UNITED NATIONS EDUCATIONAL SCIENTIFIC AND CULTURAL ORGANIZATION
INTERNATIONAL ATOMIC ENERGY AGENCY
INTERNATIONAL CENTRE FOR THEORETICAL PHYSICS
I.C.T.P., P.O. BOX 586, 34100 TRIESTE, ITALY. CABLE: CENTRATOM TRIESTE

The United Nations University

SMR/943 - 5

ICTP-UNU-MICROPROCESSOR LABORATORY:
FOURTH COURSE ON
BASIC VLSI DESIGN TECHNIQUES
18 November - 13 December 1996

CMOS TECHNOLOGY

Jose FOLGUERAS
Central Institute for Digital Research
Calle 120 No. 1700 B/W
17 y 19 Rpto.
Siboney
12100 Havana
CUBA

CMOS Technology

Introduction to VLSI Fabrication Technology.

José Folgueras Méndez

INTRODUCTION TO VLSI FABRICATION TECHNOLOGY

SOME HISTORY FOR A START.

Technical progress has been characterized almost always by the appearence and availability of new discoveries. But no discovery is at hand until it is usable. Technology has always been responsible for this and still is. A well-known example is the invention of the wheel, which made it possible for early man to achieve new transports, to communicate and widen the scope and field of action of his life.

A litte history won't do any harm and will help to clarify the important role played by technology in the field of integrated circuit fabrication. Eventhough many technologies have surged in the last twenty-five years, they may be classified in three major groups, namely Bipolar, CMOS and GaAs technologies.

If we were to point out the cornerstone of these IC technologies we should note that the bipolar transistor and the field-effect transistor (FET) made it possible to think, years later after their practical birth, of integrated circuits. This can be illustrated with the following text:

"Undoubtedly, the integrated circuit is the key to the wide-ranging exploitation of the new computer techniques. Television and radio receivers designers also will resort to the use of linear integrated circuits to reduce costs, but the transition will be gradual because of the more critical linear circuit yield control problem".

"Using the revolutionary integrated circuit techniques we are on the threshold of computer design and fabrication advances which will make the application of computer control generic to tens of thousands of new processes and procedures arising in our modern scientific culture". These premonitory words were due to Lynn, Meyer and Hamilton and time has shown that they were, in a way, a sharp although faint prediction of the future to come [1].

I feel urged to remind the reader that the invention of the FET precedes by many years the work of Bardeen, Brattain, Pearson and Shockley in the late 1940's, which resulted in the invention of the point-contact and bipolar transistors.

Back in October 1925 and 1926 J.E. Lilienfield filed patent applications entitled "Method and Apparatus for Controlling Electric Currents" [2]. Later on, in 1935, O. Heil also described a thin film field-effect device with one or two control electrodes or gates [3]. Nevertheless, it wasn't until 1959 when experiments by Atalla et al [4] showed the feasibility of passivating silicon surfaces with grown silicon oxide layers. Kahng and Atalla [5, 6] proposed a Si-$Si0_2$ structure with a metal gate, giving thus birth to the MOS Metal-Oxide-Semiconductor field-effect transistor.

At the same time another form of field-effect device, using a very thin polycristalline semiconductor layer was being developed at the RCA Laboratories by Weimar [7] and his collaborators: the so-called thin-film field-effect transistor (TFT).

The results of these discoveries and inventions made it possible to develop, in the mid sixties, the first bipolar integrated circuit. One after the other, new products have been coming into life since this time, characterized by the increase in complexity and number of functions, "integrating" in a single semiconductor chip a large number of devices. Thus, the term integration came into being, as a means of identifying the total count of individual devices in an integrated circuit.

Acronyms have popped-up like wild flowers at the start of Spring. With one of them, VLSI (Very Large Sale Integration) are deeply related the lectures we begin today. VLSI design has proved itself to be one of the pillars upon which rises, with increasing force and importance, modern Electronics.

So let's dig into some aspects of VLSI design. Here we go!!

REFERENCES

1.- Lynn D.K., C.S. Meyer, D.J. Hamilton: "Analysis and Design of Integrated Circuits", Mc Graw-Hill Book Co, MOTOROLA Series in Solid-State Electronics, 1967.

2.- Lilienfield, J.E.: "Method and Apparatus for Controlling Electric Current", US Pat. 1,745,175, Jan. 28th. 1930, Canadian Application filed Oct. 1925, US App. filed Oct. 1926.

3.- Heil, O.: "Improvements in, or Relating to, Electrical Amplifiers and other Control Arrangements", UK Pat. 439,457 Dec 1935, Appl. filed March 1935.

4.- Atalla, M.M., E. Tannebaum, E.J. Scheibner: "Stabilization of Silicon Surfaces by Thermally Grown Oxides" Bell Syst. Tech. J., v.38, 749-783, May 1959.

5.- Kahng, D., M.M. Atalla: "Silicon-Silicon Dioxide Field Induced Devices", Solid-State Dev. Res. Conf., Pittsburgh, Pa. June 1960.

6.- Kahng D: "Electrical Field Controlled Semiconductor Devices", US Pat. 3,102,230, Aug 1963, App. filed May. 1960.

7.- Weimer, P.K: "An Evaporated Thin-Film Triode", IRE Trans. Electron Devices ED-8, 421 Sept. 1961.

Some words on technology

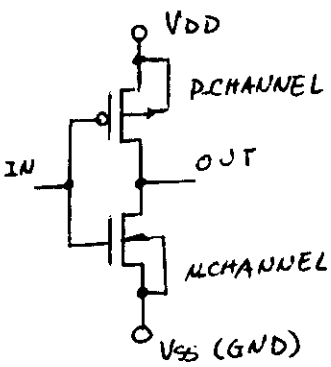
VLSI wouldn't exist if it were not for the existence of technologies. It's not easy nowadays to lecture on technologies: it would demand considerable time. The best choice seems to be to introduce one or two major technologies, thus illustrating how to reach the final product; i.e. the integrated circuit.

CMOS Process

The main issue in CMOS devices is to produce pairs of NMOS and PMOS transistors, as well as any other needed device and interconnections.

The starting point is a Silicon wafer, 4" to 6" in diameter and thickness of several hundred microns.

A pair of NMOS and PMOS transistors, connected as shown, form an inverter, which consumes no static power, occupies little area of silicon and shows an excellent reliability.



