Propagation Delay, Circuit Timing & Adder Design

ECE 152A – Summer 2009

Reading Assignment

- Brown and Vranesic
  - 2 Introduction to Logic Circuits
    - 2.9 Introduction to CAD Tools
      - 2.9.1 Design Entry
      - 2.9.2 Synthesis
      - 2.9.3 Functional Simulation
      - 2.9.4 Physical Design (2nd edition)
      - 2.9.5 Timing Simulation (2nd edition)
    - 2.9.4 Summary (1st edition)
Reading Assignment

- **Brown and Vranesic** (cont)
  - 3 Implementation Technology
    - 3.3.1 Speed of Logic Circuits
  - 3.5 Standard Chips
    - 3.5.1 7400-Series Standard Chips
  - 3.8 Practical Aspects
    - 3.8.3 Voltage Levels in Logic Gates
    - 3.8.4 Noise Margin
    - 3.8.5 Dynamic Operation of Logic Gates
    - 3.8.6 Power Dissipation in Logic Gates

---

Reading Assignment

- **Brown and Vranesic** (cont)
  - 5 Number Representation and Arithmetic Circuits
    - 5.1 Positional Number Representation
      - 5.1.1 Unsigned Numbers
      - 5.1.2 Conversion Between Decimal and Binary Systems
      - 5.1.3 Octal and Hexadecimal Representations
    - 5.2 Addition of Unsigned Numbers
      - 5.2.1 Decomposed Full-Adder
      - 5.2.2 Ripple-Carry Adder
      - 5.2.3 Design Example
Reading Assignment

- **Roth**
  - 1 Introduction Number Systems and Conversion
    - 1.2 Number Systems and Conversion
    - 1.3 Binary Arithmetic
  - 8 Combinational Circuit Design and Simulation Using Gates
    - 8.3 Gate Delays and Timing Diagrams

Properties of Digital Integrated Circuits

- **The Ideal Digital Circuit**

![Diagram of an ideal digital circuit with a power supply and input/output signals.](image)
Digital IC Definitions

- Amplitude and Voltage Transfer Characteristics

Digital IC Definitions

- Noise Margins
  - Sources of noise
  - Definition of noise margins
Propagation Delay

- When gate inputs change, outputs don’t change instantaneously
  - This delay is known as “gate” or “propagation” delay

\[ \varepsilon_1 = t_{PHL} \]
\[ \varepsilon_2 = t_{PLH} \]

\[ \varepsilon_1 \] is the propagation delay from input going high to output going low (inverting logic)
- \( t_{PHL} \)

\[ \varepsilon_2 \] is the propagation delay from input going low to output going high (inverting logic)
- \( t_{PLH} \)

- Terminology \( (t_{PHL} \text{ and } t_{PLH}) \) always refers to the transition on the output (whether circuit is inverting or not)
Propagation Delay

- **Multiple Gate Delays**
  - Example assumes that $t_{PLH}$ and $t_{PHL}$ equal 20 ns for both AND and NOR gate
    - Not always the case for different transitions or different gate types

```
Example assumes that tPLH and tPHL equal 20 ns for both AND and NOR gate
Not always the case for different transitions or different gate types
```

- **Maximum propagation delay** is the longest delay between an input changing value and the output changing value
- The path that causes this delay is called the **critical path**
  - The critical path imposes a limit on the maximum speed of the circuit
    - Max frequency = $f (\text{clk to } q + \text{critical path} + \text{setup time})$
    - … much more on this later
Propagación del Retraso

- Para un ejemplo de circuito, la ruta crítica es de cualquier cambio en la entrada A resultando en un cambio en G_2.
  - Circuito es Invertido (de A a G_2)
    - Con B = 1 y C = 0, A↑ causa G_2↓ (t_{PHL} = 20 ns) y A↓ causa G_2↑ (t_{PLH} = 20 ns)
  - Retraso de propagación máximo
    - 20 ns + 20 ns = 40 ns
      - Idéntico para cualquier A↑ o A↓
      - No siempre el caso

