united nations educational, scientific and cultural organization

international atomic energy agency

the
abdus salam
international centre for theoretical physics

The United Nations University

SMR.1073-5

ICTP-UNU-Microprocessor Laboratory
Fifth Course on Basic VLSI Design Techniques

9 November - 4 December 1998

IC DESIGN STYLES

Jorgen CHRISTIANSEN
CERN
EP Division
CH-1211 Geneva 23
SWITZERLAND

IC design styles

J. Christiansen,
CERN - EP/MIC
Jorgen.Christiansen@cern.ch

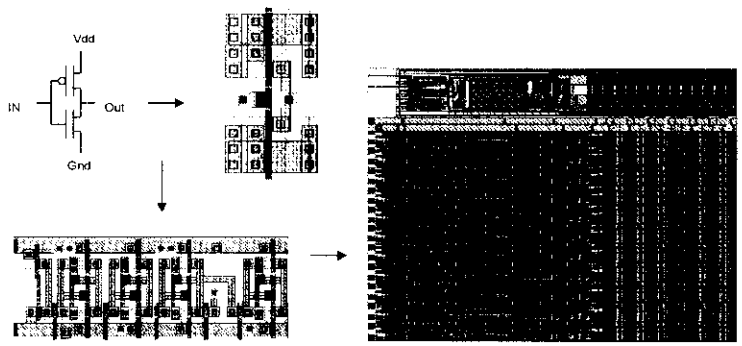
Design styles

- Full custom
- Standard cell
- Gate-array
- Macro-cell
- "FPGA"
- Combinations

Full custom

- Hand drawn geometry
- All layers customized
- Digital and analog
- Simulation at transistor level (analog)
- High density
- High performance
- Long design time

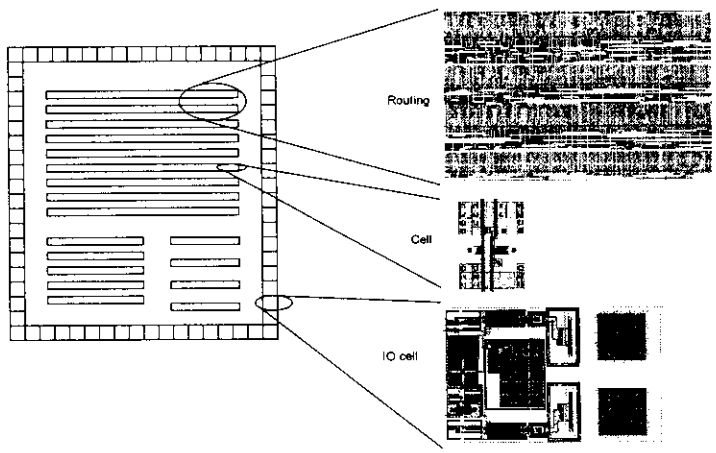
Full custom



Standard cells

- Standard cells organized in rows (and, or, flip-flops,etc.)
- Cells made as full custom by vendor (not user).
- All layers customized
- Digital with possibility of special analog cells.
- Simulation at gate level (digital)
- Medium density
- Medium-high performance
- Reasonable design time

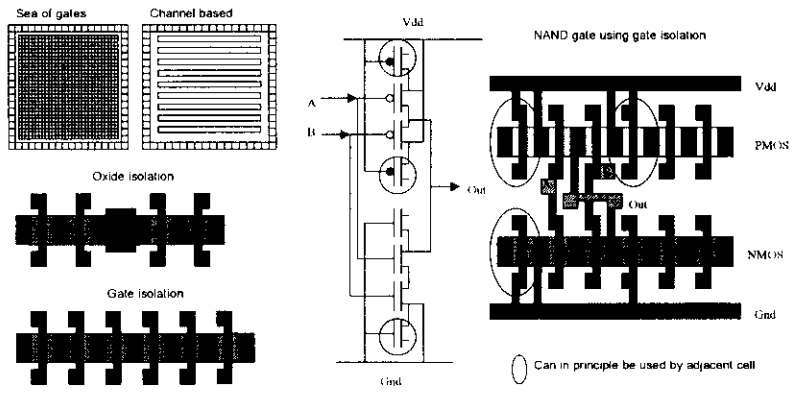
Standard cells



Gate-array

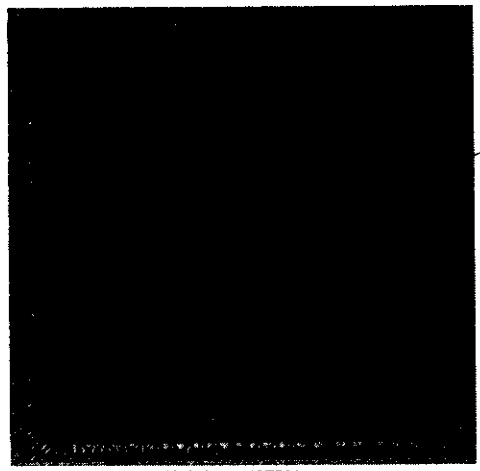
- Predefined transistors connected via metal
- Two types: Channel based
  Channel less (sea of gates)
- Only metallization layers customized
- Fixed array sizes (normally 5-10 different)
- Digital cells in library (and, or, flip-flops,etc.)
- Simulation at gate level (digital)
- Medium density
- Medium performance
- Reasonable design time

Gate-array



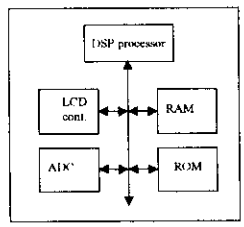
Gate-array

Sea of gates



Macro cell

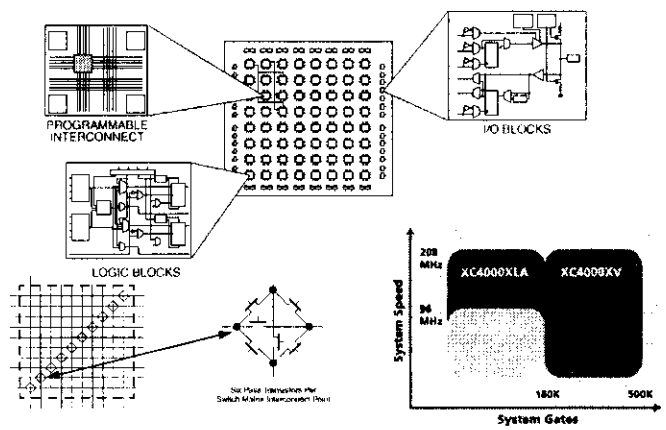
- Predefined macro blocks (Processors, RAM,etc)
- Macro blocks made as full custom by vendor
- All layers customized
- Digital and some analog (ADC)
- Simulation at behavioral or gate level (digital)
- High density
- High performance
- Short design time
- Use standard on-chip busses
- "System on a chip"



FPGA = Field Programmable Gate Array

- Programmable logic blocks
- Programmable connections between logic blocks
- No layers customized (standard devices)
- Digital only
- Low - medium performance (<50 - 100MHz)
- Low - medium density (up to ~100k gates)
- Programmable by: SRAM, EEROM, Anti_fuse, etc
- Cheap design tools on PC's
- Low development cost
- High device cost