CMOS (Complementary MOS) manufacture employs different procedures to achieve the final structure on the Si wafer. This structure is composed by layers of different materials, semiconductors, dielectrics and metals.

The adequate combination of materials, thicknesses, impurity concentration, properties, etc., bring out devices with known and controllable characteristics.

There should be no mistake thinking that semiconductor technology is some kind of cooking affair.

Even though you've got the technology on paper, even though you buy Know-How, it's no matter of "add a little salt and pepper".

Integrated circuit manufacturing demands highly-skilled operators, highly-specialized and precise pieces of equipment, high purity materials and carefully and precisely controlled processes.

You don't really have the technology until you are in a position to bring out, every day of the year, individual devices and integrated circuits with controlled and predictable characteristics.

It's quite a wise and clever decision to make use of somebody else's well-established technology instead of aiming your efforts - and your time and money as well - towards establishing your own. On the long run it simply doesn't pay back.

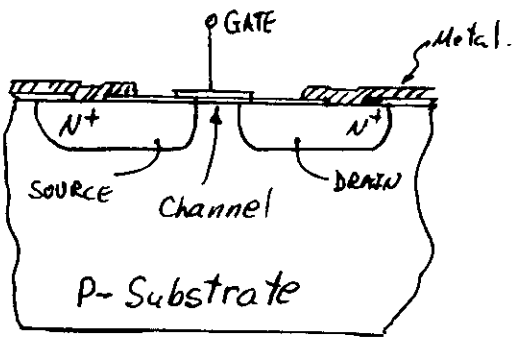
Processes used in Integrated Circuit (IC) Manufacturing.

Processes used in IC manufacturing may be of one or more of the following general kinds:

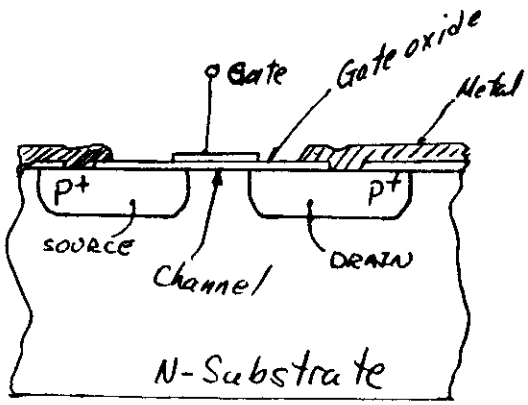
- Physical
- Chemical
- Physical and Chemical

They may be wet or dry (using or not liquid reagents) and may be classified regarding, perhaps, other characteristics.

The general and simplest structure for an MOS transistor is illustrated below for an n-MOS and a p-MOS transistor.



N-Channel MOS Transistor



P-Channel MOS Transistor

Since C-MOS (Complementary MOS) technology must produce both N-MOS and P-MOS transistors on the same wafer there must be some processes which permit to accomplish this.

In fact, "pockets" or "wells" of different conductivity than that of the substrate are created in the substrate, so you have two different substrate conductivities on the same wafer. We will go further into this topic in a short while.

Again, regarding processes, let's note down the main ones:

Oxidation

Silicon Oxide is formed when the surface of the wafer, held at a sufficiently high temperature, is placed in contact with the oxidizing species (Oxygen from steam or from gas sources have different oxide growth rates)

Masking

Some layer used to make a process locally. For instance, SiO_2 is used as a mask for Ion implantation, while Si_3N_4 is good for masking oxidation.

Ion implantation

Impurity ions are supplied enough energy as to go inside the unprotected areas of the silicon surface. Varying the energy varies the penetration of the impurity. Implant in a p-type substrate with As (Arsenic, a donor impurity) or P (Phosphorus) can convert zones of p-type Si into n-type Si, thus creating a "well", "pocket", or "tub". It's a low temperature process, although annealing at some higher temperature is necessary to take away the structural damage created by ions being implanted

Diffusion

A high-temperature (900C - 1100C) process by means of which impurity atoms at the surface, or in a shallow layer beneath the surface, of Si are driven inside bulk Silicon.

Layer deposition

A low-temperature process (400 C - 600 C) that permits to obtain layers of different materials, i.e. insulators like SiO_2, Si_3N_4 (Silicon Nitride), poly-Silicon (poly-crystalline Silicon) etc. May be of chemical (CVD Chemical Vapor Deposition), PCVD (Plasma CVD) or physical (Sputtering) natures.

Etching

Used to etch away undesired layers locally. For instance, to open a window (hole) in a Si_3N_4 mask so as to oxidate the Silicon Surface locally, or to open a contact window on SiO_2, so a metal can make contact with a diffused region.

Etching maybe wet like in the following cases:

SiO_2 — removed by BHF (buffered HF),

Si_3N_4 — removed by hot (70-80°C) hot H_3PO_4,

or dry when plasma etch is used. In plasma etch Ar^+ ions are accelerated and impinge on the surface to be removed. Some chemical means, in gaseous form, may be present to enhance the removal of unwanted material. Since there is a certain rate of removal, depending on the material, you obtain a certain selectivity while etching. This means etching speed may be quite different for different materials.

Metalization

To connect individual devices with metal, contacts defined by the contact hole mask are etched from the surface to the layer or layers to be connected below. An Al (aluminum) deposition — usually by vapor or ion beam techniques — covers the surface and fills up the contact hole. A mask for metal interconnect pattern, is then used to define where Al is to be etched away and where it is going to connect two or more devices, nodes, etc.

More than one metal layer can be deposited, separated by layers of Si_3N_4 or polymide. Up to three metal layers have been used in practical applications.

Interconnections can also be made by employing the so-called Silicide (salicide) Technology. Interconnects are form with a silicide, reacting a refractory metal (Ti, Mo, W) with polysilicon (polycrystalline Si).

