# ECE 720 – ESL & Physical Design

## Lecture 22: Signal Integrity

Spring 2013

W. Rhett Davis
NC State University

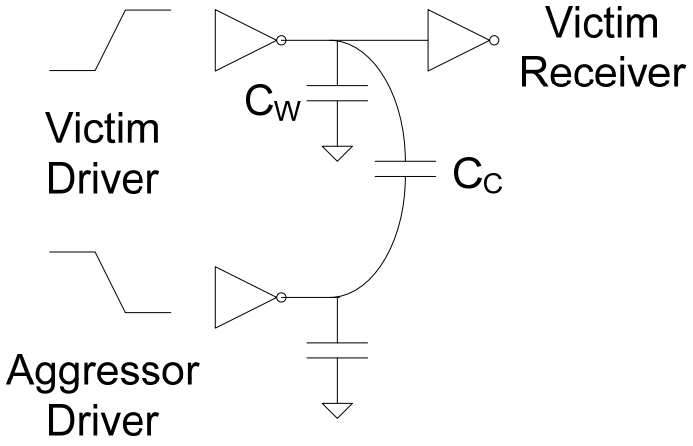
# Announcements

- Homework #7 Due Today
- Homework #8 Due in 9 days

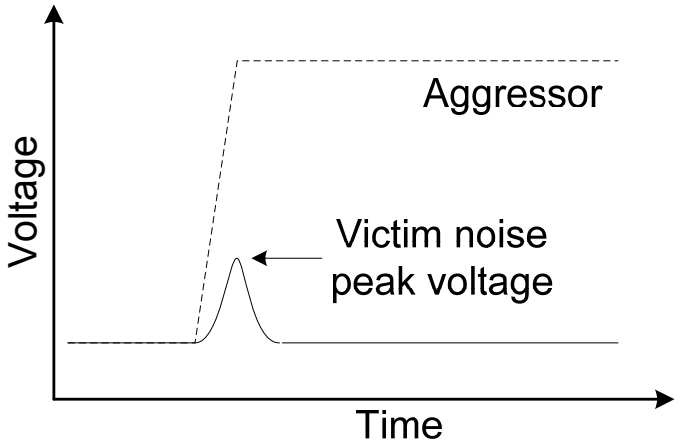
# Today's Lecture

- Basic Crosstalk Analysis
- Functional Noise Analysis
- Delay Noise Analysis
- Analyzing Signal Integrity in PrimeTime
- Reading SPEF Files