FPGA



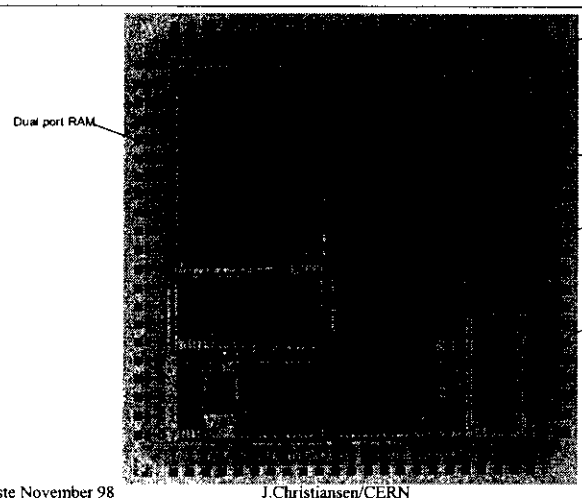
Comparison

	FPGA	Gate array	Standard cell	Full custom	Macro cell
Density	Low	Medium	Medium	High	High
Flexibility	Low (high)	Low	Medium	High	Medium
Analog	No	No	No	Yes	Yes
Performance	Low	Medium	High	Very high	Very high
Design time	Low	Medium	Medium	High	Medium
Design costs	Low	Medium	Medium	High	High
Tools	Simple	Complex	Complex	Very complex	Complex
Volume	Low	Medium	High	High	High

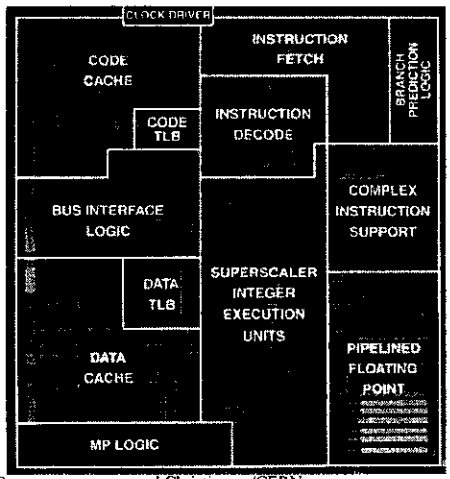
High performance devices

- Mixture of full custom, standard cells and macro's
- Full custom for special blocks: Adder (data path), etc.
- Macro's for standard blocks: RAM, ROM, etc.
- Standard cells for non critical digital blocks

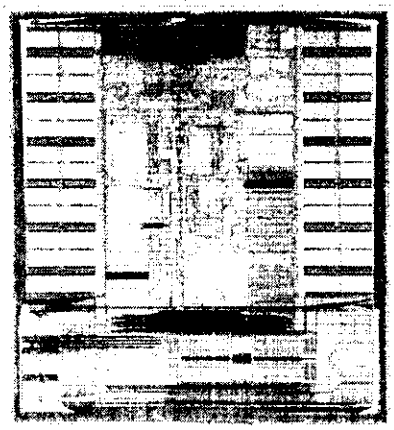
ASIC with mixture of full custom,RAM and standard cells



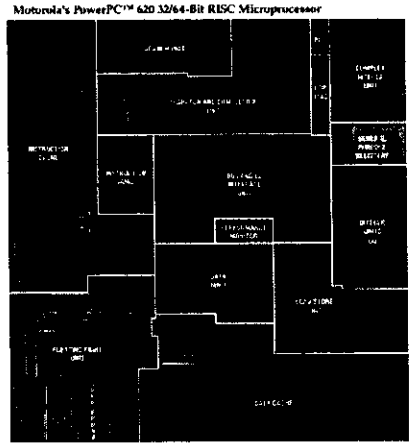
Pentium



Alpha

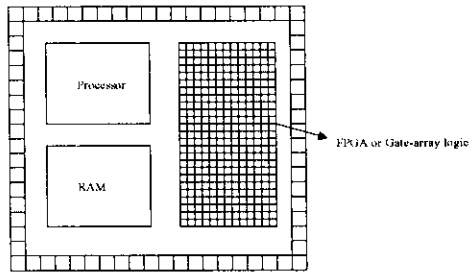


Motorola's PowerPC™ 620 32/64-Bit RISC Microprocessor



New combinations

- FPGA's with RAM, PCI interface, Processor, ADC, etc.
- Gate arrays with RAM, Processor, ADC, etc



Design methodology

J. Christiansen,
CERN - EP/MIC
Jorgen.Christiansen@cern.ch

Design Methodology

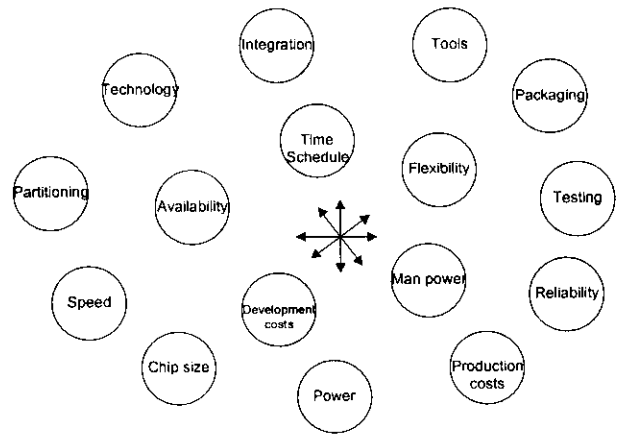
- Specification
- Trade-off's
- Design domains - abstraction level
- Top-down - Bottom up
- Schematic based
- Synthesis based
- Getting it right - Simulation
- Lower power

Specification

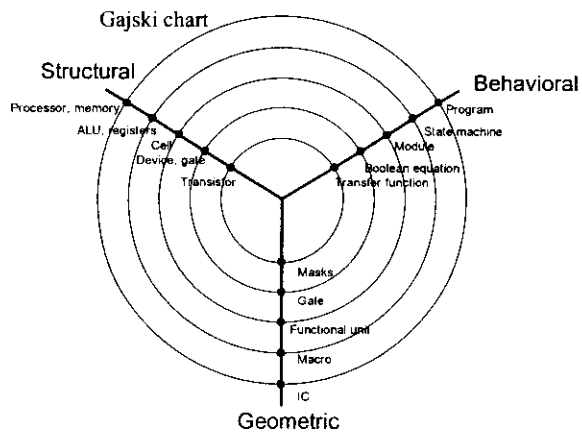
- A specification of what to construct is the first major step.
- A detailed specification must be agreed upon with the system people. Major changes during design will result in significant delays.
- Requirements must be considered at many levels

  System
  Board
  Hybrid
  IC
- Specifications can be verified by system simulations.
- Specification is 1/4 - 1/3 of total IC project !.