Passivation

Passivation is intended to cover the outer surface of the chip in order to protect it mechanically. It is accomplished with a final layer of polymide or any other insulator

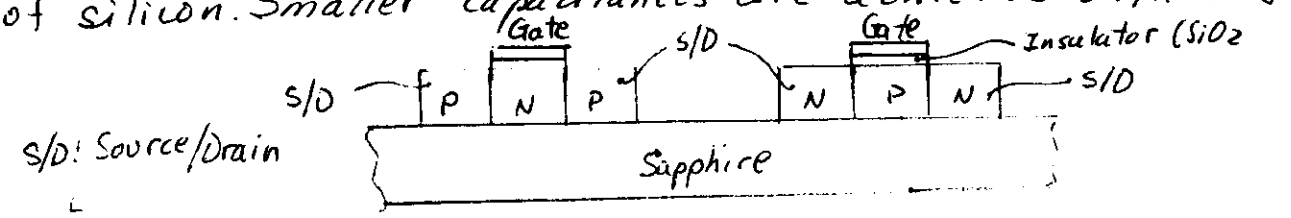
Due to the irregular surface of the circuit, a process sometimes called planarization is introduced. It consists of depositing a thick layer of PSG (Phosphosilicate glass), i.e. SiO_2 with a high P content which flows when treated at high temperatures, thus making less abrupt the steps that the metal layer must cross. This helps decrease metal interconnect discontinuities which lead to low yield in production

Other processes

Between any two processes of those already mentioned a lot of other operations like cleaning, drying, photoresist deposition, photolithography etc. are carried out. So in reality a whole manufacturing process is an ordered collection of quite a lot of operations, each of which can decrease the final wafer yield.

Some words on SOS

SOS: Stands for Silicon On Sapphire. CMOS/SOS is a CMOS process where the substrate is an insulator instead of silicon. Smaller capacitances are achieved with SOS



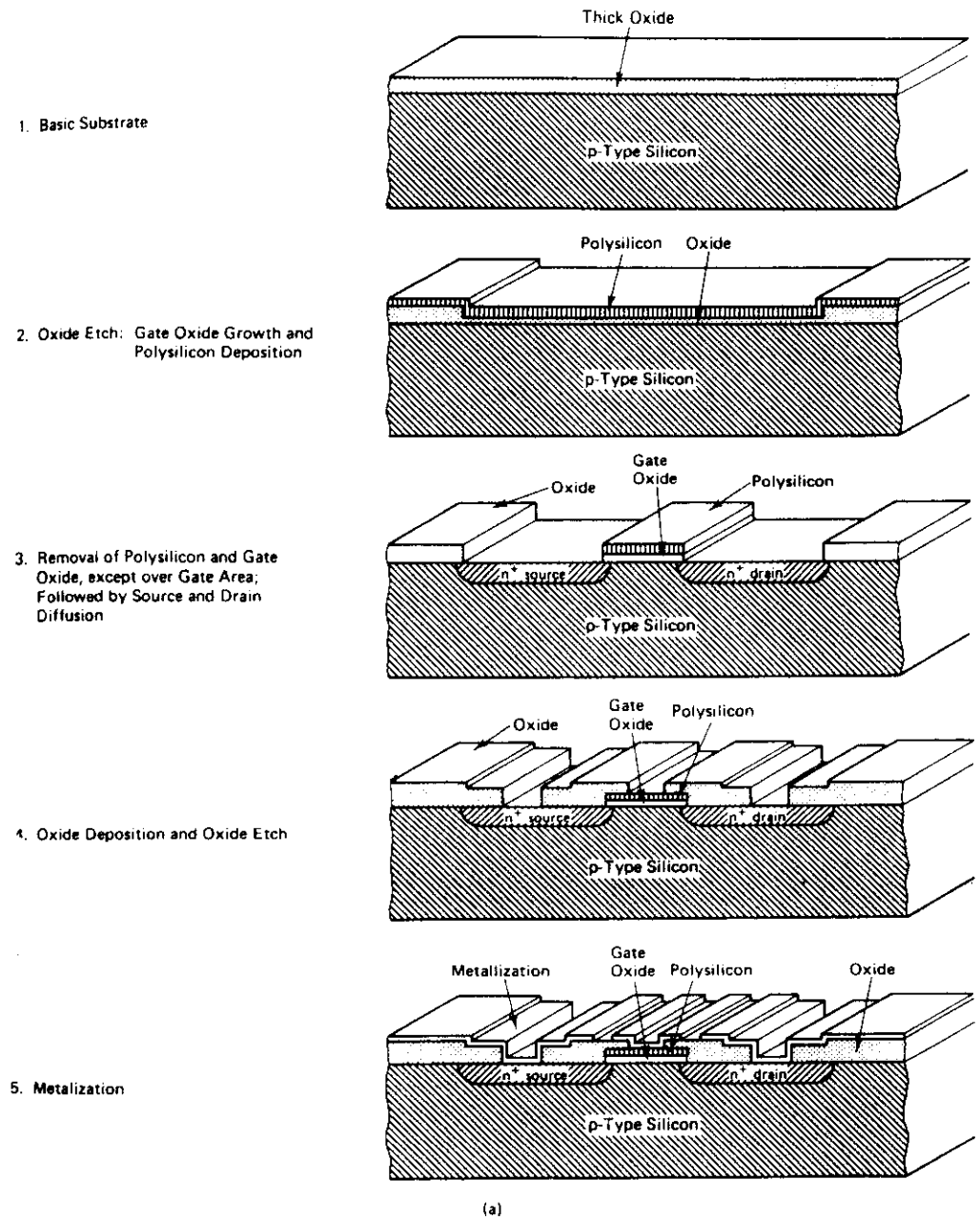
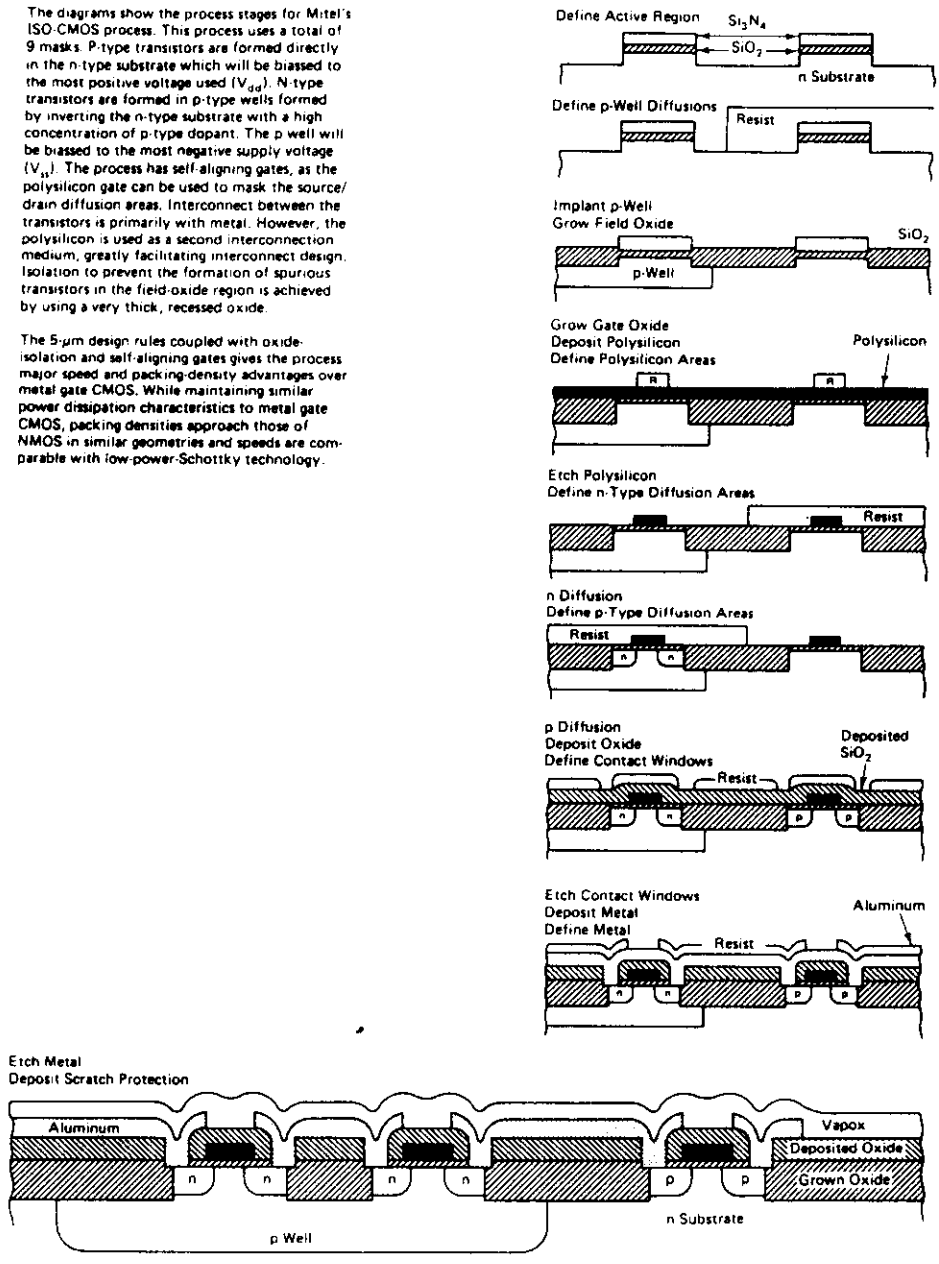