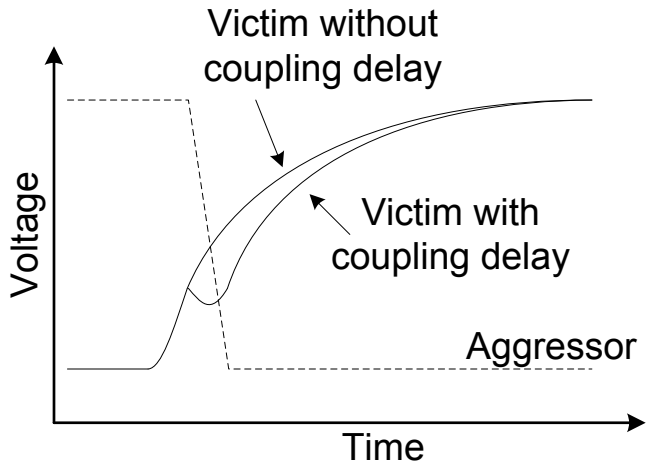
# Cross-Talk Noise




- Functional or Glitch Noise



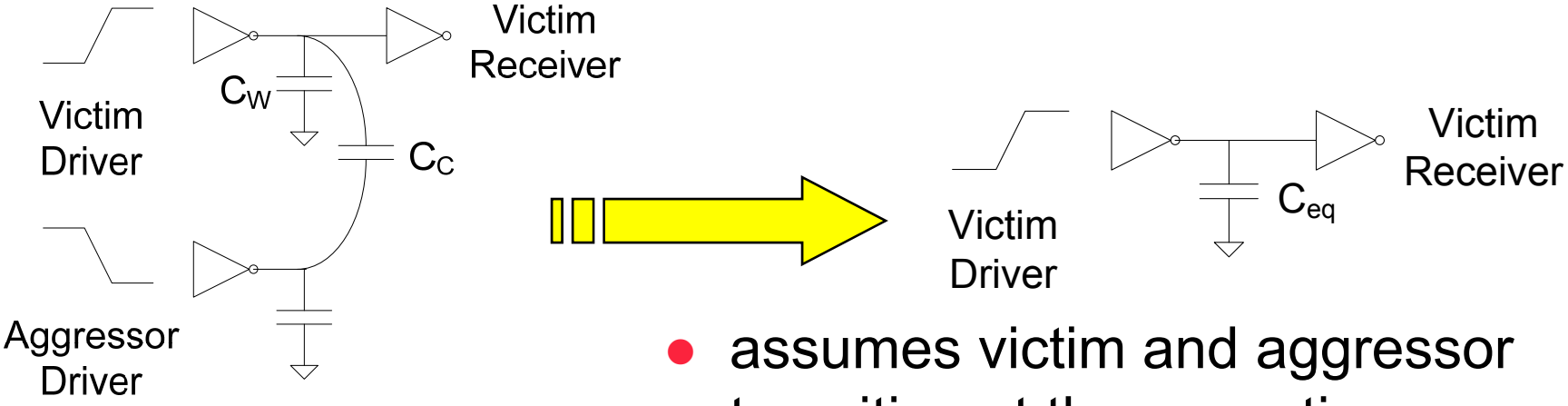

- Delay Noise



# Simplified Bump Analysis

- Use simple conservation of charge to calculate upper bound on bump height (Needed for HW#8)

# Simplified Delay-Noise Model




$$C_{eq} = C_W + (1-\beta)C_C$$

$$\beta = \frac{\Delta V_A}{\Delta V_V}$$

- assumes victim and aggressor transition at the same time
- $\Delta V_V$ is $V_{DD}/2$ when victim is rising, $-V_{DD}/2$ when falling
- $\Delta V_A$ is the voltage change of the aggressor node while the victim completes the first half of its swing ($-2 \le \beta \le 2$)

source: Zhaoran Yan, *et al*, Intl. Conf. on ASIC 2003

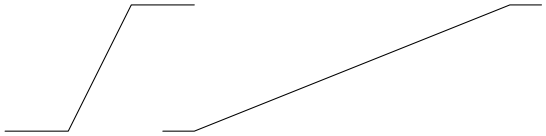
# What is $C_{eq}$ when...

victim aggressor

1)

$t_{rv} = t_{ra}$

2)

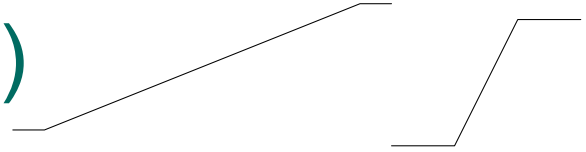


$t_{rv} << t_{ra}$

3)

$t_{rv} = t_{fa}$

4)

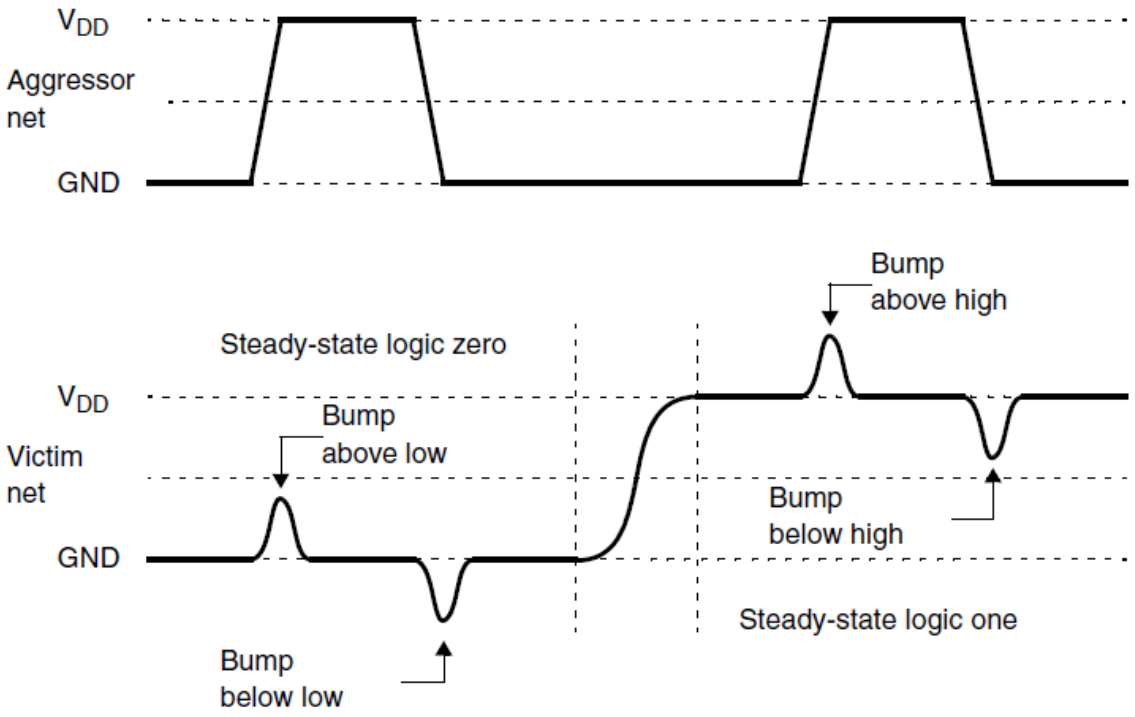


$t_{rv} >> t_{ra}$

# Today's Lecture

- Basic Crosstalk Analysis
- Functional Noise Analysis
- Delay Noise Analysis
- Analyzing Signal Integrity in PrimeTime
- Reading SPEF Files

