Sub System Design And Layout

TOPIC COVERED

- **SUB SYSTEM DESIGN & LAYOUT:** Structured design of combinational circuits – Dynamic CMOS &
- clocking – Tally circuits – (NAND-NAND, NOR-NOR and AOI logic) – EXOR structure – Multiplexer
- structures – Barrel shifter

Introduction

- Large systems are composed of sub-systems, known as Leaf-Cells
- The most basic leaf cell is the common logic gate (inverter, nand, ..etc)
- Structured Design
  - High regularity
  - Leaf cells replicated many times and interconnected to form the system
- Logical and systematic approach to VLSI design is essential

Good Design Methodology

- Define Requirements
- Partition overall architecture into appropriate sub-systems
- Consider communication paths in order to develop sensible interrelationships between subsystems
- Draw a floor-plan of how the system is to map onto silicon and iterate above as appropriate
- Aim for regular structures so that design is largely a matter of replication
- Lay-out each cell (stick diagram)
- Carry out design rule checks
- Simulate performance of each cell / subsystem

Computer Aided Design

- Early CAD systems used a graphical editor to design the layout of the chip directly - this is now impractical for anything above small scale integration
- Designing a chip requires a variety of CAD tools, both the analyze the design and synthesize parts of the design
- CAD tools are used at many stages in the design and so must be able to communicate with each other.
- Computer aided design approaches make use of cell libraries consisting of tested and debugged transistor circuits
- Analysis and design verification tools are required to achieve correct designs before chips are manufactured

Dealing with Complexity

- Divide and conquer - limit the number of components you deal with at any one time
- Group several components into larger components
  - transistors form gates
  - gates form functional units
  - functional units form processing elements
  - etc

Levels of Design

- To manage the complexity of VLSI design, models are used to abstract away all details not required to understand a design
- levels of design (or levels of abstraction)

Major Levels of Design

- Specification
  - Description of requirements
- Systems Level
  - placing and interconnecting major functional units
- Function Level
  - specification and design of major functional units
- Logic/Circuit Level
  - Gate level design, gate interconnection design
- Layout Level
  - what will actually be patterned onto the chip, how the chip will be processed
- Physics Level
  - the physics of gate and switch operation

Silicon Compilation

- Chip design systems have some similarity to language compilation systems
  - they take in a high level description of a system
  - they output the device layout (like machine code)
  - they reuse tried and tested components (like libraries)
- Scaling
  - As fabrication techniques improve, transistors get smaller and smaller
  - The libraries can be made scale - invarient so that re-engineering for smaller feature size involves only recompilation

Why Integration?

- Lower parasitics = higher speed
- Lower power consumption
- Physically smaller
- Higher reliability (due to reduced interconnections)
- Repeatability - Whole systems on single chip
- Cost
  - Integration reduces cost for large volumes
  - Relatively less manual assembly
  - Lower cost per unit

Top - Down vs Bottom - Up Design

- Top down design adds functional detail - creates lower levels of abstraction from upper levels
- Bottom up design creates abstractions from low level behaviour
- Good design effort needs both top down and bottom up

Design Validation

- Must check at every step that errors have not been introduced
  - the longer the error remains the more expensive it becomes to remove it

Manufacturing Test

- Not the same as design validation - just because design is right doesn’t mean that every chip coming off the line will be right
- Must quickly check whether manufacturing defects destroy functions of chip
- Must also speed grade

Chip Architecture

- After high level design is complete, it is necessary to decide on how design is to be implemented in silicon
- The implementation plan is known as the floor plan
- First step in laying out a floor plan is the routing of supply and clock rails
- In doing this sufficient space must be left between power rails to allow for data-buses and combinational logic cells
- Decide on relative positions of major functional blocks
- Use routing algorithm (software)
- Routing algorithm will minimise total routing area
- Critical nets (groups of wires) will be routed first to guarantee shorter, straighter paths
- Must consider clock skews and data path delays between clocked elements
- Power and ground is required for all parts of the circuit
- Usually routed in metal due to high currents
- Rarely should they be allowed to cross over
- To avoid metal migration (electromigration) metal lines should be wide enough to handle large capacitative loads

Electromigration

- The exchange of momentum between electrons and metal lattice atoms can cause physical voids or cracks at grain boundaries
- These defects grow under stress and eventually cause an open circuit

Subsystem Design 2

EE213 VLSI Design

This section contains some notes on logic implementation and more complex gates etc. Full details are in Pucknell and Eshraghian pages 146 to 159.