Figure 2.13 (a) Sequence of steps in silicon gate processing; (b) ISO-CMOS process of MITEL; (c) design and analysis of self-aligned gate MOSFETs, and terminology. (a) Courtesy of Integrated Circuits Engineering Corp. (b) Courtesy of MITEL.}

The diagrams show the process stages for Mitel's ISO-CMOS process. This process uses a total of 9 masks. P-type transistors are formed directly in the n-type substrate which will be biassed to the most positive voltage used (V_{dd}). N-type transistors are formed in p-type wells formed by inverting the n-type substrate with a high concentration of p-type dopant. The p well will be biassed to the most negative supply voltage (V_{ss}). The process has self-aligning gates, as the polysilicon gate can be used to mask the source/drain diffusion areas. Interconnect between the transistors is primarily with metal. However, the polysilicon is used as a second interconnection medium, greatly facilitating interconnect design. Isolation to prevent the formation of spurious transistors in the field-oxide region is achieved by using a very thick, recessed oxide.

The 5-μm design rules coupled with oxide-isolation and self-aligning gates gives the process major speed and packing-density advantages over metal gate CMOS. While maintaining similar power dissipation characteristics to metal gate CMOS, packing densities approach those of NMOS in similar geometries and speeds are comparable with low-power-Schottky technology.



(b)

Figure 2.13 (*cont.*)

An important matter is how to decrease the undesired overlap of the gate and the diffusions of source and Drain underneath. This overlap contributes undesired capacitances. One way is by means of the so-called Sidewall-spacer technology which, in turn gives rise to the Lightly Doped Drain (LDD) technology. (See below).

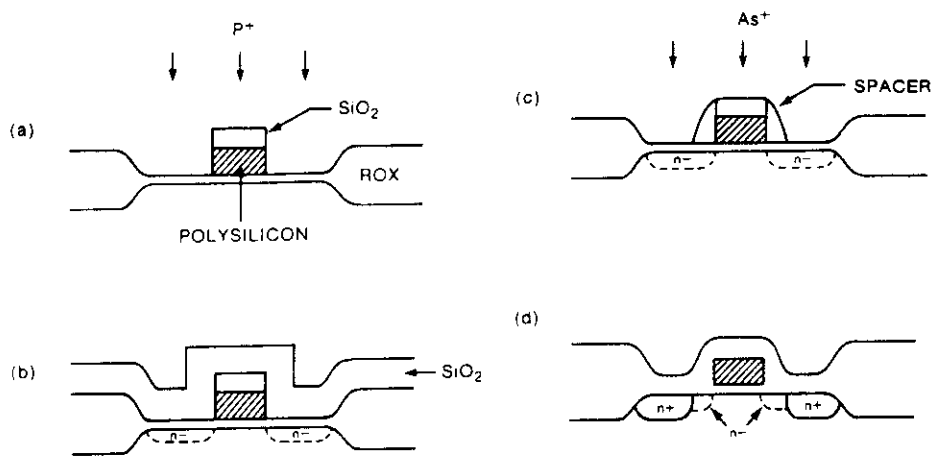


Figure 20. Sidewall-spacer technology © 1982 IEEE. [15]

" To lower the S/D resistivity a layer of metal, such as platinum (Pt) or titanium (Ti) is deposited across the whole wafer after the spacers formation as shown in Figure 21. With proper control of temperature and reaction time, the metal reacts with bare silicon only, forming a layer of silicide such as PtSi and $TiSi_2$. The unreacted metal over the SiO_2 is then removed by selective etching, leaving a layer of silicide on top of the gate and the S/D regions. After proper annealing the sheet resistance of the diffusion regions reduces from 20–50 Ω/□ down to 2–5 Ω/□. This process, which does not require an additional mask for alignment, is called *the self-aligned silicide*, or *salicide*, process. Since the spacers are used only to avoid creating a silicide bridge over the gate, its thickness is not as critical as that in Figure 20.