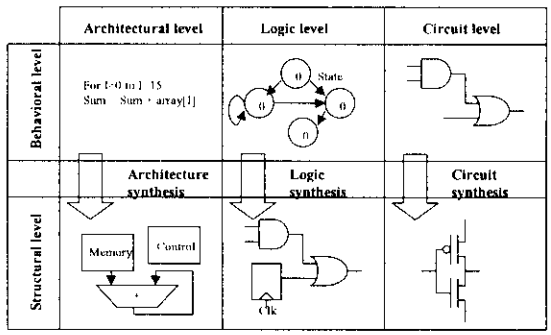
Trade offs



Design domains

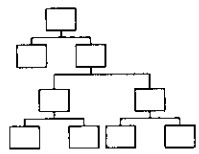


Design domains and synthesis



Top - down design

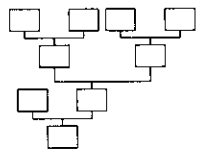
- Choice of algorithm (optimization)
- Choice of architecture (optimization)
- Definition of functional modules
- Definition of design hierarchy



- Split up in small boxes - split up in small boxes - split up in small boxes
- Define required units (adders, state machine, etc.)
- Floor-planning
- Map into chosen technology (synthesis, schematic, layout)
  (change algorithms or architecture if speed or chip size problems)
- Behavioral simulation tools

Bottom - up

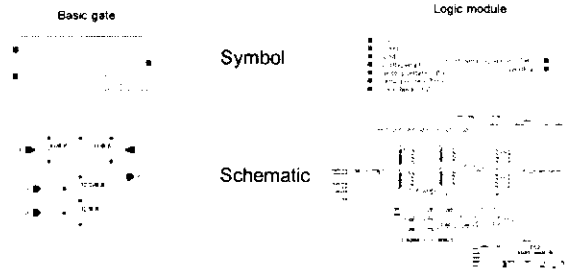
- Build gates in given technology
- Build basic units using gates
- Build generic modules of use
- Put modules together
- Hope that you arrived at some reasonable architecture
- Gate level simulation tools



Comment by one of the main designers of the Pentium processor
The design was made in a typical top - down , bottom - up , inside - out design methodology

Schematic based

- Symbol of module defines interface
- Schematic of module defines function
- Top - down: Make first symbol and then schematic
- Bottom - up: Make first Schematic and then symbol



Synthesis based

- Define modules and their behavior in a proper language
  (also used for simulation)
- Use synthesis tools to generate schematics and symbols (netlists)

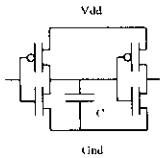
```
always @(posedge clk)
        begin
        if (set) coarse <   #(testff_delay) offset;
        else if (coarse     count_roll_over)
                            coarse <   #(testff_delay) 0;
        else coarse <= #(testff_delay) coarse + 1;
        end
```

Getting it right - Simulation

- Simulate the design at all levels (transistor, gate, system)
- Analog simulator (SPICE) for full custom design
- Digital gate level simulator for gate based design
- Mixed mode simulation of mixed analog-digital design
- Behavioral simulation at module level (Verilog, VHDL)
- All functions must be simulated and verified.
- Worst case data must be used to verify timing
- Worst - Typical - Best case conditions must be verified
- Use programming approach to verify large set of functions (not looking at waveform displays)

Low power design

- Low power design gets increasingly important:
  Gate count increasing > increasing power
  Clock frequency increasing > increasing power
  Packaging problems for high power devices
  Portable equipment working on battery
- Where does power go:
  1: Charging and dis-charging of capacitance: Switching nodes
  2: Short circuit current: Both N and P MOS conducting during transition
  3: Leakage currents: MOS transistors (switch) does not turn completely off

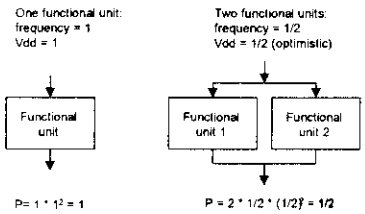


$$P = N_{switch} * f * C * V_{dd}^2 + N_{switch} * f * E_{short} + N * I_{leak} * V_{dd}$$

$K * Vdd^2$

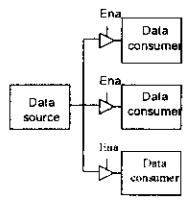
Decrease power

- Lower Vdd:
  5v > 2.5v gives a factor 4 !
  New technologies use lower Vdd because of risk of gate-oxide break-down and hot electron effect.
- Lower Vdd and duplicate hardware

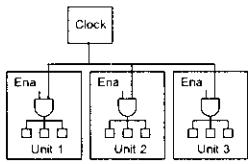


- Lower number of switching nodes:
  The clock signal often consumes 50% of total power:
  Gate clocks for modules not working
  Not use clocks
  Lower signal activity

Lower signal activity

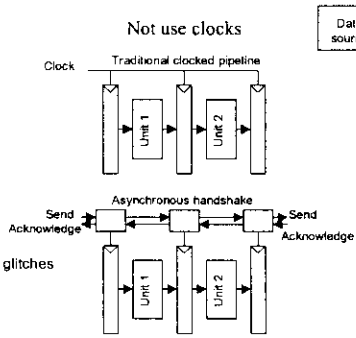


Clock gating



Warning: Clock gating may introduce glitches

Not use clocks



IC design Tools

J. Christiansen,
CERN - EP/MIC
Jorgen.Christiansen@cern.ch

Cell development

- **Schematic entry** (transistor symbols)
- **Analog simulation** (SPICE models)
- **Layout** (layer definitions)
- **Design Rule Checking**, DRC (design rules)
- **Extraction** (extraction rules and parameters)
- **Electrical Rule Checking**, ERC (ERC rules)
- **Layout Versus Schematic**, LVS (LVS rules)
- Analog simulation.
- Characterization: delay, setup, hold, loading sensitivity,etc.
- Generation of **digital simulation** model with back annotation.
- Generation of synthesis model
- Generation of symbol and black-box for place & route

Digital design

- **Behavioral simulation**
- **Synthesis** (synthesis models)

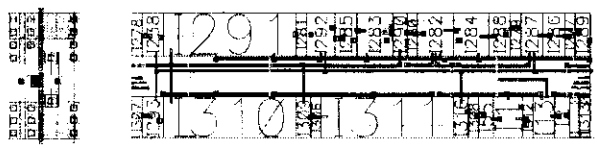
} Or direct **schematic entry**

- Gate level simulation (gate models)
- **Floor planning**
- **Loading estimation** (loading estimation model)
- Simulation with estimated back-annotation
- **Place and route** (place and route rules)
- **Design Rule Check**, DRC (DRC rules)
- Loading **extraction** (rules and parameters)
- Simulation with real back-annotation
- Design export
- Testing: Test generation, Fault simulation, Vector translation

Design entry

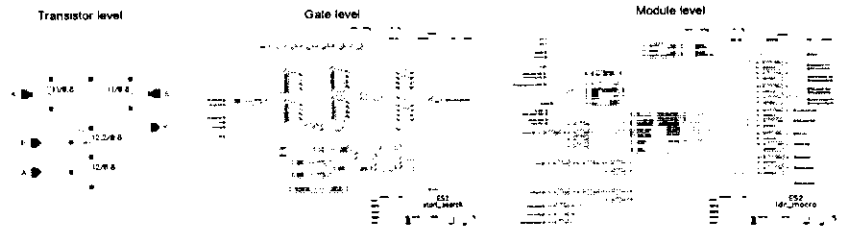
- Layout
  - Drawing geometrical shapes: Defines layout hierarchy
    Defines layer masks

Requires detailed knowledge about CMOS technology
Requires detailed knowledge about design rules
Requires detailed knowledge about circuit design
Slow and tedious
Optimum performance can be obtained
No yield guarantee from manufacturer when making full custom cells



- Schematic
  - Drawing electrical circuit: Defines electrical hierarchy
    Defines electrical connections
    Defines circuit: transistors, resistors,,,

Requires good circuit design knowledge for analog design
Requires good logic design knowledge for digital design (boolean logic, state machines)
Gives good overview of design hierarchy
Significant amount of time used for manual optimization



- Behavioral
  - Writing behavior (text): Defines behavioral hierarchy
    Defines algorithm
    Defines architecture
  - Synthesis tool required to map into gates
  - Often integrated with graphical block diagram tool.