Logic Implementation

- When designing digital circuits with MOS technology, there are 2 basic approaches in building logic circuits
- Switch Logic
- Gate / Restoring Logic

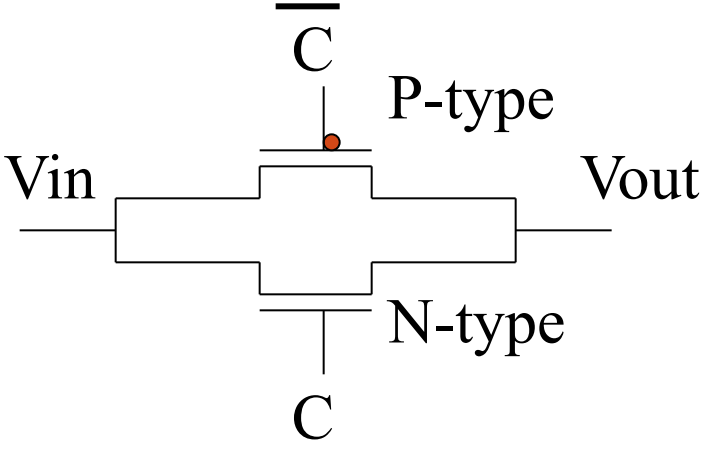
Switch Logic

- Based on pass transistors or transmission gates
- Fast for small arrays
- No static current from supply lines
- Logic based on relay logic
- Easy implement basic AND / OR connections
- Pass transistor logic levels get degraded by Vt effects
- Transmission gates do not suffer from Vt effects but
  - more complex
  - more area
  - potential fabrication problems
  - requires true and complement of gate signal
- No pass transistor gate may be driven through one or more pass transistors

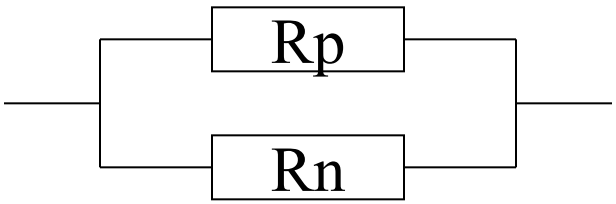
Switch Logic Levels

- See Kamran and Eshraghian pages 148 - 150 for discussion on logic levels
  - Logic 1 degraded for n-type pass
  - Logic 0 degraded for p-type pass
  - Transmission gates - good logic levels
  - Loss of logic level 1 if the gate of a pass transistor is driven from another pass transistor

Transmission Gate

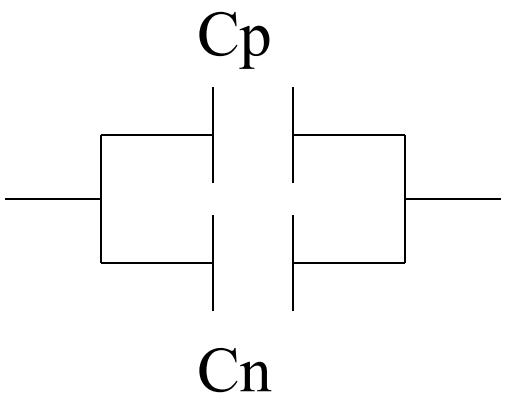


Resistive Model



R = Rp // Rn

Capacitative Model



C = Cp // Cn

Gate (Restoring) Logic

- Based on general arrangement of inverter
- NAND gates can be constructed with both CMOS and NMOS technology
- For NMOS, L:W ratio must be considered to achieve desired Zpu / Zpd ratio
- What is the required L:W ratio for NMOS NAND gate with N inputs?
  - Zpu / (n* Zpd) = 4 / 1
  - See derivation in Kamran and Eshraghian pages 150 - 156

NAND Gates

- NMOS NAND gate requirements larger than those for inverter. Need additional transistors and corresponding adjustment in length of pull-up transistor
- NMOS NAND gate delays are increased in direct proportion to number of inputs added, Tnand = N*Tinv
- For CMOS NAND gate natural asymmetry is improved
- Must allow for extended fall time on capacitative loads due to additional n-transistors in series

NOR Gates

- NMOS NOR Gates
  - can be easily extended to large number of inputs
  - each pd has same ratio as for an inverter
  - area reasonable since pu not affected by number of inputs
  - speed as fast as an inverter -> preferred nmos gate
- CMOS NOR Gates
  - pull-up resistance effect aggravated by number of transistors connected in series (p-type) -> rise and fall time asymmetries increased
  - shift in transfer function -> noise immunity decreased
  - will need L:W ratio adjustments