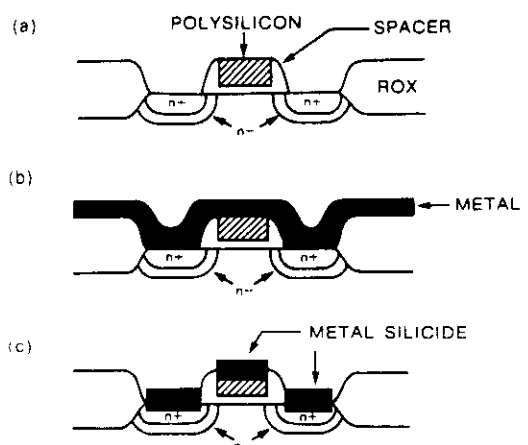


Figure 21. Salicide technology.

Polysilicon has a relatively high sheet resistance value, from 20 to 50 Ω/□ typically. Since polysilicon is used as an interconnecting layer, sheet resistance must be lowered in order to decrease delays due to transmission line effects. Silicide or salicide process accomplishes this. (See figure at left)

other hand, only the basic steps are covered in our discussion. In reality almost every step is followed by several minor or routine procedures such as annealing and wafer cleaning so as to maximize yield.

1. *Wafer Preparation and Gettering*. The process starts with a p-type wafer (substrate) with <100> orientation. The <100> orientation generates a minimum number of surface states. After scribing and initial cleaning, the wafer is subjected to gettering.

   Gettering is the process of introducing impurity traps in the substrate to absorb mobile ions that cause leakage currents. It can be accomplished by applying Ar^+ (argon) ion implant or focused laser beam to the back side of a wafer. Damages caused by this impact process will attract impurities from the surface in subsequent hot processes.

2. *N-well Definition*. In CMOS processes the n-well mask defines the n-well regions. The subsequent P (phosphorus) ion implant and drive-in creates the n-wells in which the p-channel devices will be formed (see Figure 1). The photoresist (PR) is an effective mask for ion implantation.

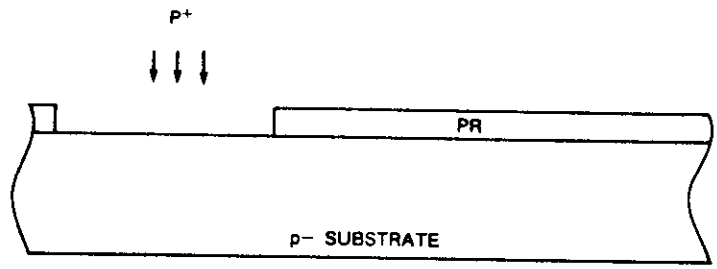


Figure 1(a). N-well ion implantation.

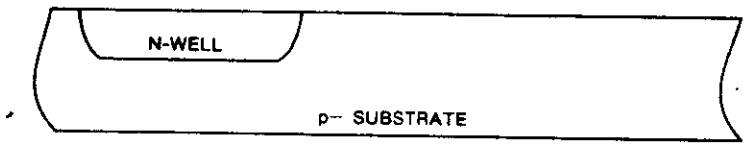


Figure 1(b). N-well formation.

3. *Gate Oxide Growth*. To grow gate oxide on devices, a layer of pad SiO_2 is first grown on the surface of the whole wafer, followed by a layer of Si_3N_4. A mask, called ROX or FOX, is used to remove the Si_3N_4 selectively as shown in Figure 2(a). The n-well mask is then used once more to apply photoresist covering all n-well regions. With the photoresist still on top of the Si_3N_4, a blanket B^+ (boron) implant is carried out. This is called *field tailoring* or *channel stopping*. It raises the threshold voltage of the parasitic n-channel transistors to be formed underneath the thick oxide.

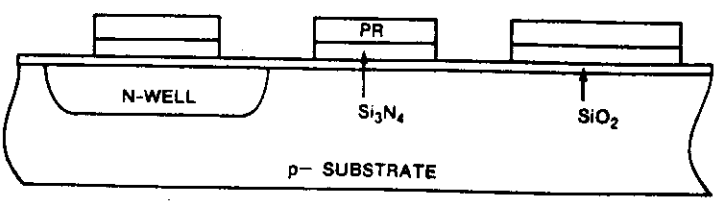


Figure 2(a). ROX definition.

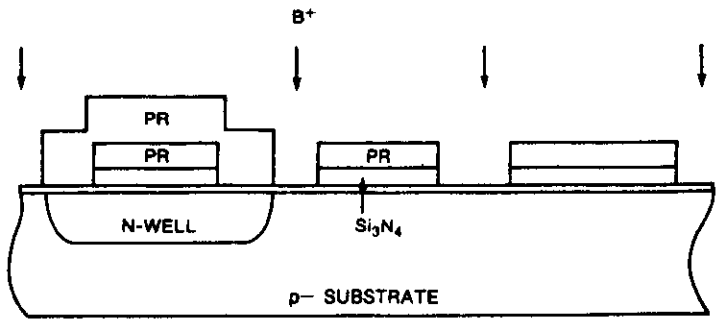


Figure 2(b). Channel stop implant.

After the PR has been stripped the wafer is subjected to oxidation. Areas covered by Si_3N_4 will remain undisturbed. Areas not protected by Si_3N_4 will be oxidized to form a thick layer of SiO_2. Si_3N_4 has very high tensile stress, so the pad SiO_2 helps relieve tension during the oxide growth. Since oxidation consumes silicon, the profile of the thick oxide is half-way "underground" as shown in Figure 3. This is called *semirecessed oxide* (SROX or simply ROX) or *field oxide* (FOX). The thick ROX forms parasitic MOS transistors with very high threshold voltages that can isolate active devices. This isolation technique is called the LOCOS (*local oxidation of silicon*) process. The semirecessed oxide also makes the wafer surface more planar.