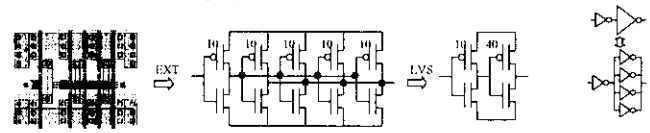
```
assign #(tcq-logic_delay)
bar_clk = (m_extest | m_sample | m_intest) | clk_dr;
bar_shift = (m_extest | m_sample | m_intest) & shift_dr;

                always @(posedge clk)
                        begin
                        if (set) counse <= #(tcq-ff_delay) offset;
                        else if (counse == count_roll_over)
                                counse <= #(tcq-ff_delay) 0;
                        else counse <= #(tcq-ff_delay) counse + 1;
                        end
                                                module add_and_mult( a,b,c, out)
                                                input[31:0]     a,b,c;
                                                output[31:0]    out;
                                                wire[31:0] internal_add;

                                                adder32         add1(a,b, internal_add);
                                                multiplier32    mult1( internal_add, c, out);
                                                endmodule
```

Verification

- **Design Rule Check:**
  Checks geometrical shapes: width, length, spacing, overlap, etc.
- **Electrical rule check:**
  Checks electrical circuit: unconnected inputs
  shorted outputs
  correct power and ground connection
- **Extraction:**
  Extracts electrical circuit: transistors, connections, capacitance, resistance
- **Layout versus schematic:**
  Compares electrical circuits: transistors: parallel or serial
  (schematic and extracted layout)



Simulation

- Simulates behavior of designed circuit
  - Input: Models (transistor, gates, macro)
    Textual netlist (schematic, extracted layout, behavioral)
    User defined stimulus
  - Output: Circuit response (waveforms, patterns)
    Warnings
- Transistor level simulation using analog simulator (SPICE)
  - Time domain
  - Frequency domain
  - Noise

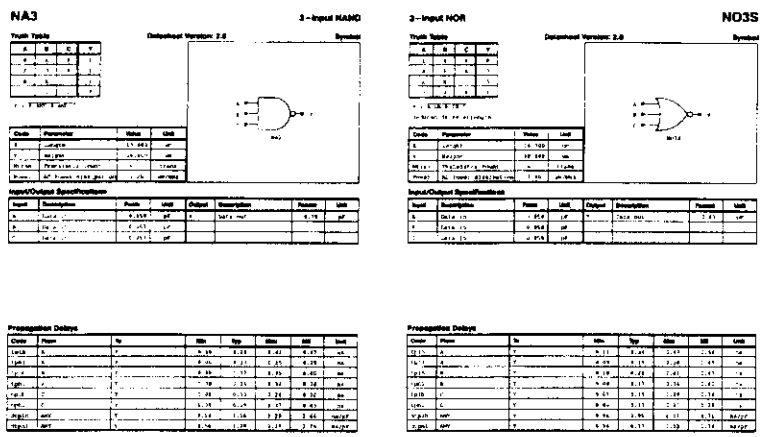
- Gate level simulation using digital simulator
  - Logic functionality
  - Timing: Operating frequency, delay, setup & hold violations
- Behavioral simulation
  - System and IC definition (algorithm, architecture)
  - Partitioning
  - Complexity estimation

Normally same simulator

Gate level models

- Border between transistor domain (analog) and digital domain
- Digital gate level models introduced to speed up simulation.
- Gate level model contains:
  - Logic behavior
  - Delays depending on: operating conditions, loading, signal slew rates
  - Setup and hold timing violation checks
- Gate level model parameters extracted from transistor level simulations and characterization of real gates.

Gate data sheet

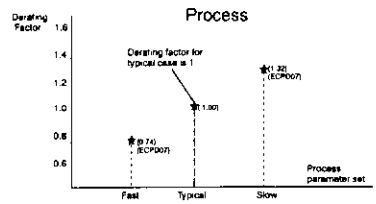


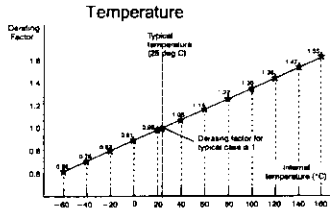
De-rating factors

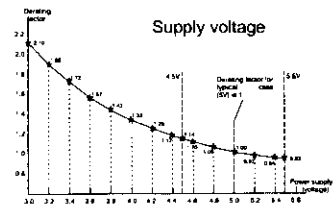
Worst case: 1.32 x 1.55 x 1.14 = 2.33 (2)

Typical case: 1 x 1 x 1 = 1

Best case 0.74 x 0.68 x 0.93 = 0.45 (0.5)

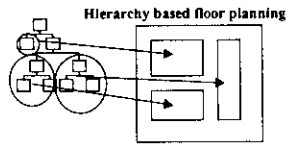


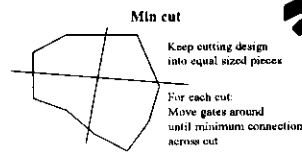


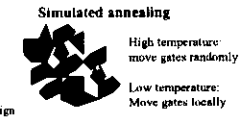


Place and Route

- Generates final chip from gate level netlist
  - Goals: Minimum chip size
    Maximum chip speed.
- Placement:
  - Placing all gates to minimize distance between connected gates
    - Floor planning tool using design hierarchy
    - Specialized algorithms (min cut, simulated annealing, etc.)
    - Timing driven
    - Manual intervention
  - Very compute intensive

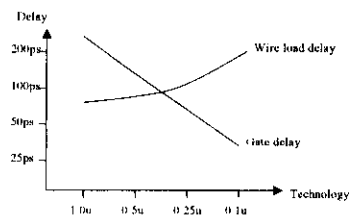


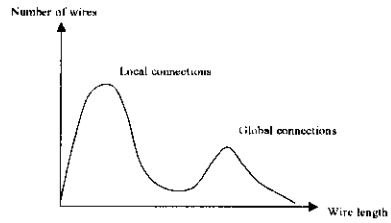




- Routing:
  - Channel based: Routing only in channels between gates
    (few metal layers: 2)
  - Channel less: Routing over gates
    (many metal layers: 3 - 5)
  - Often split in two steps:
    - Global route: Find a coarse route depending on local routing density
    - Detailed route: Generate routing layout

- Performance of sub-micron CMOS IC's are to a large extent determined by place & route.
  - Loading delays bigger than intrinsic gate delays
  - Wire R-C delays starts to become important in sub-micron
  - Clock distribution over complete chip gets critical at operating frequencies above 100Mhz.





Design tool framework

- Design tools from one vendor normally integrated into a framework which enables tools to exchange data.
  - Common data base
  - Automatic translation from one type to another
  - (Allows third part tools to be integrated into framework)
- Few standards to allow transport of designs between tools from different vendors.
  - VHDL and Verilog behavioral models and netlists
  - EDIF netlist, SPICE netlist for analog simulation
  - GDSII layout
  - Standard Delay Format (SDF) for gate delays.
  - Small vendors must be compatible with large vendors.