# Terminology




Source: Synopsys PrimeTime SI User Guide

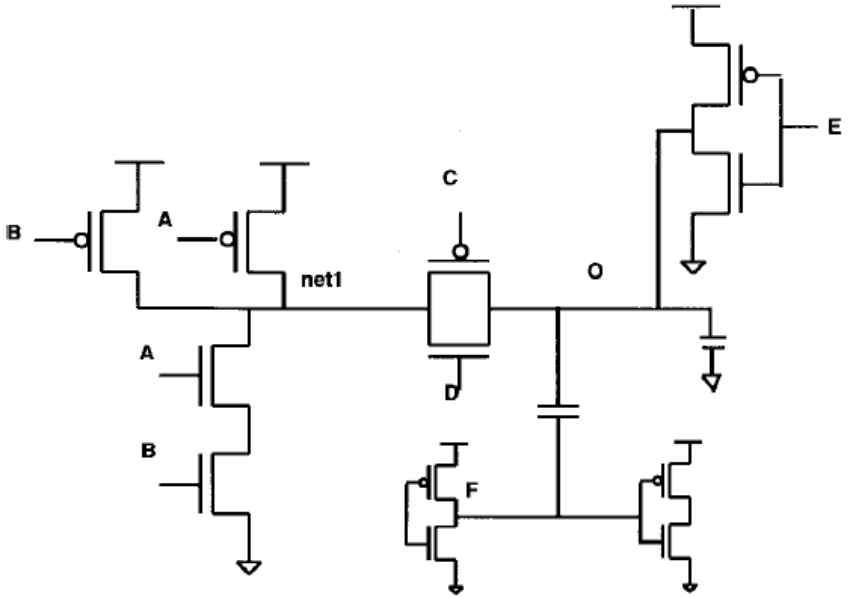
- Functional noise also called Peak or Bump Noise
- Primary distinction is what the node voltage "should" be
  - High (VH) or Low (VL)
- Secondary distinction is direction of deviation (above or below)

# Questions

- What kinds of failure can be caused by functional noise?
- Which of these possibilities are the most likely?
- We tend to care more about delay noise, but functional noise analysis provides the basis for delay noise analysis
- Shown here: Shepard/*Harmony*, basis for *CadMOS/CeltIC*, Acquired by *Cadence*, now a part of *Encounter*

# Node Sensitization

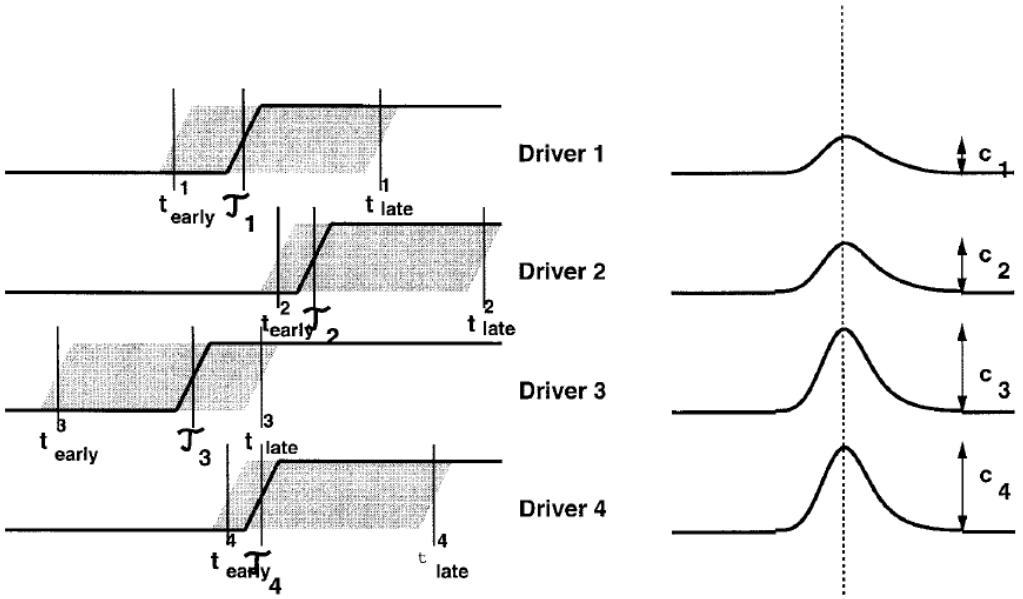
- First step in analysis is to determine how to set inputs on a gate to generate a noise peak
- Shown here: complete sensitization possibilities for "below high" noise at node O
  - 0/1 indicates steady value
  - rise/fall indicates transition
  - $V_L/V_H$ indicates a low or high noise peak at input