Propagación del Retraso

- Definiciones de transiciones y tiempos de retraso para (invertiendo) circuitos digitales
The CMOS Inverter

- Alternate symbol and more details
  - Current flows only when output switching
  - Power is frequency dependent

The CMOS Inverter

- Output switching requires charging (or discharging) parasitic and gate capacitance through a resistor(s)
  - Transistor “on resistance”
  - Wire capacitance and resistance
  - Gate capacitance
The CMOS Inverter

- SPICE Simulation of CMOS inverter pair
  - First inverter driven by ideal source
  - Full (distributed) and lumped RC loads

Transistor-Transistor Logic (TTL)

- Bipolar Junction Transistor (BJT) based technology and logic family
- Both input and output stages implemented with transistors (hence, TTL)
  - Earlier logic families used resistors (RTL) or diodes (DTL) in the input stage
- TTL first commercialized in mid 1960’s
  - Driven by many issues, not the least of which was the need for an on-board computer for the Lunar Excursion Module (LEM) in NASA’s Apollo program
Transistor-Transistor Logic (TTL)

- First “complete” family of digital integrated circuits
  - Small and medium scale integration (SSI and MSI)
    - SSI < 10 gates per device
    - MSI > 10 and < 100 gates per device
    - LSI and VLSI followed
- Commercial and military temperature ranges
  - 74XX – Commercial temperature range
    - 0 – 70°C
  - 54XX – Military temperature range
    - -55 – 125°C

“Significant” evolution of Texas Instruments’ TTL technology
- Standard TTL (1965) 54/74XX
- Schottky-Clamped TTL (1970) 54/74SXX
- Low Power, Schottky-Clamped TTL (1975) 54/74LSXX
- Advanced, Low Power,
  - Schottky-Clamped TTL (1980) 54/74ALSXX
- TTL compatible CMOS (1985) 54/74ACTXX

Compatible TTL families from other vendors
- Fairchild, Intel, Motorola, National and others
Transistor-Transistor Logic (TTL)

- Standard TTL, 2-input NAND Gate

TTL Electrical Characteristics

- Standard TTL (54/74)

<table>
<thead>
<tr>
<th>TABLE 7.3</th>
<th>Standard transistor-transistor logic (54/74 TTL): typical electrical characteristics at $T_A = 25^\circ C$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_{OH}/V_{OL}$</td>
<td>3.5 V/0.2 V</td>
</tr>
<tr>
<td>$V_{IH}/V_{IL}$</td>
<td>1.5 V/0.5 V</td>
</tr>
<tr>
<td>$N_M H/N_M L$</td>
<td>2.0 V/0.3 V</td>
</tr>
<tr>
<td>Logic swing</td>
<td>3.3 V</td>
</tr>
</tbody>
</table>
TTL Electrical Characteristics

- Comparison of Standard TTL (74), Schottky Clamped TTL (74S) and Low Power Schottky TTL (74LS)

### Table 7.4
Transistor-transistor logic: performance characteristics at $T_A = 25^\circ C$

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Series 74</th>
<th>Series 74S</th>
<th>Series 74LS</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_{DD(max)}$</td>
<td>2.4 V</td>
<td>2.7 V</td>
<td>2.7 V</td>
</tr>
<tr>
<td>$I_{H(max)}$</td>
<td>2.8 mA</td>
<td>2.0 mA</td>
<td>2.0 mA</td>
</tr>
<tr>
<td>$V_{IL(max)}$</td>
<td>-0.4 mA</td>
<td>-1.8 mA</td>
<td>-0.4 mA</td>
</tr>
<tr>
<td>$I_{H(min)}$</td>
<td>10 μA</td>
<td>20 μA</td>
<td>10 μA</td>
</tr>
<tr>
<td>$I_{L(min)}$</td>
<td>3 nA</td>
<td>3 nA</td>
<td>3 nA</td>
</tr>
<tr>
<td>Typical propagation delay time</td>
<td>10 μs</td>
<td>10 μs</td>
<td>10 μs</td>
</tr>
<tr>
<td>Typical power dissipation per gate</td>
<td>20 μW</td>
<td>20 μW</td>
<td>20 μW</td>
</tr>
</tbody>
</table>

