EE 330
Lecture 44

Digital Circuits

- Ring Oscillators
- Sequential Logic
- Array Logic
- Memory Arrays

Final: Tuesday Dec 12 12:00 - 2:00
Dynamic Logic

Basic Dynamic Logic Gate

Any of the PDNs used in complex logic gates would work here!

- Have eliminate the PUN!
- Ideally will have a factor of 4 or more reduction in $C_{IN}$
- Ideally will have a factor of 4 or more reduction in dynamic power dissipation relative to that of equal rise/fall!
- Ideally will have a factor of 2 reduction in dynamic power dissipation relative to that of minimum size!
Dynamic Logic

Basic Dynamic Logic Gate

Advantages:

• Lower dynamic power dissipation (Ideally 4X)
• Improved speed (ideally 4X)

Limitations:

• Output only valid during evaluate state
• Need to route a clock
  (and this dissipates some power)
• Premature Discharge !
• More complicated
• Charge storage on internal nodes of PDN
• No Static hold if output H
Review from Last Time

Domino Logic

Diagram of Domino Logic with symbols and connections.
Dynamic Logic

- p-channel logic gate will pre-charge low
- Phasing of PUN and PDN networks is reversed
- Some performance loss with p-channel logic devices
- Direct coupling between alternate type dynamic gates is possible without causing a premature discharge problem

Review from Last Time
Review from Last Time

Zipper Logic

Acceptable Implementation in Zipper
Digital Circuit Design

- Hierarchical Design
- Basic Logic Gates
- Properties of Logic Families
- Characterization of CMOS Inverter
- Static CMOS Logic Gates
  - Ratio Logic
- Propagation Delay
  - Simple analytical models
    - FI/OD
    - Logical Effort
  - Elmore Delay
- Sizing of Gates
- Propagation Delay with Multiple Levels of Logic
- Optimal driving of Large Capacitive Loads
- Power Dissipation in Logic Circuits
  - Other Logic Styles
    - Dynamic Logic
    - Array Logic
- Ring Oscillators
  - Sequential Logic Circuits
  - Memory Arrays
Ring Oscillators

• Odd number of stages will oscillate (even number will not oscillate)
• Waveform nearly a square wave if \( n \) (number of stages) is large
• Output will slightly imbalance ring and device sizes can be compensated if desired
• Usually use a prime number (e.g. 31)
• Number of stages usually less than 50 (follow by dividers)
• Frequency highly sensitive to process variations and temperature

\[ f_{\text{osc}} \approx \frac{1}{nt_{\text{PROP}}} \]

• \( n \) is the number of stages
• \( t_{\text{PROP}} \) is the propagation delay of a single stage (all assumed identical)
Digital Building Blocks

- Combinational Logic
- Sequential Logic
  - Shift Registers (stack)
- Array Logic
- Memory Arrays
Sequential Logic Circuits

- Flip Flops needed for sequential logic circuit
- Only one type of flip flop is required
- Invariably require clocked edge-triggered master-slave flop flops
- Flip flop circuits can be very simple
- Flip flops are part of Standard Cell Libraries
Sequential Logic Elements

**Latch**
- D is held on Q when $C_{LK}$ is low
- $\overline{Q}$ may also be present

**Pulsed Latch**
- D is held on Q when $C_{LK}$ is low
- $C_{LK}$ is a pulse
- $\overline{Q}$ may also be present
- Special case of Latch

**Flip Flop**
- May have one or two inputs
- Inputs transferred to output on $C_{LK}$
- Four basic types
  - J-K, S-R, D, T
- Any type can be obtained from other with simple combinational logic ckts
- Often edge triggered and Master-Slave
- Most widely used sequential logic element

All can have preset or clear input added
Latch Flip-Flop Terminology

- **Transparent:**
  - Input to Latch Appears on the Latch Output immediately while in transparent state

- **Opaque**
  - Input to the Latch does not appear at the output while in the opaque state

- **Edge Triggered**
  - Input to latch at a clock transition determines when the input is transferred to the latch output

- **Master-Slave**
  - Two-stage cascaded structure where output from one serves as input to the second stage
Latches