| sensitization | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 1 | 0 | fall |
| 2 | 0 | 1 | 0 | 1 | 0 | fall |
| 3 | 1 | 0 | 0 | 1 | 0 | fall |
| 4 | 1 | 1 | $V_L$ | 1 | 0 | fall |
| 5 | rise | 1 | 0 | 1 | 0 | fall |
| 6 | 0 | 0 | 1 | 0 | 0 | fall |
| 7 | $V_L$ | 1 | 0 | 1 | 0 | fall |
| 8 | 1 | $V_L$ | 0 | 1 | 0 | fall |

Source: Shepard et. al, "Harmony…" Trans. CAD 1999

# Switching Window Calculation




Source: Shepard et. al, "Harmony…" Trans. CAD 1999

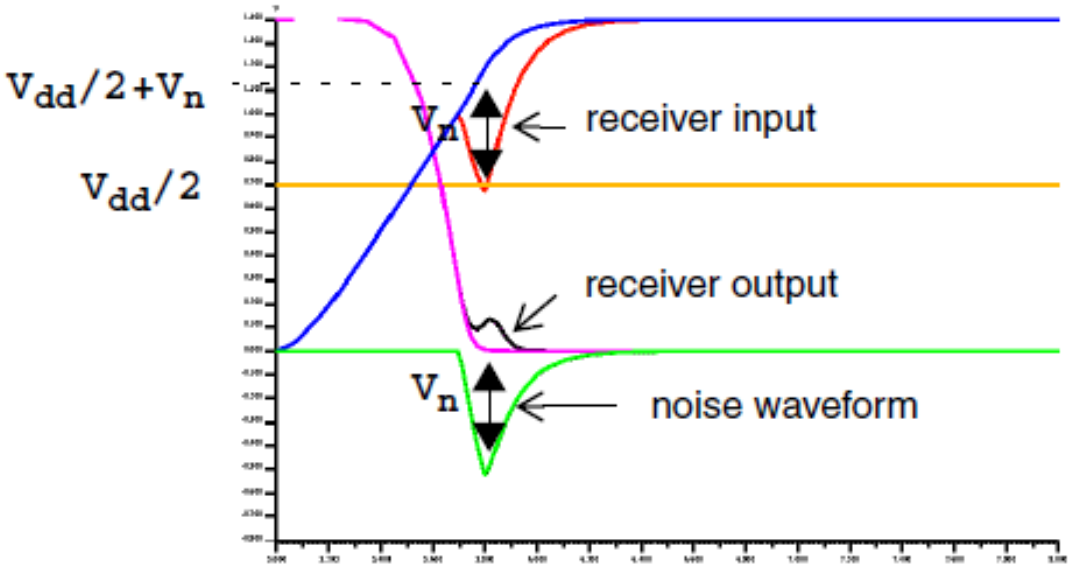
- Next step is to determine alignment of multiple aggressors
- Example
  - $t^1_{early}$ – earliest that driver 1 rise may occur
  - $t^1_{late}$ – latest that driver 1 rise may occur
  - $\tau_1$ – time that maximizes noise peak voltage

- Allows calclation of a "worst case" that isn't overly pessimistic
- Modified Nodal Analysis is then used to calculate noise-peaks

# Today's Lecture

- Basic Crosstalk Analysis
- Functional Noise Analysis
- Delay Noise Analysis
- Analyzing Signal Integrity in PrimeTime
- Reading SPEF Files