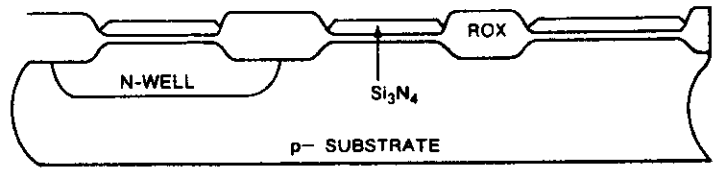


Figure 3. ROX formation.

The Si_3N_4 layer is subsequently removed by hot H_3PO_4 and the pad oxide removed by BHF (buffered HF) until bare silicon is exposed as shown in Figure 4. Right after the ROX etchback, a high-quality oxide layer is grown thermally over the whole wafer. This will be the gate oxide for all devices. The thickness and uniformity of the film must be carefully controlled.

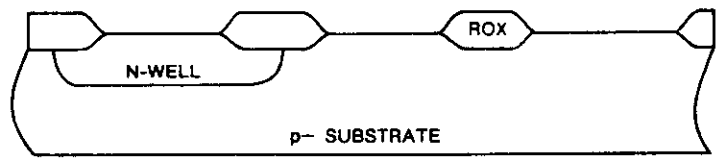


Figure 4. ROX etchback.

The wafer profile is now shown in Figure 5. Notice the bird's beak shapes extending from the ROX to the thin oxide. These formations and the boron's out-diffusion underneath the ROX are largely responsible for the large parasitic capacitances around the source/drain of the device and the so-called narrow-channel effect on threshold voltages.

Now that the gate oxide is formed, channel doping profiles of different devices such as the low-threshold enhancement mode and

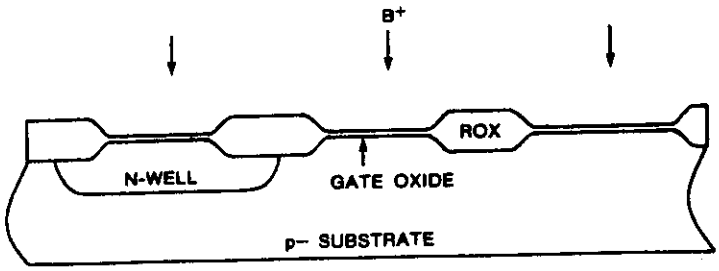


Figure 5. Gate oxide growth and channel tailoring.

depletion mode devices can be achieved by ion implantation with appropriate kinds of dopant over areas selected by device masks. For example, the B^+ implant in Figure 5 adjusts the threshold voltages of all devices. The formation of depletion mode devices is described below.

4. *Depletion Mode Device Channel Tailoring and Buried Contact Openings.* As discussed in Chapter 1, the channel of a depletion mode device is formed by a shallow n-type layer underneath the gate. As shown in Figure 6 the depletion mask generates a PR pattern with openings over the areas where the devices are to be formed. A subsequent As^+ (arsenic) ion implant drives the threshold voltage of these regions to the required below zero level.

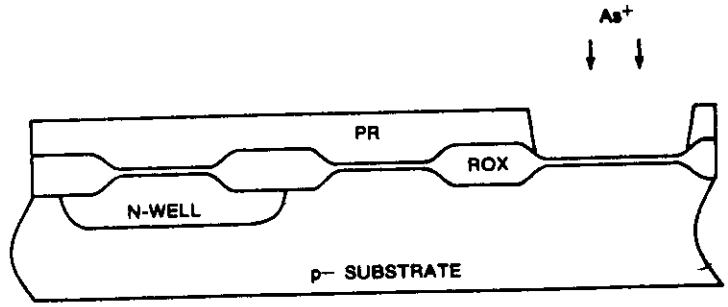


Figure 6. Depletion mode device channel tailoring.

A buried contact is a contact between polysilicon and diffusion. The buried contact mask defines the areas where the thin oxide will be removed as shown in Figure 7. After the buried contact etch, a layer of polysilicon is grown across the whole wafer with a low-pressure chemical vapor deposition (LPCVD) process.

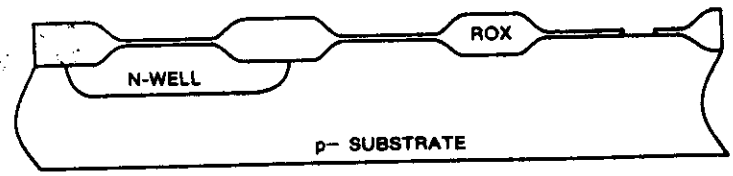


Figure 7. Buried contact opening.

5. *Polysilicon Gate Definition.* A layer of polysilicon oxide is then grown over the first polysilicon layer by LPCVD to define device gates. Both layers are then doped by P^+ ion implant as shown in Figure 8. The gate

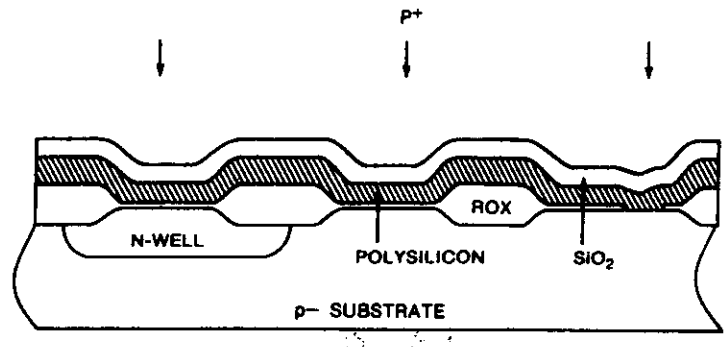


Figure 8. Polysilicon deposition and doping.

mask allows selective etching of the polysilicon oxide and polysilicon to define the gates. After that, the wafer is oxidized and a thin layer of SiO_2 covers the bare silicon areas as shown in Figure 9. Since the oxidation rate of polysilicon is three times faster than that of silicon, sidewalls of SiO_2 are formed around the gates. In a double polysilicon process, another polysilicon layer can now be deposited on top of the sidewalls. In practice up to three layers of polysilicon—P1, P2, and P3—have been used in some dynamic RAM cells.