Transporting designs between tools from different vendors often cause problems

Required tools for different designs

- FPGA
  - A: PC based schematic entry with time estimator and simple Place & route
  - B: Behavioral modeling with synthesis, simulation and place & route.
- Gate array
  - A: Schematic entry and simulation
  - B: Behavioral modeling with synthesis and simulation.
    - Place and route performed by vendor
- Full custom
  - Layout, DRC, extraction and transistor level simulation
- Standard cell, macro and full custom
  - All tools described required

Source of CAE tools

- Cadence
  - Complete set of tools integrated into framework
- Mentor
  - Complete set of tools integrated into framework
- Synopsis
  - Power full synthesis tools
  - VHDL simulator
- Avant
  - Power full place and route tools
  - Hspice simulator with automatic characterization tools
- Div commercial:
  - View-logic, Summit, Tanner, etc.

- Free shareware:
  - Spice, Magic, Berkley IC design tools, Aliance
  - Diverse from the web.
- Complete set of commercial high performance CAE tools cost ~1 M$ per seat ! (official list price).
- University programs: Complete set of tools ~10K$
  - Europe: Eurochip
  - US: Mosis
  - Japan: ?

Hardware describing languages and Synthesis

J. Christiansen,
CERN - EP/MIC
Jorgen.Christiansen@cern.ch

Hardware describing languages (HDL)

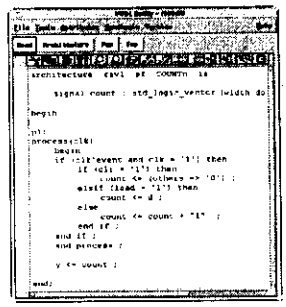
- Describe behavior not implementation
- Make model independent of technology
- Model complete systems
- Specification of sub-module functions
- Speed up simulation of large systems
- Standardized text format
- CAE tool independent

- VHDL
  - Very High speed integrated circuit Description Language
  - Initiated by American department of defense as a specification language.
  - Standardized by IEEE
- Verilog
  - First real commercial HDL language from gateway automation (now Cadence)
  - Default standard among chip designers for many years
  - Until a few years ago, proprietary language of Cadence.
  - Now also a IEEE standard because of severe competition from VHDL. Result: multiple vendors

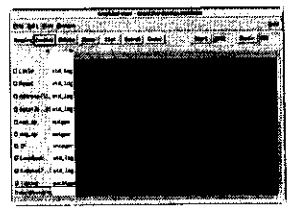
- Compiled/Interpreted
  - Compiled:
    - Description compiled into C and then into binary or directly into binary
    - Fast execution
    - Slow compilation
  - Interpreted:
    - Description interpreted at run time
    - Slow execution
    - Fast "compilation"
    - Many interactive features
  - VHDL normally compiled
  - Verilog exists in both interpreted and compiled versions

Design entry

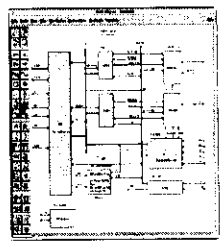
- Text:
  - Tool independent
  - Good for describing algorithms
  - Bad for getting an overview of a large design

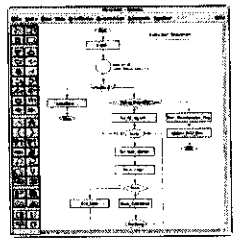


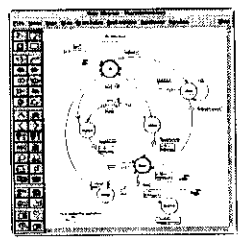
- Add-on tools
  - Block diagrams to get overview of hierarchy
  - Graphical description of final state machines (FSM)
    - Generates synthesizable HDL code
  - Flowcharts
  - Language sensitive editors
  - Waveform display tools



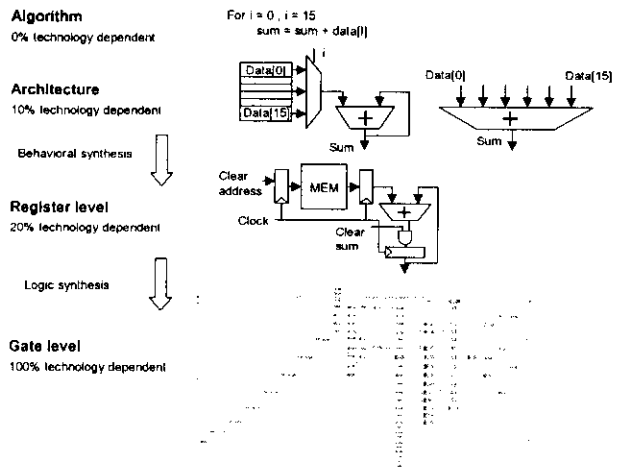
From Visual HDL, Summit design





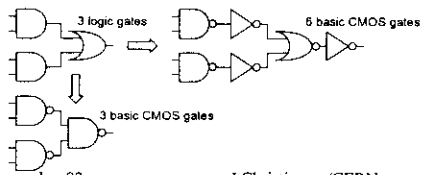


Synthesis

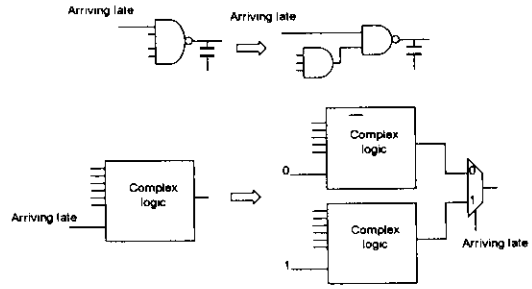


Logic synthesis

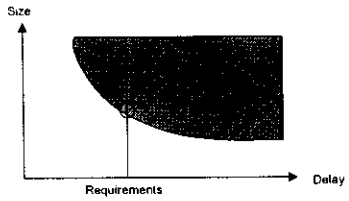
- HDL compilation (from VHDL or Verilog)
  - Registers: Where storage is required
  - Logic: Boolean equations, if-then-else, case, etc.
- Logic optimization
  - Logic minimization (similar to Karnaugh maps)
  - Finds logic sharing between equations
  - Maps into gates available in given technology
  - Uses local optimization rules



- Timing optimization
  - Estimate loading of wires
  - Defined timing constraints (clock frequency, delay, etc.)
  - Perform transformations until all constraints fulfilled



- Combined timing - size optimization
  - Smallest circuit complying to all timing constraints



  - Best solution found as a combination of special optimization algorithms and evaluation of many alternative solutions
    (Similar to simulated annealing)

- Problems in synthesis
  - Dealing with "single late signal"
  - Mapping into complex library elements
  - Regular data path structures:
    - Adders: ripple carry, carry look ahead, carry select,etc.
    - Multipliers, etc.

    Use special guidance to select special adders, multipliers, etc..

  Performance of sub-micron technologies are dominated by wiring delays (wire capacitance)
- Synthesis in many cases does a better job than a manually optimized logic design.

  (in much shorter time)

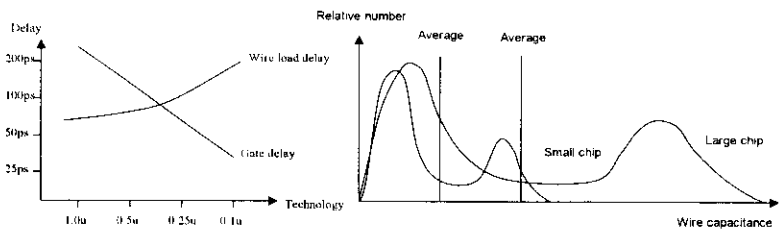
- Wire loading

  Timing optimization is based on a wire loading model.