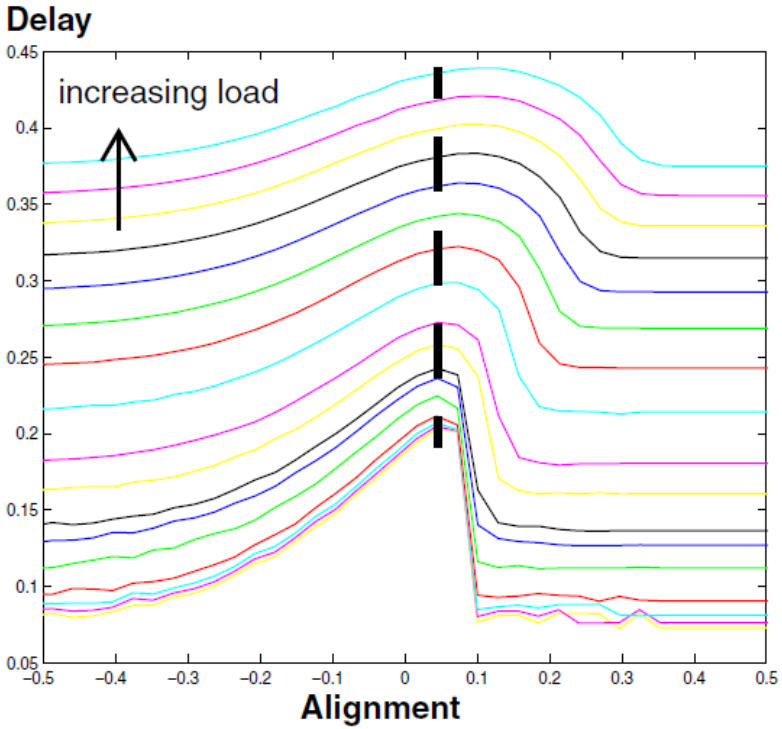
# Extension of Functional Noise

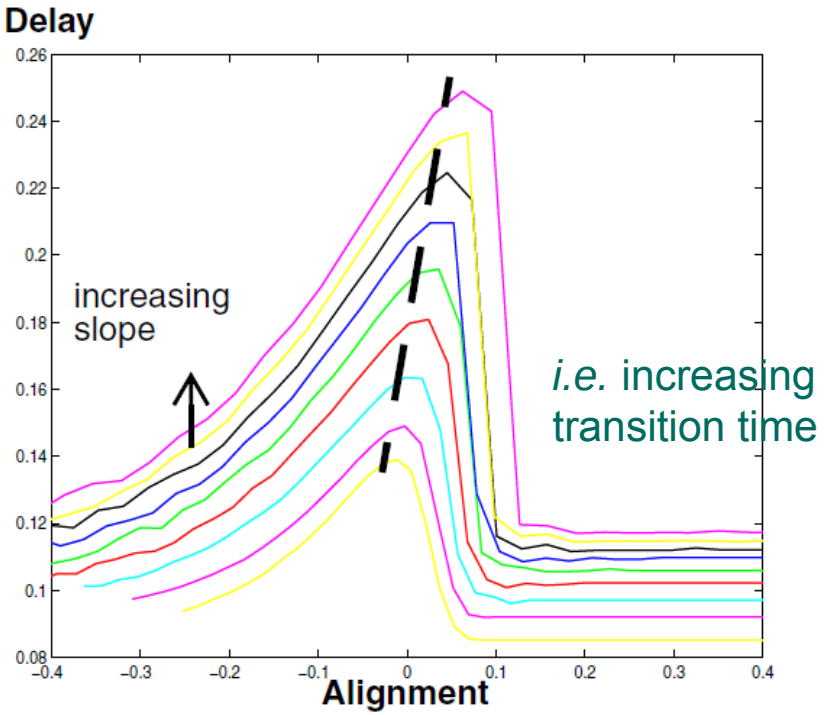



Source: Sirichotiyakul, Blaauw et. al, DAC 2001

- Once the worst-case peak noise is known for a non-switching victim ($V_n$), a simple extension can be used for a switching victim
  - Determine worst-case alignment along victim transition
  - Generally reached when noiseless victim transition reaches $V_{dd}/2 + V_n$
  - Below-low case with rising victim shown (7 other cases)

