Prof. Kaustav Banerjee
Electrical and Computer Engineering
E-mail: kaustav@ece.ucsb.edu
Timing Issues
Flip-Flop – Based Timing

Representation after
M. Horowitz, VLSI Circuits 1996.
Flip-flops are used only with static logic
Latch timing

When data arrives to transparent latch:
Latch is a ‘soft’ barrier

When data arrives to closed latch:
Data has to be ‘re-launched’
Single-Phase Clock with Latches

\[ \phi \]

Latch

Logic

\[ T_{skl} \]

\[ T_{skl} \]

\[ T_{skt} \]

\[ T_{skt} \]

Clk

\[ PW \]

\[ P \]
Latch-Based Design

L1 latch is transparent when $\phi = 0$

L2 latch is transparent when $\phi = 1$
Slack-borrowing

Slack-passing results from level sensitivity of latches and takes place if: $T_{CLK} < t_{pd,A} + t_{pd,B}$ and the logic functions correctly.

Max. time that can be borrowed = $T_{CLK}/2$

Max. logic cycle delay = $1.5XT_{CLK}$

“a” valid well before edge 2, since previous block did not use its entire time allotment, producing slack time denoted by the shaded area. Now CLB_B starts computing using the slack provided by CLB_A, and completes before its allocated time (edge 4) and passes a small slack to the next cycle.
Latch-Based Timing

L1 latch

L2 latch

Can tolerate skew!

L1 latch

L2 latch

Static logic

Skew

Long path

Short path

$\phi = 1$

$\phi = 0$
Self-timed and Asynchronous Design

Functions of clock in synchronous design

1) Acts as completion signal: ensures that physical timing constraints are met
2) Ensures the correct ordering of events

Problems: skew, jitter, noise, system performance determined by slowest segment in pipeline
Alternatives to Synchronous Design

**Truly asynchronous design**
1) Completion is ensured by careful timing analysis
2) Ordering of events is implicit in logic

**Self-timed design**
1) Completion ensured by completion signal
2) Ordering imposed by handshaking protocol
The Handshaking protocol is used to communicate between adjacent combinational function that a particular computation has been completed.
**Self-Timed Designs**

Features:
- Separates the physical and logical ordering functions implied in circuit timing
- The completion signal *Done* ensures that physical timing constraints are met
- Logical ordering of the operation is ensured by the *ack-req* scheme or the *HS* protocol
- Choice of protocol significantly impacts performance and robustness

Advantages:
- Timing signals generated locally—avoids overheads associated with distributing high-speed clocks
- Separating physical and logical ordering mechanisms results in potential increase in performance. In syn. systems the $T_{\text{CLK}}$ must be large enough to account for the worst case delay
- Robust against temperature variations

Cons:
- Substantial circuit-level overhead
Completion Signal Generation: Dual Rail Coding

Requires introduction of redundancy in the data representation in order to signal that a particular bit is in either a transition or a steady-state mode.

<table>
<thead>
<tr>
<th>B</th>
<th>B0</th>
<th>B1</th>
</tr>
</thead>
<tbody>
<tr>
<td>in transition (or reset)</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>illegal</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Two bits B0 and B1 are used to represent a single data bit B.

Using Redundant Signal Encoding

Key idea in all redundant signal representations is to capture the transition state to indicate that the circuit is in evaluation mode and the output data are not valid.
Dual Rail Coding Using Dynamic DCVSL
Example: A Self-Timed Adder

(a) Differential carry generation

(b) Completion signal
Completion Signal Generation

Widely used to generate the internal timing in memories
Completion Signal Using Current Sensing

- Inputs
- Start
- Input Register
- Static CMOS Logic
- Current Sensor
- Min Delay Generator
- Output

- $V_{DD}$
- $GND_{sense}$
- $A$
- $B$
- $t_{delay}$
- $t_{overlap}$
- $t_{MDG}$
- $t_{pd-NOR}$
- $valid$

- Start
- A
- B
- Done
- Output

Lecture 14, ECE 225
Kaustav Banerjee
Hand-Shaking Protocol

(a) Sender-receiver configuration

(b) Timing diagram

Two Phase Handshake
Event Logic – The Muller-C Element

(a) Schematic

(b) Truth table

A | B | $F_{n+1}$
---|---|---
0 | 0 | 0
0 | 1 | $F_n$
1 | 0 | $F_n$
1 | 1 | 1

(a) Logic

(b) Majority Function

(c) Dynamic
2-Phase Handshake Protocol

Advantage: FAST - minimal # of signaling events (important for global interconnect)

Disadvantage: edge-sensitive, has state
Example: Self-timed FIFO

All 1s or 0s -> pipeline empty
Alternating 1s and 0s -> pipeline full
2-Phase Protocol
Example

From [Horowitz]
Example
Example
Example
4-Phase Handshake Protocol

Also known as RTZ

Slower, but unambiguous
4-Phase Handshake Protocol

Implementation using Muller-C elements
Self-Resetting Logic

Precharged Logic Block (L1)