- Very simple
- When $X_C$ is high, in transparent state (tracks input)
- Negative edge triggered

Limitations:
- $Q$ can get to only $V_{DD}-V_T$
- Leakage of charge will occur when $X_C$ is low
- Any loads placed on $Q$ will further degrade held signal
Latches

- Buffering eliminates loading
- Provides rail-rail output
- Provides signal inversion
- Second inverter will eliminate signal inversion
- Output transparent when $X_C$ is high

- $C$ is input capacitance to inverter
- Need not show load since buffered
- Charge will eventually leak off of $C$
Latches

- Becomes nonvolatile
- Provides complimentary output
- Output transparent when $X_C$ is high
- Contention on output when loading
  - (must size devices so latch will load)
• Becomes nonvolatile
• Provides complementary output
• Output transparent when $X_C$ is high
• Contention on output removed
• 6-transistor Cell
• Actually the basic memory element used in many SRAM arrays
Flip Flops

Four Basic Flip Flops

- **S-R Flip Flop**
  - Inputs: S, R
  - Outputs: Q, \( \overline{Q} \)
  - Clock: \( C_{LK} \)

- **J-K Flip Flop**
  - Inputs: J, K
  - Outputs: Q, \( \overline{Q} \)
  - Clock: \( C_{LK} \)

- **D Flip Flop**
  - Inputs: D
  - Outputs: Q, \( \overline{Q} \)
  - Clock: \( C_{LK} \)

- **T Flip Flop**
  - Inputs: D
  - Outputs: Q, \( \overline{Q} \)
  - Clock: \( C_{LK} \)
Flip Flops

Implementation of the S-R Flip Flops

- Many different flip flops exist
- Extremely high number of flip flops in a design warrants design of a good structure
- Not clocked or Master Slave
Flip Flops

Implementation of the S-R Flip Flops

S-R Flip Flop

Clocked S-R flip flop

20 transistor implementation
Output transparent when clock is H
Flip Flops

Implementation of the S-R Flip Flops

Clocked S-R flip flop

Clocking Edge-Triggered S-R Master Slave Flip Flop
Flip Flops

Implementation of the S-R Flip Flops

Clock edge-triggered S-R Master Slave Flip Flop

Master-Slave Edge-triggered Clocked S-R Flip Flop

Large Device Count : 40 transistors
Flip Flops

D Flip Flop

5 transistors

With static hold
6 transistors

Output transparent when in clock is high
Flip Flops

Master-Slave Edge-triggered D Flip Flop

Timing Diagram

- 12 transistors (but will work with 10)
- Many other simple D Flip-flops exist as well
Shift Registers

- Basic 1-bit dynamic shift register
- Data is stored on parasitic capacitor $C_p$
- $C_p$ is generally input capacitance to inverter and omitted from diagram
Shift Registers

Dynamic Shift Register

- 6 transistor cell
- Must be clocked to retain data

Timing Diagram:
Shift Registers

Dynamic Shift Register
6 transistor cell

Dynamic Shift Register with Static Hold
Shift Registers

Dynamic Shift Register

Simple redrawing
Shift Registers

Dynamic Shift Register

- FIFO Operation
- Layout so stages can be simply abutted
Shift Registers

Dynamic Shift Register

Bi-directional Dynamic Shift Register

If $T_L$ and $T_R$ replaced with $H \cdot T_L$ and $H \cdot T_L$, have static hold operation
Shift Registers

Dynamic Shift Register

\[
\begin{align*}
&d_1 \\
&D \\
&Q_L \\
&X_L
\end{align*}
\]

\[
\begin{align*}
&d_2 \\
&SR \\
&TL \\
&SL \\
&TR
\end{align*}
\]

\[
\begin{align*}
&d_3 \\
&SR \\
&TL \\
&SL \\
&TR
\end{align*}
\]

\[
\begin{align*}
&d_n \\
&SR \\
&TL \\
&SL \\
&TR
\end{align*}
\]

n-bit Parallel-Load, Parallel-Read Bidirectional Dynamic Shift Register

- Useful for Parallel to Serial and Serial to Parallel Conversion
- Can be put in static hold state if \( T_L \) and \( T_R \) replaced with \( H \oplus TL \) and \( H \oplus TL \)
Digital Building Blocks