# Delay Sensitivity to Alignment



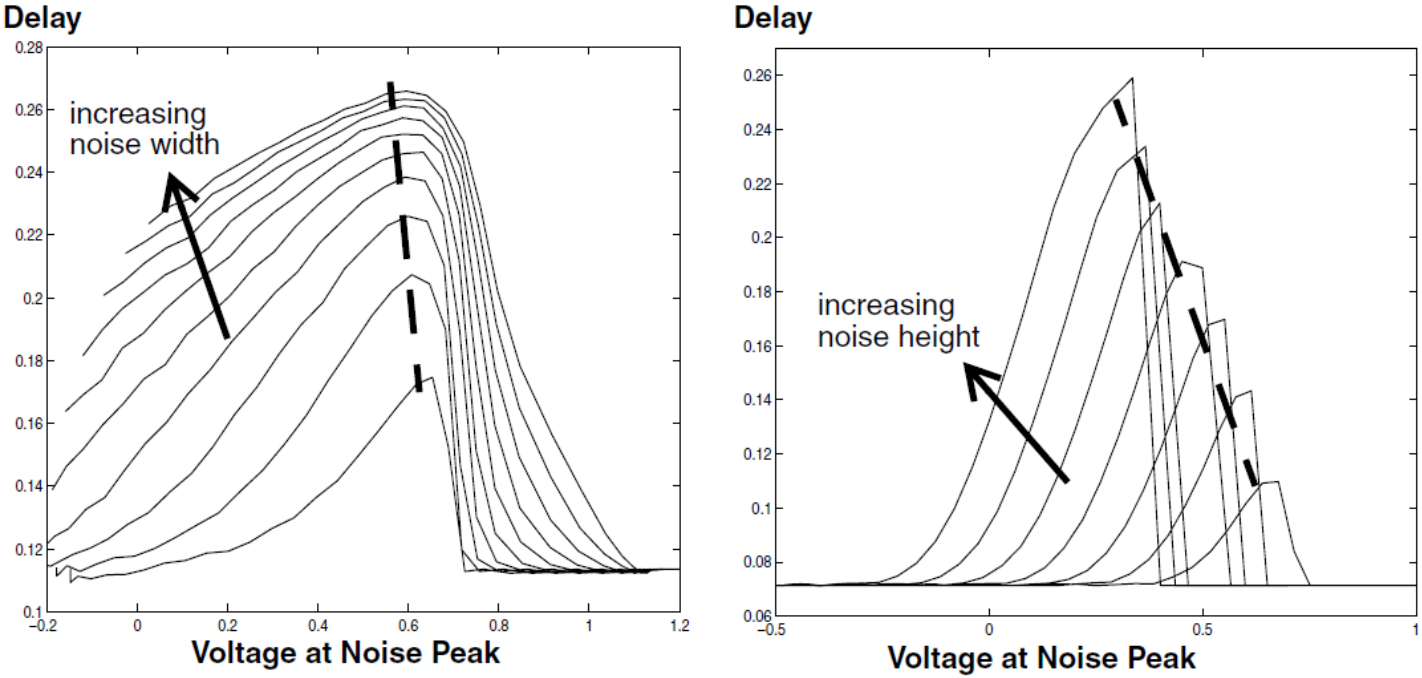




Source: Sirichotiyakul, Blaauw et. al, DAC 2001

- Worst case alignment
  - not sensitive to load, delay less sensitive as load increases
  - sensitive to victim slope, can be linearly interpolated between max and min slopes

# Delay Sensitivity to Alignment

- Worst case alignment not a linear function of bump width (duration) or height (peak voltage)
- Sirichotiyakul et al. pre-characterized these delays
- PrimeTime suggests using Composite Current Source (CCS) model to best estimate this effect



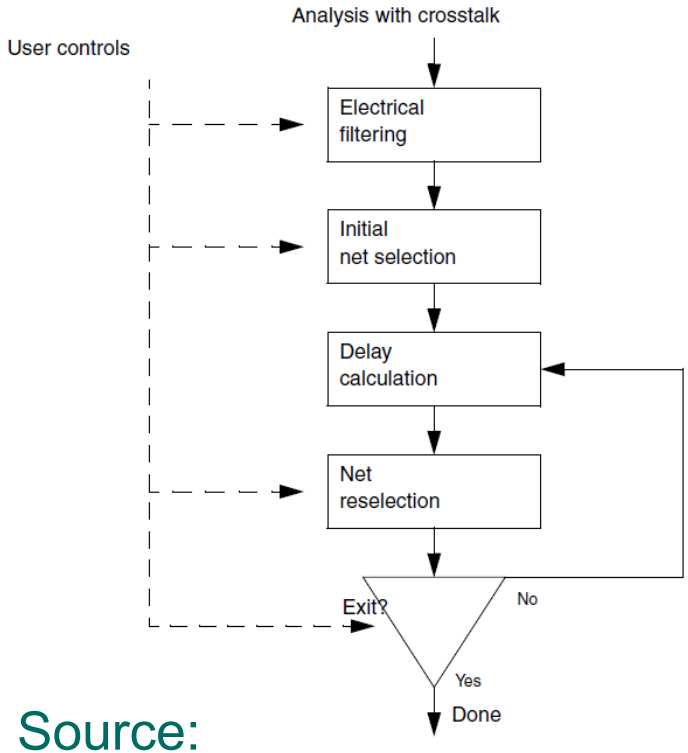

Source: Sirichotiyakul, Blaauw et. al, DAC 2001

# Note on Driver Models

- NLDM – Non-Linear Delay Model
  - Older models used by a large number of tools
- "Current Source" Models
  - Introduces circa 2005 to allow "SPICE-like accuracy" for newer technologies
  - CCS – Composite Current Source – Supported by Synopsys Tools
  - ECSM – Effective Current Source Model – Supported by Cadence Tools