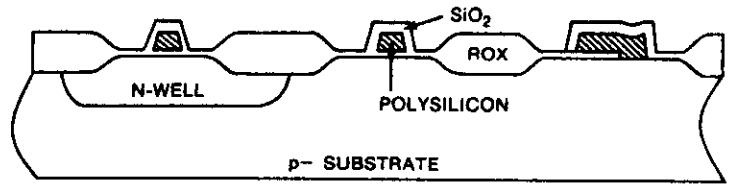


Figure 9. Polysilicon gate definition.

6. *Source/Drain (S/D) Formation.* After the device geometry has been defined, blanket implants of appropriate dopant form the S/D of devices. Since implants are stopped only by gates, the S/D of a device is actually defined by the gate itself. The process is therefore called *self-aligned silicon gate process*.

   In CMOS technology two implants are required. A blocking mask blocks the p-channel device areas when the S/D of n-channel devices are formed by As^+ implant. Similarly, a complementary mask blocks the n-channel devices when the p-channel S/D are formed by B^+ implant. See Figures 10(a) and (b).

   The wafer is then heated to 1000°C for S/D drive-in. The dopants out-diffuse toward the center of the channel, covering the gap overshadowed by sidewalls in previous ion implant. The high temperature also drives dopants in the polysilicon into the silicon through buried

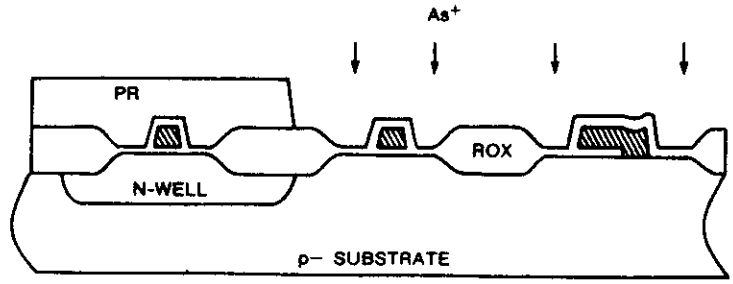


Figure 10(a). S/D implant for n-channel devices.

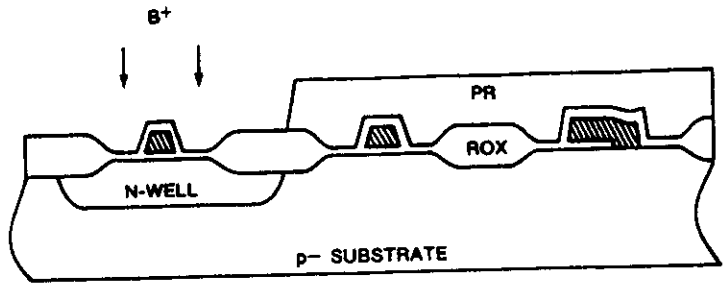


Figure 10(b). S/D implant for p-channel devices.

contact openings. In the end, the polysilicon is connected to the source of the depletion mode device. After drive-in, a layer of PSG (phosphosilicate glass) is deposited over the whole wafer. The PSG softens and flows at high temperatures. It covers the sharp steps of the polysilicon gates, creating a much smoother surface for subsequent metal films. The PSG layer also traps impurity ions such as Na^+, preventing them from contaminating the devices from above.

7. *Metalization.* To connect devices with metal, contacts defined by the contact hole mask are etched from the surface to the polysilicon and diffusion regions below. A subsequent Al (aluminum) vapor deposition covers the surface and fills up the contact holes. A mask for metal interconnect patterns is then used to remove all unused metal. In case there is more than one metal layer required, additional layers can be formed in the same fashion with composite dielectric layers of Si_3N_4 and polyimide as the insulating material. Up to three layers of metal—M1, M2, and M3—have been used in practical applications.

   A contact hole that connects metal in different layers is called a *via* (hole). A via hole placed on top of another contact hole is a *stacked via*. Depending on the metal process a stacked via may or may not be allowed in a particular technology. Figure 11 shows profiles of devices completed with M1 contacts.

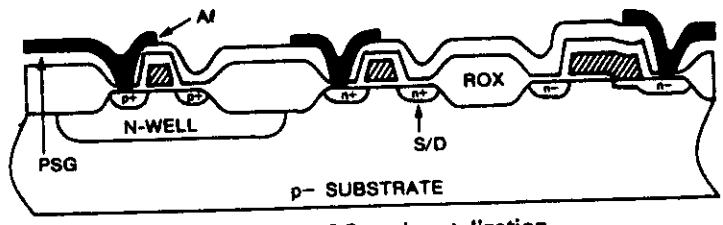


Figure 11. PSG and metalization.

8. *Passivation.* Finally, the wafer is coated with a layer of passivation material such as SiN or polyimide for protection.

2. PROCESS AND DEVICE DESIGN CONSTRAINTS

To optimize circuit performance, various requirements are demanded of device parameters. This, in turn, imposes constraints on the process. Since, in practice, circuit optimization does not lead to a consistent set of device parameters, the final selection of a process is always a compromise among conflicting factors.

VOLUME 11 NUMBER 5
MAY 1973

THE PHYSICS TEACHER

"BUT WE JUST DON'T HAVE THE TECHNOLOGY TO CARRY IT OUT."

Briefly speaking, just in order to finish up with this topic, luckily enough you can usually design resorting to a manufacturer without no need to fight against the Dark Monster of Technology.

Bear in mind that you can design on the basis of known values of the individual electrical and time parameters your manufacturer's techaology produces.

Design techniques

Three major design techniques are in common use:

- Full custom design
- Standard cells
- Gate arrays

* Full custom design offers the best performance and the least real state (silicon area). Requires experience from the designer and takes time and a work team to carry it out. The process requires the design of each transistor in the circuit. Unreal to think of a complex design in a short time without an experienced team supported by a powerful CAD (Computer Aided Design) system. All photolithographic masks must be fabricated. Highest cost and best performance. Quite long turn-around.

* Standard cells

Uses predesigned complex cells, optimized in area and parameters. Offers a smaller turn-around, small design time, medium costs and doesn't require of so much experience as for full custom design.