### Table 7.2
Bipolar and CMOS logic performance characteristics ($T_A = 25^\circ C$)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>74LS</th>
<th>74HC</th>
<th>74HCT</th>
<th>74AC</th>
<th>74ACT</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_{DD(max)}$</td>
<td>2.7V</td>
<td>4.4V</td>
<td>4.4V</td>
<td>4.4V</td>
<td>4.4V</td>
</tr>
<tr>
<td>$I_{H(min)}$</td>
<td>2.0mA</td>
<td>3.1mA</td>
<td>2.0mA</td>
<td>3.1mA</td>
<td>2.0mA</td>
</tr>
<tr>
<td>$V_{IL(max)}$</td>
<td>0.4V</td>
<td>0.4V</td>
<td>0.4V</td>
<td>0.4V</td>
<td>0.4V</td>
</tr>
<tr>
<td>$I_{H(min)}$</td>
<td>20 μA</td>
<td>20 μA</td>
<td>20 μA</td>
<td>20 μA</td>
<td>20 μA</td>
</tr>
<tr>
<td>Typical propagation delay time</td>
<td>10 μs</td>
<td>10 μs</td>
<td>10 μs</td>
<td>10 μs</td>
<td>10 μs</td>
</tr>
<tr>
<td>Typical power dissipation per gate</td>
<td>2 mW</td>
<td>2.5 mW</td>
<td>2.5 mW</td>
<td>2.5 mW</td>
<td>2.5 mW</td>
</tr>
</tbody>
</table>

For Series 74LS: $V_{CC} = 5$ V, $C_L = 15$ pF
For CMOS Series: $V_{CC} = 4.5$ V, $C_L = 50$ pF

TTL vs. CMOS

- Comparison of Electrical Characteristics
Binary Numbers

- Unsigned and Signed Integers
  - Unsigned integers represent all positive values in the range 0 to $2^n - 1$
  - Signed integers in several flavors
    - Sign magnitude
    - One’s complement
    - Two’s complement
- We will be concerned with unsigned binary integers for this discussion of adders

Conversion Between Binary and Decimal

- Binary to Decimal
  
  \[1011.11_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} = 8 + 0 + 2 + 1.5 + 0.25 = 10.75_{10}\]

- Decimal to Binary

  Convert 55.12 to binary:
  
  \[
  \begin{array}{c|c}
  \hline
  n & \text{bits} \\
  \hline
  6 & 1 \\
  5 & 0 \\
  4 & 1 \\
  3 & 0 \\
  2 & 1 \\
  1 & 0 \\
  0 & 0 \\
  \hline
  \end{array}
  \]

  Convert 625.12 to binary:
  
  \[
  \begin{array}{c|c}
  \hline
  \text{bits} & \text{decimal} \\
  \hline
  4 & 625 \\
  3 & 0 \\
  2 & 0 \\
  1 & 0 \\
  0 & 0 \\
  \hline
  \end{array}
  \]
Octal and Hexadecimal Representation

Octal (2^3)

↕

Binary

↕

Hexadecimal (2^4)

Addition of Unsigned Numbers

- Half Adder
  - 2 input bits
    - x
    - y
  - 2 output bits
    - s (sum)
    - c (carry)
TTL Implementation

- SN7400: Quad, 2-input, positive NAND gates with totem pole outputs
  - SN indicates Texas Instruments
  - Pin assignments (top view) for dual-in-line package (DIP)

Schematic with SN7400's
- 2 IC's, 1 spare NAND gate
TTL Implementation

- SN7400
  - Switching characteristics (propagation delays)
    - $t_{PLH} \text{ (max)} = 22 \text{ ns}$
    - $t_{PHL} \text{ (max)} = 15 \text{ ns}$