# Iterative Delay Calculation

- Cycle in calculations:
  - Noise Peak Voltages depend heavily on window calculations
  - Delay Noise depends heavily on noise peak voltages
  - Window calculations depend heavily on delay noise
- To resolve this cycle, a signal integrity analyzer must iteratively solve these values until they converge
  - Both PrimeTime and Encounter do this




Source: Synopsys PrimeTime SI User Guide

# How to Fix SI Violations?

# Today's Lecture

- Basic Crosstalk Analysis
- Functional Noise Analysis
- Delay Noise Analysis
- Analyzing Signal Integrity in PrimeTime
- Reading SPEF Files

# Enabling SI Analysis

- Modified/New commands and arguments beyond the original ***run_pt.tcl*** script shown below

```
set si_enable_analysis TRUE
read_verilog "${RTL_DIR}${modname}_${type}.v"
current_design $modname
. . .
read_parasitics -keep_capacitive_coupling -format spef
"${RTL_DIR}${modname}_${type}.spef"
report_timing -input_pins -transition_time \
      -crosstalk_delta -delay_type min_max -path_type \
      full_clock_expanded > timing_ptsi_${corner}_${type}.rpt
```

# PrimeTime Min Path Analysis

```
Path Type: min

  Point                              DTrans   Trans    Delta     Incr       Path
  ------------------------------------------------------------------------------
  clock HCLK (rise edge)                                        0.0000     0.0000
  clock source latency                                          0.0000     0.0000
  HCLK (in)                                   0.0000            0.0000 &   0.0000 r
  HCLK__L1_I0/A (INV_X32)            0.0000   0.0013   0.0000   0.0011 &   0.0011 r
  HCLK__L1_I0/ZN (INV_X32)                    0.0122            0.0107 &   0.0117 f
  HCLK__L2_I1/A (INV_X32)            0.0000   0.0122   0.0000   0.0129 &   0.0246 f
  HCLK__L2_I1/ZN (INV_X32)                    0.0255            0.0247 &   0.0493 r
  u_logic_Gji2z4_reg/CK (DFFR_X2)
                                     0.0000   0.0255   0.0000   0.0073 &   0.0566 r
  u_logic_Gji2z4_reg/Q (DFFR_X2)              0.0070            0.0755 &   0.1321 r
  u_logic_U4450/B1 (OAI22_X1)        0.0000   0.0071   0.0000   0.0000 &   0.1321 r
  u_logic_U4450/ZN (OAI22_X1)                 0.0141            0.0248 &   0.1569 f
  u_logic_Gji2z4_reg/D (DFFR_X2)
                                     0.0000   0.0141  -0.0023  -0.0022 &   0.1547 f
  data arrival time                                                       0.1547
```

# PrimeTime Max Path Analysis

```
Path Type: max

  Point                              DTrans   Trans    Delta     Incr       Path
  ------------------------------------------------------------------------------
  clock HCLK (rise edge)                                        0.0000     0.0000
  clock source latency                                          0.0000     0.0000
  HCLK (in)                                    0.0000           0.0000 &   0.0000 r
  HCLK__L1_I0/A (INV_X32)            0.0000    0.0021   0.0000  0.0017 &   0.0017 r
  HCLK__L1_I0/ZN (INV_X32)                     0.0170           0.0126 &   0.0144 f
  HCLK__L2_I1/A (INV_X32)            0.0000    0.0273   0.0000  0.0185 &   0.0328 f
  HCLK__L2_I1/ZN (INV_X32)                     0.0309           0.0347 &   0.0675 r
  u_logic_Emi2z4_reg/CK (DFFR_X2)
                                     0.0000    0.0399   0.0000  0.0185 &   0.0860 r
  u_logic_Emi2z4_reg/Q (DFFR_X2)               0.1227           0.1985 &   0.2845 r
  FE_OFC95_u_logic_n17504/A (BUF_X32)
                                     0.0167    0.1396   0.0273  0.0317 &   0.3162 r
  FE_OFC95_u_logic_n17504/Z (BUF_X32)          0.0301           0.1254 &   0.4417 r
  u_logic_U1797/A1 (NAND2_X2)        0.0007    0.0310   0.0016  0.0068 &   0.4484 r
  u_logic_U1797/ZN (NAND2_X2)                  0.0780           0.0962 &   0.5446 f
```

# Notes

- Output shows that delay noise is not calculated on clock nets (haven't figured out why)
- Negative crosstalk delta for min-path analysis (expected)
- Positive crosstalk delta for max-path analysis (expected)

