often conducted at best process corner under best PVT conditions
- **Validate registration**
  - Checks that data held long enough at destination circuit
  - Generally same-edge test
    - Master clock edge causes both capture clock to close and new data to launch

NOTE: cannot slow cycle time to remedy these violations

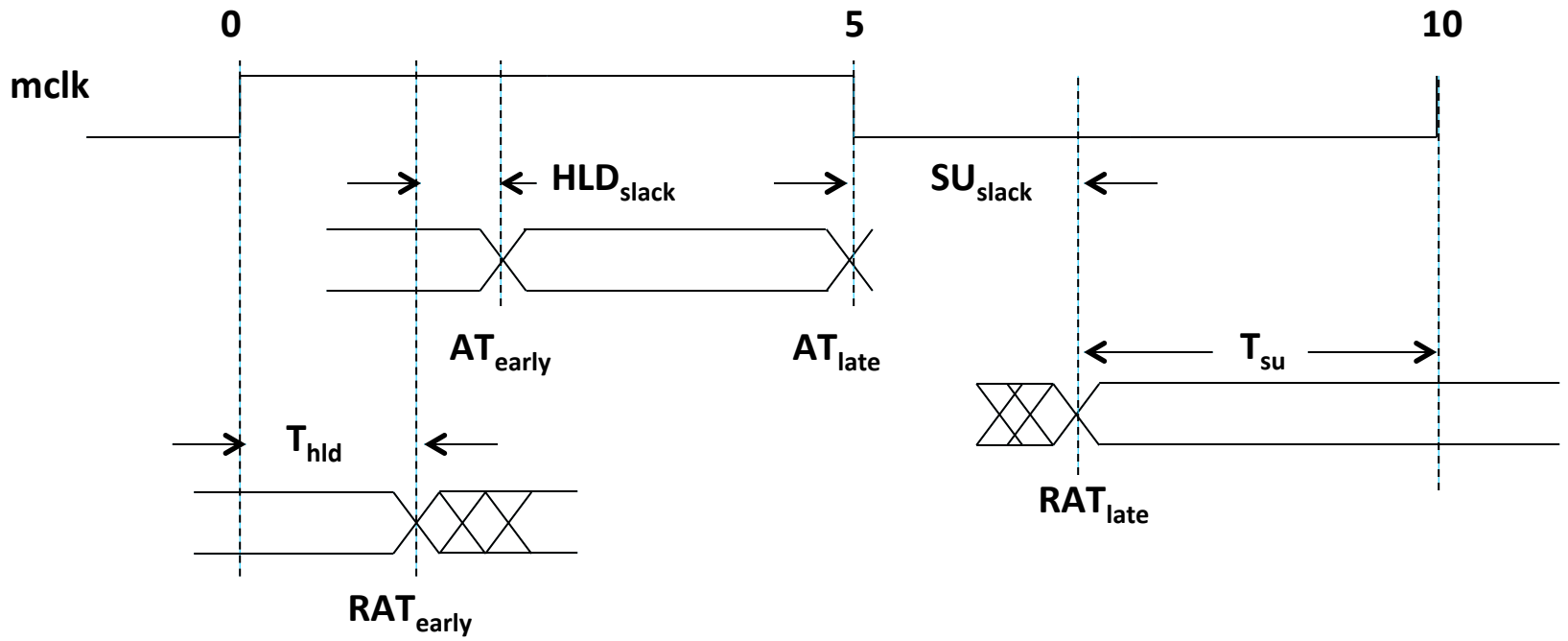
Early Mode Slack Calculation

- **Early mode slack = AT – RAT**
  - AT is the earliest time the event can occur
  - RAT is the latest requirement for the event to occur
    - RAT = AT(late reference) + u + h + adjust (if the requirement is clock derived)
    - e.g, reference edge for latch is same as in late mode; clock gate must be checked such that data is held beyond the de-asserting clock edge (falling clock edge to a nand/and)
- **Required element hold time (h)**
  - determined from spice simulations (refer to FF lecture)
- **Clock uncertainty (u)**
  - same idea as in late mode except it will probably be gauged against a different design point (faster process and conditions)
- **Cycle time adjustments for hold slack**
  - again, relate correct data with appropriate clock event
    - e.g., latches and clock gates: when the same master clock event causes both the launch of data and the reference checking edge, then the adjust = 0 clock cycle
  - Must be done carefully – avoid "timing escapes" for odd/wrong designs

Early & Late Mode Analysis (Full cycle Path)



Early & Late Mode Analysis (Full cycle Path)