• Combinational Logic
• Sequential Logic
  – Shift Registers (stack)
• Array Logic
• Memory Arrays
Array Logic

- Array logic is often used for sections of logic that may change later in the design or that will be changed for different variants of a product.
- FPGA are a special case of array logic.
- Can personalize array logic with only one layer of metal:
  - Very quick turn-around and low incremental costs (as few as one additional mask).
Array Logic

Will consider only two types
  – Gate Array
  – Sea of Gates

Variants of the following approach are possible depending upon process but this will convey the basic concepts
Array Logic

Gate Array

- Can add M1 (blue), M2 (purple), contact (M1 to Poly), via (M1 to M2)
- Upper and lower metal shown actually lie above poly and are already present
- Assume upper M1 is $V_{DD}$ and lower M1 is $V_{SS}$
- Array can be very large
- Routing channels between segments of array
Array Logic

Sea of Gates

- Can add M1 (blue), M2 (purple), contact (M1 to Poly), via (M1 to M2)
- Upper and lower metal shown actually lie above poly and are already present
- Assume upper M1 is $V_{DD}$ and lower M1 is $V_{SS}$
- Array can be very large
- Routing channels between segments of array
Array Logic

Gate Array

Example:

- Via (M1 to M2)
- Contact (M1 to diff, Poly)
Array Logic

Gate Array

$V_{DD}$

A

B

$V_{SS}$

F

G

Example:

A

B

F

G

Via (M1 to M2)

Contact (M1 to diff,Poly)
Array Logic

Gate Array

V_{DD}

A

B

V_{SS}

F

Example:

\[ \text{Via (M1 to M2)} \]

\[ \text{Contact (M1 to diff,Poly)} \]
Array Logic

Sea of Gates

$V_{DD}$

$V_{SS}$

Example:

- Via (M1 to M2)
- Contact (M1 to diff, Poly)
Array Logic

Sea of Gates

$V_{DD}$

$V_{SS}$

Diffusion Partition

Example:

- Via (M1 to M2)
- Contact (M1 to diff, Poly)
Array Logic

Sea of Gates

\[ V_{DD} \]

\[ V_{SS} \]

Example:

- Via (M1 to M2)
- Contact (M1 to diff, Poly)
Digital Building Blocks

• Combinational Logic
• Sequential Logic
  – Shift Registers (stack)
• Array Logic
  Memory Arrays
Row Decoder Architectures

Row decoder is Pseudo n-MOS NOR Gate

Typically n/2 inputs where n is the address length

\[ R_k = \overline{A_1} \cdot \overline{A_2} \cdot \overline{A_3} \]

\[ R_k = (A_1 + A_2 + A_3) \]
Row Decoder Architectures

Transistor sites typically reserved in the layout for efficient, compact layout.
Mem Cells

Static RAM (SRAM)

- Uses PTL and cross-coupled inverters
- Sizing of “switches” must be strong enough to write to cell
- No static power dissipation in this PTL implementation
Mem Cells

- Site reserved for possible transistor
- Actually programmed with contact to gate and diffusion
- Can personalize with one or two masks
- Single transistor per bit
- Uses only one column line
Mem Cells

EPROM or EEPROM

n-channel MOSFET

- Very thin floating gate
- Charge tunnels onto gate during programming to change $V_T$ a lot
- Conceptual diagram only
- Somewhat specialized processing for reliable floating gate devices
Mem Cells

- Floating Gate Transistor
- Programmed by Changing the Threshold Voltage
- Nonvolatile Memory
- Can be electrically programmed with EEPROM
- Limited number of read/write cycles (but enough for most applications)
- Uses only one column line
Mem Cells

- Charge stored in small parasitic capacitor
- Very small cells
- Volatile and dynamic
- Special processes to make $C_P$ large in very small area
- $C_P$ is actually a part of the transistor
- Somewhat tedious architecture (details not shown) needed to sense very small charge
End of Lecture 44