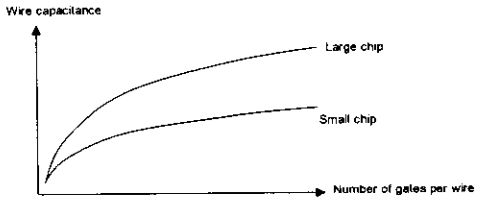
  Loading of gate = input capacitance of following gates + wire capacitance

  Gate loading known by synthesizer

  Wire loading must be estimated



- Estimate wire capacitance from number of gates connected to wire.

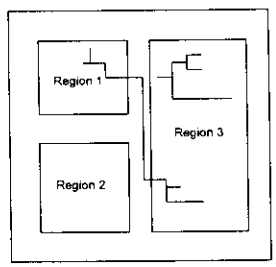


Advantage: Simple model
Disadvantage: Bad estimate of long wires
(which limits circuit performance)

- Estimate using floor plan

Inside local region:
Estimate as function of number of gates and size of region

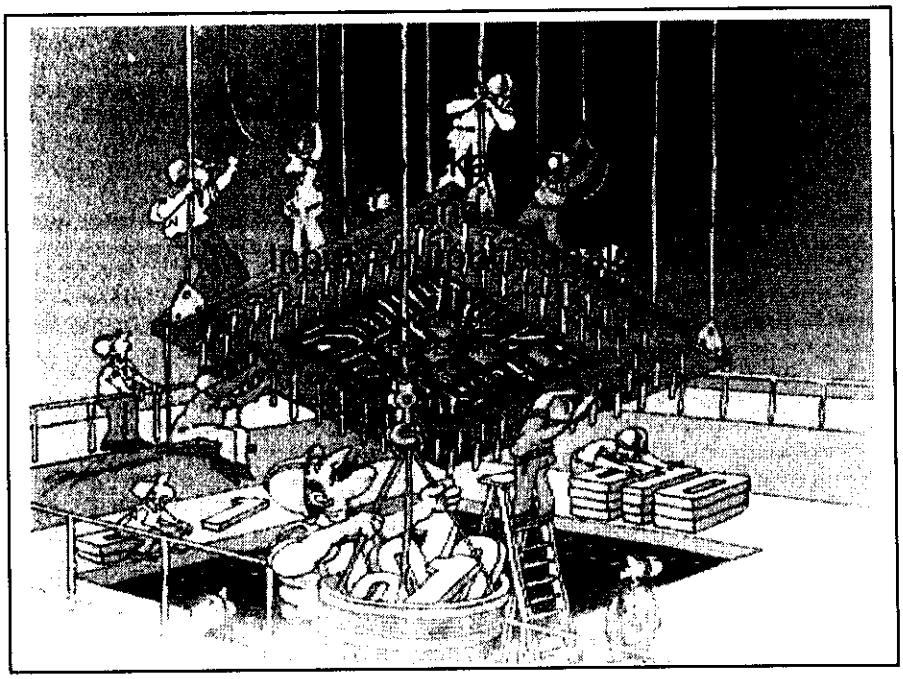
Between regions:
Use estimate of physical distance between routing regions.



Advantage: Realistic estimate
Disadvantage: Synthesizer most work with complete design

- Iteration
  - Synthesis with crude estimation
  - P&R with extraction of real loading
  - Re-synthesize starting from real loads
  - Repeat X times
- Timing driven P&R
  - Synthesize with crude estimation
  - Use timing calculations from synthesis to control P&R
- Integration of synthesis and P&R
  - Floor planning - timing driven - iteration

- Synthesis in the future
  - Integration of synthesis and P&R
  - Synthesizable standard modules (processor, PCI interface, Digital filters, etc.)
  - Automatic insertion of scan path for production testing.
  - Synthesis for low power
  - Synthesis of self-timed circuits (asynchronous)
  - Behavioral synthesis
  - Formal verification



Requirements to package

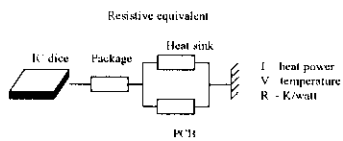
- Protect circuit from external environment
- Protect circuit during production of PCB
- Mechanical interface to PCB
- Interface for production testing
- Good signal transfer between chip and PCB
- Good power supply to IC
- Cooling
- Small
- Cheap

Materials

- Ceramic
  - Good heat conductivity
  - Hermetic
  - Expensive (often more expensive than chip itself !)
- Metal (has been used internally in IBM)
  - Good heat conductivity
  - Hermetic
  - Electrical conductive (must be mixed with other material)
- Plastic
  - Cheap
  - Poor heat conductivity
    Can be improved by incorporating metallic heat plate.

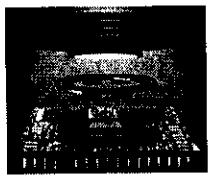
Cooling

- Package must transport heat from IC to environment
- Heat removed from package by:
  - Air: Natural air flow, Forced air flow improved by mounting heat sink
  - PCB: Transported to PCB by package pins
  - Liquid: Used in large mainframe computers

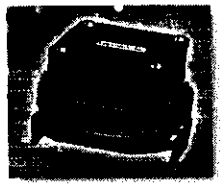


- Package types:
  - Below 1 watt: Plastic
  - Below 5 watt: Standard ceramic
  - Up to 30 watt: Special

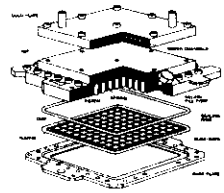
Passive heat sink



Active heat sink



Water cooled mainframe computer



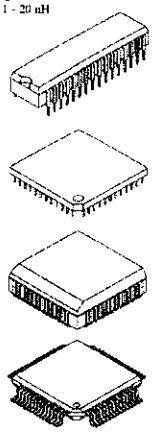
Chip mounting

- Pin through hole
  - Pins traversing PCB
  - Easy manual mounting
  - Problem passing signals between pins on PCB (All layers)
  - Limited density
- Surface Mount Devices (SMD)
  - Small footprint on surface of PCB
  - Special machines required for mounting
  - No blocking of wires on lower PCB layers
  - High density

Traditional packages

- DIL (Dual In Line)
  - Low pin count
  - Large
- PGA (Pin Grid Array)
  - High pin count (up to 400)
  - Previously used for most CPU's
- PLCC (Plastic leaded chip carrier
  - Limited pin count (max 84)
  - Large
  - Cheap
  - SMD
- QFP (Quarter Flat pack)
  - High pin count (up to 300)
  - small
  - Cheap
  - SMD

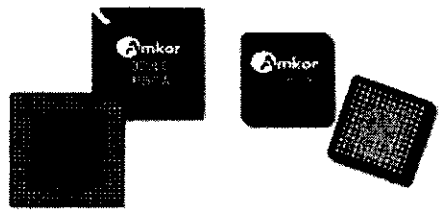
Package inductance:
1 - 20 nH



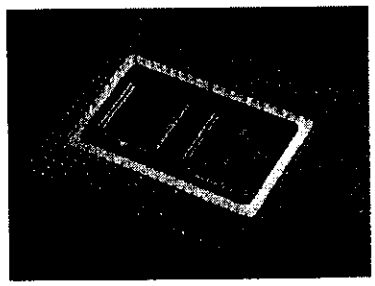
New package types

- BGA (Ball Grid Array)
  - Small solder balls to connect to board
  - small
  - High pin count
  - Cheap
  - Low inductance
- CSP (Chip scale packaging)
  - Similar to BGA
  - Very small packages