<table>
<thead>
<tr>
<th>Type</th>
<th>Test Conditions</th>
<th>$t_{PLH} \text{ (max)}$</th>
<th>$t_{PHL} \text{ (max)}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>'00' '10</td>
<td></td>
<td>11</td>
<td>7</td>
</tr>
<tr>
<td>'04' '20</td>
<td>$C_L = 15 \text{ pF}$, $R_L = 400 \Omega$</td>
<td>12</td>
<td>8</td>
</tr>
<tr>
<td>'30'</td>
<td></td>
<td>13</td>
<td>9</td>
</tr>
</tbody>
</table>

Worst case propagation delay
- Critical path is x (or y) to sum
- Three levels of gate delay and three levels of inversion
  - Two possibilities
    - $t_{PLH} + t_{PHL} + t_{PLH}$
    - $t_{PHL} + t_{PHL} + t_{PHL}$
- Max delay is $t_{PLH} + t_{PHL} + t_{PHL}$
  - $22 \text{ ns} + 15 \text{ ns} + 22 \text{ ns} = 59 \text{ ns}$
  - Max frequency = $1 / (\text{clk to q} + 59 \text{ ns} + \text{setup time})$
Programmable Logic Devices

- A Programmable Logic Device (PLD) is a single, programmable device capable or replacing multiple, discrete TTL chips
  - PLD is comprised of “uncommited” gates and programmable switches to interconnect the gates
  - Simple PLD’s can realize 2 to 10 functions of 4 to 16 input variables
  - Complex PLD’s can implement circuits requiring 100’s of thousands of gates

Half Adder Implementation with a Programmable Logic Device (PLD)

- Schematic Capture (Design Entry)
  - Using “Primitive” library of logic elements
    - Specify logic function using generic logic gates rather than selecting physical devices (e.g., 7400 TTL)
    - CAD tool will determine actual implementation
PLD Implementation of Half Adder

- **Functional Simulation**
  - All propagation delays set to zero

<table>
<thead>
<tr>
<th>Name</th>
<th>0.0ns</th>
<th>40.0ns</th>
<th>60.0ns</th>
<th>80.0ns</th>
<th>100.0ns</th>
<th>120.0ns</th>
</tr>
</thead>
<tbody>
<tr>
<td>0+0=00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0+1=01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1+0=01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1+1=10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

PLD Implementation of Half Adder

- Map logical design onto a target architecture and physical device using CAD tool
  - *Logical function* is specified via the primitive library and implemented using *logical structures* incorporated into the target architecture
  - The physical device is a single chip hardware implementation of the design incorporating the structures of the target architecture
    - Altera MAX 7000 Complex Programmable Logic Device (CPLD) family for this example
PLD Implementation of Half Adder

- Timing Simulation
  - Must know specific device and package combination in PLD environment
    - Both contribute to performance
  - Simulation of physical implementation of design
    - Logical (gate) delays
    - Physical (interconnect) delays
    - I/O (package input/output) delays

Approximately 6ns delay from input to output
- $t_{PLH}$ and $t_{PHL}$

<table>
<thead>
<tr>
<th>Name</th>
<th>0</th>
<th>20.0ns</th>
<th>40.0ns</th>
<th>60.0ns</th>
<th>80.0ns</th>
<th>100.0ns</th>
<th>120.0ns</th>
</tr>
</thead>
<tbody>
<tr>
<td>y'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>y1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D(sum1th)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D(carry1th)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$t_{PLH}$</td>
<td>$\rightarrow$ 6ns</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$t_{PHL}$</td>
<td>$\rightarrow$ 6ns</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
I/O Delays

- Circuit to measure I/O delay
  - X1 to iodelay path through input receiver and output driver
    - Allows I/O delay to be separated from internal (core) delays

Timing Simulation
- Simulation indicates I/O delay dominates logic circuit delays for this (very small) design
VLSI Circuits

- **Intel 8080**

  - Address Bus Drivers
  - Ground Pad
  - Bidirectional Data Bus Driver/Receivers

---

VLSI Circuits

- **Intel Pentium**
**Full Adder**

- **Full Adder**
  - By adding a carry in input, multiple-bit numbers can be added by cascading full adder stages.
  - The sum and carry out become functions of three variables \( x, y \) and \( cin \).