# Additional SI Commands

- **report_delay_calculation** gives a breakdown of aggressors

```
report_delay_calculation -crosstalk -from u_logic_U4450/ZN \
        -to u_logic_Gji2z4_reg/D
. . .
Victim is rising:
     Victim             Coupling    Driver          Clocks
      Net                 Cap       Lib Cell
  --------------      ---------   ------------    ---------------
  u_logic_U227_Z_0    0.000310    OAI22_X1         HCLK

   Aggressor        Coupling    Driver       Clocks      Attributes  Switching Bump
      Net             Cap       Lib Cell                             (ratio of VDD)
  --------------  ---------   -----------  ---------   ----------  --------------
  HCLK__L2_N1     0.000176    INV_X32       HCLK          A          0.057070
  u_logic_n14518  0.000134    OAI221_X1     HCLK          A          0.042436
```

# Additional SI Commands

- **report_noise** performs functional noise analysis

```
report_noise > noise_ptsi_${corner}_${type}.rpt
```

```
slack type: area

 noise_region: above_low
 pin name (net name)        width    height     slack
 ---------------------------------------------------
 FE_OFC295_u_logic_n17660/A (u_logic_n17660)
                            0.7794    0.4195
 noise_region: below_high
 pin name (net name)        width    height     slack
 ---------------------------------------------------
 FE_OFC295_u_logic_n17660/A (u_logic_n17660)
                            1.2922    0.5626
```

# Additional SI Commands

- **report_noise_calculation** gives details on aggressors for bump calculation

```
report_noise_calculation -below -high -from u_logic_U2176/ZN -to
FE_OFC295_u_logic_n17660/A
. . .
                        Height    Width     Area    Attributes
-----------------------------------------------------------------
 Aggressors:
 HRDATA[2]               0.0407    1.2140    0.0247
 u_logic_n20602          0.0136    1.1630    0.0079    A
 u_logic_U689_DATA2_0    0.0810    1.4330    0.0580    A
 FE_OFN242_u_logic_n18366
                         0.0272    1.0749    0.0146    A
 HADDR[28]               0.0468    1.0344    0.0242    A
 u_logic_n17759          0.0154    1.3757    0.0106    A
 . . .
 Total:                  0.5626    1.2922    0.3635
```

# Today's Lecture

- Basic Crosstalk Analysis
- Functional Noise Analysis
- Delay Noise Analysis
- Analyzing Signal Integrity in PrimeTime
- Reading SPEF Files

# SPEF

- SPEF = Standard Parasitic Exchange Format
- Syntax is defined in section 9.3 of the IEEE Standard 1481-1999
- Refer to that document (posted on resources page) for more info

# SPEF Name Map

- After the header, all port, net, and instance names are mapped to a shorter name
  - Why?
- Perhaps the most confusing aspect of SPEF

```
*NAME_MAP

*1 clock
*2 clock__L1_N0
*3 clock__L2_N0
*4 dec
*5 in[0]
```

# Port Definitions

- Each port definition contains the name, followed by direction (I, O, or B) and coordinates (*C) in microns

```
*PORTS

*1 I *C 0 48.93
*29 I *C 0 49.35
*28 I *C 0 47.53
*26 I *C 0 46.55
```

# Net definitions (1)

```
*D_NET *182 0.0002175

*CONN
*I *395:ZN O *C 45 50 *L 0 *D NAND4_X1
*I *392:A2 I *C 46 49 *L 0.000781 *D NOR4_X1

*CAP
1 *395:ZN 1.809e-05
2 *182:2 4.051e-05
3 *182:3 4.051e-05
4 *392:A2 1.809e-05
5 *182:2 *221:97 5.015e-05
6 *182:3 *221:101 5.015e-05

*RES
1 *182:3 *392:A2 6
2 *182:3 *182:2 3.5
3 *395:ZN *182:2 6
*END
```

- Each net is specified by *D_NET *ref total_cap*
- Capacitance units are defined in header (in this case, pF)
- Connections (*CONN)
  - *I = Internal
  - O/I = output/input
  - *C = coordinates
  - *L = Load cap
  - *D = Driving cell

# Net definitions (2)

```
*D_NET *182 0.0002175

*CONN
*I *395:ZN O *C 45 50 *L 0 *D NAND4_X1
*I *392:A2 I *C 46 49 *L 0.000781 *D NOR4_X1

*CAP
1 *395:ZN 1.809e-05
2 *182:2 4.051e-05
3 *182:3 4.051e-05
4 *392:A2 1.809e-05
5 *182:2 *221:97 5.015e-05
6 *182:3 *221:101 5.015e-05

*RES
1 *182:3 *392:A2 6
2 *182:3 *182:2 3.5
3 *395:ZN *182:2 6
*END
```

- Capacitors (*CAP)
  - id
  - node name (to which the other side of the cap is connected)
  - value (these should all sum to total_cap)
- Resistors (*RES)
  - id
  - node 1
  - node 2
  - value