Package inductance
1 - 5 nH



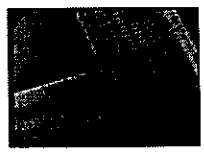
- MCP (Multi Chip Package)
  - Mixing of several technologies in same component
  - Yield improvement by making two chips instead of one



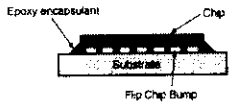
P6: processor + second level cache

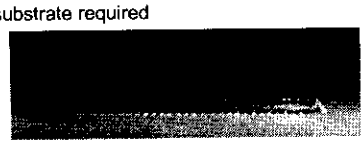
Chip to package connection

- Bonding
  - Only periphery of chip available for IO connections
  - Mechanical bonding of one pin at a time (sequential)
  - Cooling from back of chip
  - High inductance (~1nH)



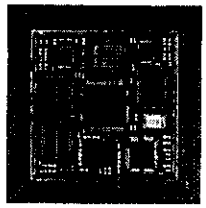
- Flip-chip
  - Whole chip area available for IO connections
  - Automatic alignment
  - One step process (parallel)
  - Cooling via balls (front) and back if required
  - Thermal matching between chip and substrate required
  - Low inductance (~0.1nH)

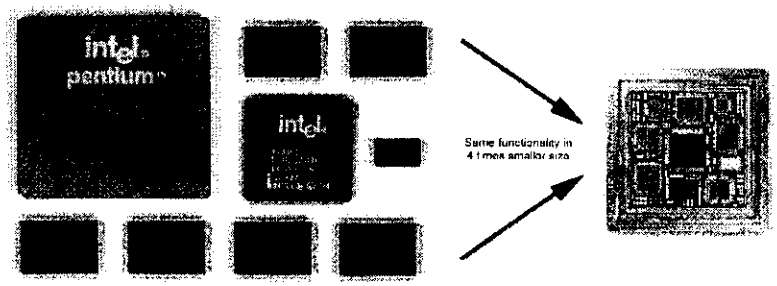




Multiple Chip Module (MCM)

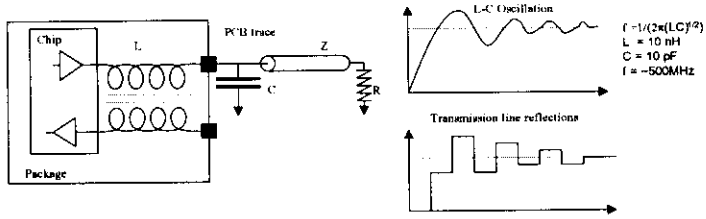
- Increase integration level of system (smaller size)
- Decrease loading of external signals > higher performance
- No packaging of individual chips
- Problems with known good die:
  - Single chip fault coverage: 95%
  - MCM yield with 10 chips: $(0.95)^{10} = 60\%$
- Problems with cooling
- Still expensive





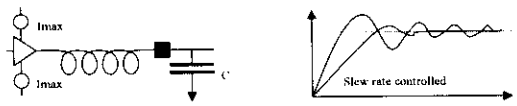
Signal Interface

- Transfer of IC signals to PCB
  - Package inductance.
  - PCB wire capacitance.
  - L - C resonator circuit generating oscillations.
  - Transmission line effects may generate reflections
  - Cross-talk via mutual inductance

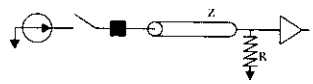


IO signals

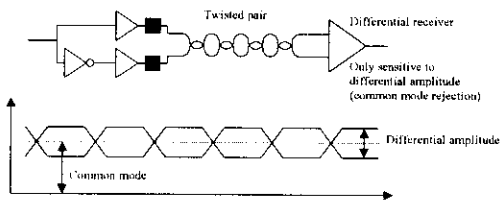
- Direct voltage mode
  - Simple driver (Large CMOS inverter)
  - TTL, CMOS, LV-TTL, etc. Problems when Vdd of IC's change.
  - Large current peaks during transitions resulting in large oscillations
- Slew rated controlled
  - Limiting output current during transitions
  - Reduced oscillations
  - (Reduced speed)



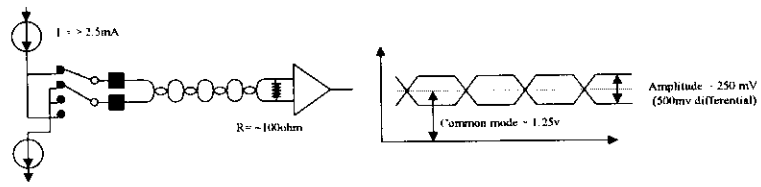
- Current mode
  - Switch current instead of voltage
  - Reduced current surge in power supply of driver
  - Reduced oscillations
  - External resistor to translate into voltage or
    Low impedance measuring current directly
  - Very good to drive transmission lines (similar to ECL)



- Differential
  - Switch two opposite signals: signal and signal inverted
  - Good for twisted pairs
  - Common mode of signal can be rejected
  - Two pins per signal required
  - High speed

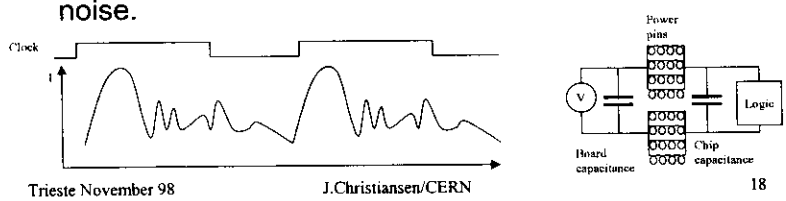


- LVDS (Low Voltage swing Differential signaling)
  - High speed (up to 250 MHz or higher)
  - Low voltage (independent of Vdd of different technologies)
  - Differential
  - Current mode
  - Constant current in driver power supply (low noise)



Power supply

- Power supply current to synchronous circuits strongly correlated to clock
- Large current surges when normal CMOS output drivers change state
- Inductance in power supply lines in package.
- 10% - 25% of IC pins dedicated to power to insure on-chip power with low voltage drop and acceptable noise.



Good design practices (What not to do)

J. Christiansen,
CERN - EP/MIC
Jorgen.Christiansen@cern.ch

Purpose of good design practices

- Improve chance of chip working first time
- Reduce (total) design time
- Reduce development cost
- Improved reliability
- Improved production yield.
- Follow vendor rules to get standard guarantees.
- Some performance reduction may have to be accepted
- (Be smart but not to smart)

Specification

- Specification must be complete before starting to do detailed design.
- Specification must be agreed upon and "signed" by involved partners
- Specification must be exact and leave no space for different interpretations
- Specification should be simulated at system level
- HDL specification is preferable.
- Realistic guesses of design time, design costs and production costs must be made (factor 2).