$HLD_{SLACK} = AT_{early} - RAT_{early}$

Example:
$AT_{early} = 3$
$RAT_{early} = 2$

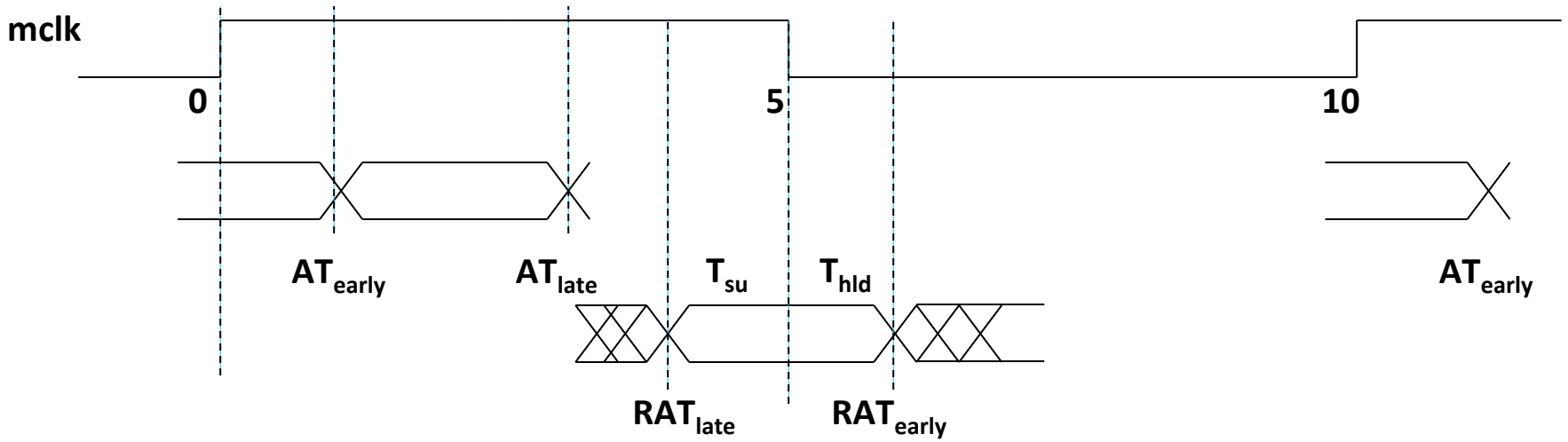
$HLD_{SLACK} = 3 - 2 = 1$

$SU_{SLACK} = RAT_{late} - AT_{late} + CYCLE\_ADJUST$

Example:
$AT_{late} = 5$
$RAT_{late} = -3$

$SU_{SLACK} = -3 - 5 + 10 = 2$

Early & Late Mode Analysis (Half cycle Path)



$SU_{SLACK} = RAT_{late} - AT_{late} + CYCLE\_ADJUST/2$

Example:
$AT_{late} = 3$
$RAT_{late} = -1$

$SU_{SLACK} = -1 -3 + 5 = 1$

$HLD_{SLACK} = AT_{early} - RAT_{early}$

Example:
$AT_{early} = 2$
$RAT_{early} = 1$

$HLD_{SLACK} = 2 -1 = 1$

GLOSSARY

Glossary

arc – a path between pins or nodes of a timing graph that illustrates a signal can pass arrival time and slew from the input pin/node to the output (considering polarity); represents delay/slew of logic blocks or wires between pins of logic blocks

AT – arrival time; the time a signal arrives at a node

check – a test between two signals to guarantee event order; setup checks test that the signal arrives before the reference does, hold checks test that the signal arrives after the reference

clipping – an arrival time adjustment by an amount that would barely allow the failing upstream check to pass in order to judge the arrival times downstream

clock – signal which defines synchronous behavior; requirement that defines cycle time and required arrival times at timing elements

clock domain – see "phases"

clock gate – non-transparent timing element which imposes a half cycle path; setup tested against asserting clock edge, hold against de-asserting edge

clock tracing – operation of the static timer to identify all block pins that are clocks

controlled arc – a timing arc that will not propagate data unless an enable signal is in the proper state

Glossary (cont.)

cycle accounting – adjusting the cumulative arrival time to maintain events within the cycle modulus

cycle adjusts – cycle modulus changes to the required arrival time in order to relate the correct data to the clock event that it must be checked against

domino gate – transparent timing element

data – a signal that is not a clock; a signal that changes once per cycle unless it is a dynamic signal (e.g., domino) which is affected by both edges of the clock

early mode – timing mode to check that shortest paths meet proper registration/don't arrive too early; use earliest arriving data and compare to latest required arrival time

flip flop – edge triggered (non-transparent) timing element

hold time – timing check to assure new data didn't change until after reference closed; sometimes refers to the margin associated with particular timing elements to assure the slack calculation takes all necessary circuit issues into consideration

latch – transparent timing element

Glossary (cont.)

late mode – mode to check that longest paths meet timing goals; use latest arriving data and compare to earliest required arrival time

near-domino gate – transparent timing element

phases – clock and data signal attribute which indicates the relationship of the signal compare to the master clock (signal from which its timing context is derived); sometimes referred to as signal's clock domain; phases are used in order to provide the static timer the ability to perform cycle adjusts

RAT – required arrival time; defines the acceptable boundaries of a signal's min and max arrival time and considers the reference events, circuit mechanics, and system tolerances

setup time – timing check to assure data arrives before reference closes; sometimes refers to the margin associated with particular timing elements to assure the slack calculation takes all necessary circuit issues into consideration

skew – difference in same clock event arrival time at different pins throughout the design; usually not analyzed by static timer

slack – residual time of the difference between when an event actually occurs versus when it is required to occur; a negative value means that the order of the two events was wrong.

SSTA – Statistical static timing analysis

Glossary (cont.)

static timing – exhaustive method of measuring a design's timing robustness by building a timing graph of the design, providing signal arrival times, propagating these and identifying critical paths

timing elements – logical context defining arcs and checks among three points in a timing graph; examples: latch, flip flop, clock gate

timing graph – a collection of arcs and checks which represents the timing behavior of a logic design

transparent – data propagation through a controlled arc when the controlling signal is enabled

uncertainty – safety margin included in the slack calculation to account for clock skew and other variables affecting clock edge events

Timing Loop