Precharged Logic Block (L2)

Precharged Logic Block (L3)

Completion detection (L1)

Completion detection (L2)

Completion detection (L3)

Post-charge logic

$V_{DD}$

A

B

C

int

out

Lecture 14, ECE 225

Kaustav Banerjee
Clock-Delayed Domino

CLK1 → GND → CLK2 (to next stage) → Q1 (also D2)

D1 → Pulldown Network → V_DD
Asynchronous-Synchronous Interface

Asynchronous system → Synchronization → Synchronous system

$f_{in}$ → $f_{CLK}$
Synchronizers and Arbiters

- **Arbiter**: Circuit to decide which of 2 events occurred first
- **Synchronizer**: Arbiter with clock $\phi$ as one of the inputs
- **Problem**: Circuit HAS to make a decision in limited time - which decision is not important
- **Caveat**: It is impossible to ensure correct operation
- But, we can decrease the error probability at the expense of delay
A Simple Synchronizer

- Data sampled on rising edge of the clock
- Latch will eventually resolve the signal value, but ... this might take infinite time!
Synchronizer: Output Trajectories

Single-pole model for a flip-flop

\[ v(t) = V_{MS} + (v(0) - V_{MS})e^{t/\tau} \]
Mean Time to Failure

\[ N_{sync}(0) = \frac{P_{init}}{T_\phi} = \frac{\left(\frac{V_{IH} - V_{IL}}{V_{swing}}\right) t_r}{T_{signal}} \frac{1}{T_\phi} \]

\[ N_{sync}(T) = \frac{P_{init} e^{-T/\tau}}{T_\phi} = \frac{(V_{IH} - V_{IL}) e^{-T/\tau}}{V_{swing}} \frac{t_r}{T_{signal} T_\phi} \]
Example

\[ T_f = 10 \text{ nsec} = T \]
\[ T_{signal} = 50 \text{ nsec} \]
\[ t_r = 1 \text{ nsec} \]
\[ t = 310 \text{ psec} \]
\[ V_{IH} - V_{IL} = 1 \text{ V} \quad (V_{DD} = 5 \text{ V}) \]

\[ N(T) = 3.9 \times 10^{-9} \text{ errors/sec} \]
\[ MTF (T) = 2.6 \times 10^8 \text{ sec} = 8.3 \text{ years} \]
\[ MTF (0) = 2.5 \text{ \mu sec} \]
Influence of Noise

Uniform distribution around VM

Initial Distribution

$p(v)$

$0 \quad V_{IL} \quad V_{IH} \quad T$

Still Uniform

Logarithmic reduction

Low amplitude noise does not influence synchronization behavior
Typical Synchronizers

2 phase clocking circuit

Using delay line
Cascaded Synchronizers Reduce MTF

In

\[ \text{Sync} \rightarrow O_1 \rightarrow \text{Sync} \rightarrow O_2 \rightarrow \text{Sync} \rightarrow \text{Out} \]

\[ \phi \]

\[ \text{Sync} \]

\[ \text{Sync} \]

\[ \text{Sync} \]
Arbiters

(a) Schematic symbol

(b) Implementation

(c) Timing diagram

V_T gap

metastable
PLL-Based Synchronization

\[ f_{\text{system}} = N \times f_{\text{crystal}} \]

Crystal Oscillator

Divider

PLL

Chip 1

Data

reference clock

Chip 2

Digital System

PLL

Clock Buffer

Data

Digital System

Data

Reference clock

Data
PLL Block Diagram

Reference clock → Phase detector → Charge pump → Loop filter → VCO

Local clock → Divide by \( N \)

Up → Down

\( V_{cont} \)

System Clock
Phase Detector

Output before filtering

Output (Low pass filtered)

Transfer characteristic

Lecture 14, ECE 225

Kaustav Banerjee
Phase-Frequency Detector

(a) schematic

(b) state transition diagram

(c) Timing waveforms
PFD Response to Frequency

A

B

UP

DN
PFD Phase Transfer Characteristic

Average (UP-DN) vs. phase error (deg)
Charge Pump

\[ V_{DD} \]

- UP
- DN

To VCO Control Input
PLL Simulation

Control Voltage (V)

Time (μs)

ref

div

vco

ref

div

vco
Clock Generation using DLLs

Delay-Locked Loop (Delay Line Based)

Phase-Det Charge Pump DL

Phase-Locked Loop (VCO-Based)

PD CP VCO ÷N

\[ f_{\text{REF}} \]

\[ f_0 \]
Delay Locked Loop

\[ F_{REF} \xrightarrow{\Delta PH} \text{Phase detect} \xrightarrow{U} \text{Charge pump} \xrightarrow{D} C \xrightarrow{V_{CTRL}} \text{VCDL} \xrightarrow{V_{CTRL}} F_O \]

(a)

(b)

(c)
DLL-Based Clock Distribution

GLOBAL CLK → VCDL → CP/LF → Phase Detector → VCDL → CP/LF → Phase Detector → Digital Circuit...

Digital Circuit