Choice of technology

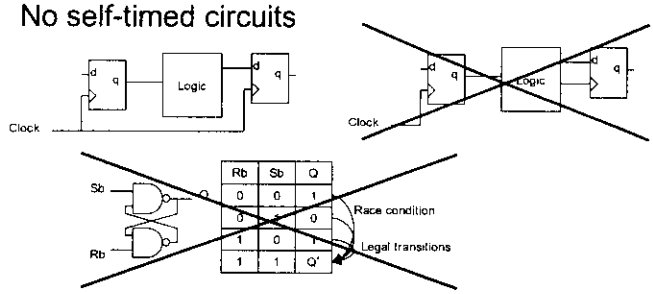
- Performance (speed, complexity)
- Design tools : Synthesis, P&R, etc.
- Libraries (gates, adders, RAM, ROM, PLL's, etc.)
- Development costs
  - Full engineering run: NRE
  - Multi Project Wafer (MPW)
- Life time of technology
  - Modern CMOS only have a life time of ~5 years
- Production
  - Price as function of volume
  - Production testing

Well planned design hierarchy

- The hierarchy of a design is the base for the whole design process.
  - Define logical functional blocks
  - Minimize connections between branches of hierarchy
  - Keep in mind that Hierarchy is going to be used for synthesis, simulation, Place & route, testing, etc.
- Define architecture in a top-down approach
- Evaluate implementation and performance of critical blocks to see if architecture must be changed.

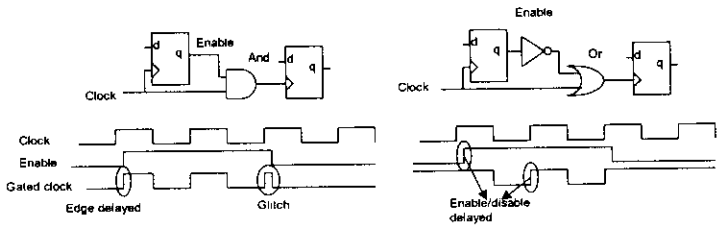
Synchronous design

- All flip-flops clocked with same clock.
- Only use clocked flip-flops
  - no RS latches, cross coupled gates, J-K flip-flops, etc.
- No asynchronous state machines
- No self-timed circuits



Clock gating

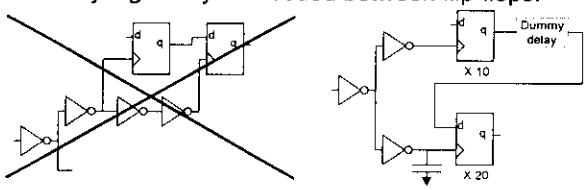
- Clock gating has the potential of significant power savings disabling clocks to functions not active.
- Clock gating introduces a significant risk of malfunctions caused by glitches when enabling/disabling clock



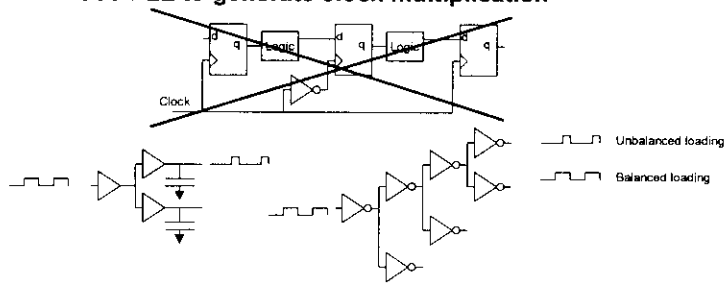
- It may be required to use clock gating on timing control signals to on-chip RAM:
  - Write enable is often used to latch address on rising edge and data on falling edge
  - Simulate very carefully circuit generating write-enable pulse.

Clock distribution

- Even in synchronous designs, race conditions can occur if clock not properly distributed
  - Flip-flops have set-up and hold time restrictions
  - Clocks may not arrive at same time to different flip-flops.
  - Especially critical for shift registers where no logic delays exists between neighbor flip-flops.
  - Clock distribution must be very carefully designed and dummy logic may be needed between flip-flops.

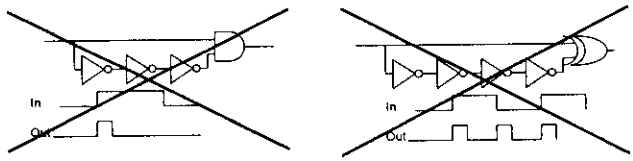


- Use of both rising and falling edge of clock
  - Doubling effective speed of circuits
  - Strict requirements to clock duty cycle from external source
  - Duty cycle distortion in clock distribution
  - **Use PLL to generate clock multiplication**



Delay based circuits

- Pulse generator
- Clock doubler

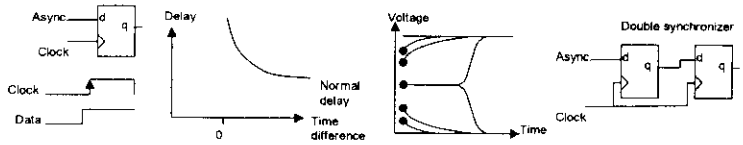


Resets

- Asynchronous resets must still be synchronized to clock to insure correct start when reset released
- Synchronous reset made by simple gating of input

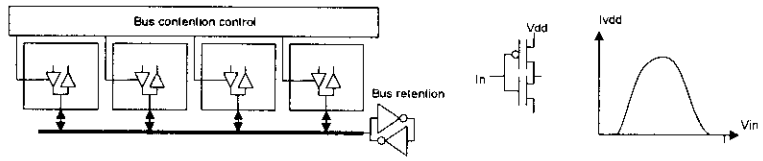
Interface to asynchronous world

- It is in some applications necessary to interface to circuits not running with the same clock.
  - Natural signals are asynchronous
  - Signals between different systems
  - Many chips today uses special internal clocks (eg. X 2)
- Asynchronous signals must be synchronized
  - Synchronizers are sensitive to meta-stability
  - Use double or triple synchronizers



On-chip data busses

- Data busses are often required to exchange data between many functional units.
  - Insure that only one driver actively driving bus
    Also before chip have been properly initialized
    Bus drivers are often power full and a bus contention may be destructive.
  - Insure that bus is never left in a tri-state state.
    A floating bus may result in significant short circuit currents in receivers
    - Always have one source driving the bus
    - Use special bus retention generators.



Mixed signal

- Extreme care must be taken in mixed analog - digital integrated circuits to limit coupling to the sensitive analog part.
  - Separate power supplies for analog and digital
  - Guard ring connected to ground around analog blocks
  - Separate test of analog and digital (scan path)
  - Use differential analog circuits to reject common mode noise
  - Be careful with digital outputs which may inject noise into analog part (use if possible differential outputs)

Simulation

- Simulation is the most important tool to insure correct behavior of IC.
  - Circuit must be simulated in all possible operating modes
  - Digital simulator output should not only be checked by looking at waveforms
  - Circuit must be simulated under all process and operating conditions
    - Best case: -20 deg. , good process, Vdd + 10% x ~0.5
    - Typical: 20 deg., typical process, Vdd x 1.0
    - Worst case: 100 deg., bad process, Vdd - 10% x ~2.0
    - Worst N - best P: NMOS bad process, PMOS good process (analog)
    - Best N - worst P: NMOS good process, PMOS bad process (analog)

Testing

- One can "never" put to much test facilities in chips.
- Put scan path where ever possible.
- Have special test outputs which can be used for monitoring of critical circuits.
- Put internal test pads on special tricky analog circuits.
- If in doubt about critical parameters of design make it programmable if possible.
- Do not forget about production testing.
- Do not make a redesign before problems with current version well understood.
- Most designs needs some kind of redesign.