---

**Full Adder**

- **Generic Circuit Implementation**
Full Adder Implementation

- Schematic Capture

As with the half adder, I/O delays dominate

\[ t_{PLH} = t_{PHL} \rightarrow 6\text{ns} \]
Ripple Carry Adder

- n-bit, Ripple Carry Adder
  - By cascading full adders, carry “ripples” from least significant bit toward most significant bit
  - Critical path becomes input to full adder 0 to output of full adder n

Two-Bit Ripple Carry Adder

- Schematic with I/O test circuit, half-adder, full adder and two-bit ripple carry adder
CPLD Implementation

- Timing Simulation

- Note propagation delay from y1 to carry2 is measured at 9.5 ns
  - Greater than simulated I/O delay of 6ns
  - Internal delays now visible (and measurable) at device pins
- Note also 3.5 ns “glitch” at 66ns
  - Resolution of simulation implied to be 3.5ns
Addendum:
Power Dissipation in CMOS Circuits

ECE 152A – Summer 2009

Power Dissipation in CMOS Circuits

- There are two components that establish the amount of power dissipation in a CMOS circuit
  - Static Power Dissipation
    - Constant current
  - Dynamic Power Dissipation
    - Currents attributed to switching
Power Dissipation in CMOS Circuits

■ Static dissipation
  □ Reverse bias leakage current
    ■ Parasitic diode between diffusion regions and substrate
  □ Subthreshold leakage current in static CMOS circuits
    ■ pMOS and/or nMOS devices not completely turned off
  □ Constant current in non static CMOS circuits
    ■ Psuedo-nMOS, I/O, Analog circuits, etc.

Power Dissipation in CMOS Circuits

■ Dynamic dissipation
  □ Switching transient current
    ■ Occurs on transition from 1 to 0 (or 0 to 1)
      □ Results in short current pulse from $V_{DD}$ to $V_{SS}$
      □ Referred to as “short-circuit dissipation”
    ■ Dependent on rise and fall times
      □ Slow rise and fall times increase short circuit current
    ■ Critical in I/O buffer design
      □ Dominant component of dynamic power with little or no capacitive loading
Dynamic dissipation (cont)

- Charging and discharging of load capacitances
  - As capacitive loading is increased, the charging and discharging currents begin to dominate the current drawn from the power supplies.

\[
P_D = \frac{1}{t_p} \int_0^{t_p} i_n(t) V_{nss} \, dt + \frac{1}{t_p} \int_{t_p/2}^{t_p} i_p(t) (V_{DD} - V_{nss}) \, dt,
\]

where

- \( i_n \) = n-device transient current
- \( i_p \) = p-device transient current.

Now, let's consider the power dissipation for a step input with a duration of \( \Delta t \). The total power dissipation can be expressed as

\[
P = \frac{C_L}{t_p} \int_0^{\Delta t} V_{nss} \, P_{on} \, dt + \frac{C_L}{t_p} \int_0^{\Delta t} (V_{DD} - V_{nss}) \, P_{off} \, dt
\]

where

- \( P_{on} = \frac{C_L}{t_p} \int_0^{\Delta t} V_{nss} \, dt \)
- \( P_{off} = \frac{C_L}{t_p} \int_0^{\Delta t} (V_{DD} - V_{nss}) \, dt \)

resulting in

\[
P_D = \frac{C_L}{t_p} V_{DD} \, t_P.
\]
Power Dissipation in CMOS Circuits

- Dynamic short-circuit vs. capacitive current

FIGURE 4.36 SPICE circuits and results showing dynamic short-circuit current and capacitive current for a CMOS circuit. The volt/ampere transients are shown. The current sources are used to measure currents.