* Gate arrays

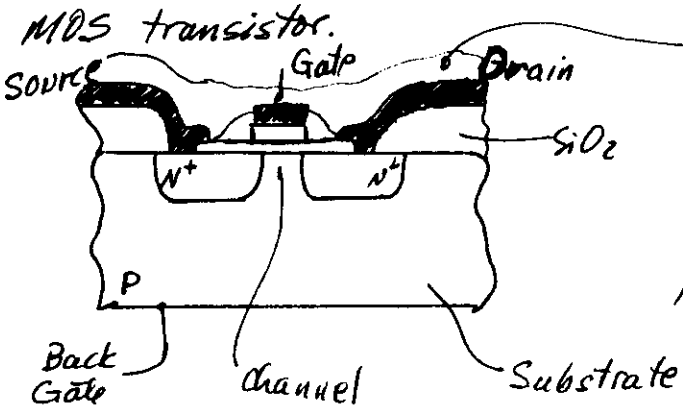
Quick design possible by designers with little experience, very good for prototyping and later optimization via full-custom design. Lower performance than full custom or standard cells, but lower cost as well.
We'll look in more detail at gate arrays in a few lectures.

Remember That

- Full custom demands more knowledge on Microelectronics and experience in the field, which is not the case for gate arrays
- Gate arrays give you the fastest turnaround, full custom the slowest
- Full custom is intended for high productions. Gate arrays for smaller ones.

The MOS Transistor

Let's review for a moment the basic structure of the MOS transistor.

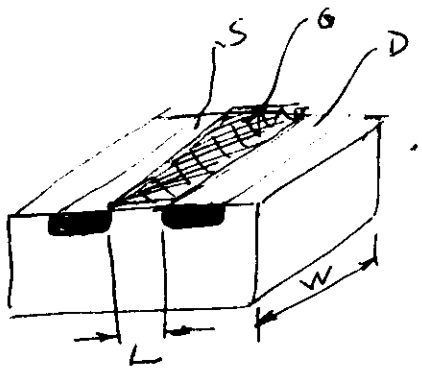


A pair of N-type diffused regions is formed on top of a P-type substrate (for N-channel MOS devices).

There are two distinct types of MOS transistors regarding the channel conductivity:

- N-type transistors (also N-channel)
- P-type transistors (also P-channel)

MOS transistors may be also classified regarding some specific characteristic, for instance channel length or width. Definitions of channel length L and channel width W are illustrated in the oversimplified representation of the structure of an MOS transistor shown in the figure.



So, considering L and W we might talk about

- Long channel transistors,
- Short channel transistors,
- Wide channel transistors and
- Narrow channel transistors

But There's another important classification considering The type of channel formed.
Recalling that the channel is formed in between two diffused zones of the same type of conductivity (Source and Drain) and on the surface of the substrate - different type (opposite) of conductivity There's no doubt that the channel must be - in order to exist - of the same type of conductivity Source and Drain are.
There are two distinct possibilities, referring to the previous figure:

1- In someway we "build" a channel joining, thus connecting S and D. Here we have what is sometimes referred to as a built-in channel MOS transistor

2- We make a channel or conducting layer appear between S and D, This is accomplished by The voltage applied to the Gate. If This voltage is of adequate polarity and value it may "invert" The substrate conductivity at The surface between S and D, creating The desired channel.
This polarity must be

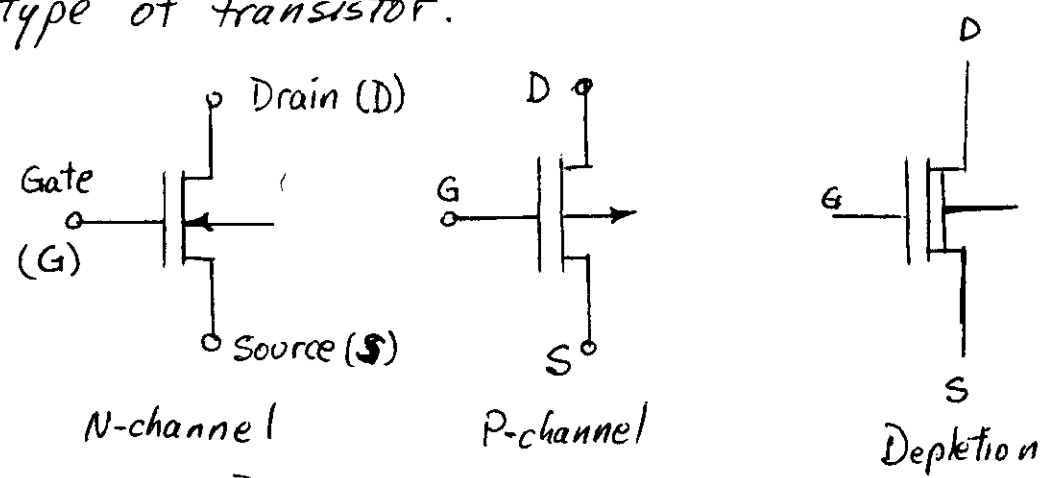
$V_{GS} > 0$ for N-channel devices
$V_{GS} < 0$ for P-channel devices

The first type of transistor ("built-in channel") is called a Depletion device, regarding to what happens to majority carriers in the channel. When V_{GS} is lowered (decreased), carriers are "pushed-back" thus decreasing the layer conductivity. This kind of transistor is usually used as a load in NMOS Technology. These devices are difficult to cut-off ($I_{DS}=0$).

The second type of transistor is called an enhancement device, because carrier concentration in the inversion layer is enhanced (increased) as V_{GS} increases. In CMOS both transistors in an inverter are of the enhancement type. (p-CH and N-ch). Enhancement devices are easily cut-off if $|V_{GS}| < |V_T|$, where V_T is the so-called threshold voltage

N-channel	$V_T > 0$
P-channel	$V_T < 0$

Below we include a brief representation of the symbols used in schematics for each type of transistor.



N-channel P-channel Depletion

Enhancement

Drain current in an MOS transistor

Current from S to D (or D to S) in an MOS transistor of the enhancement type can be shown to be described by the following expression:

$$I_{DS} = \pm \beta \left[(V_{GS} - V_T) V_{DS} - \frac{V_{DS}^2}{2} \right]$$ in the linear region when $V_{DS} < V_{DS\,saturation}$

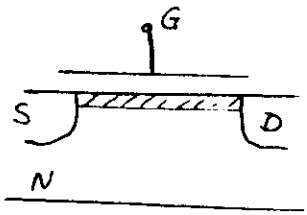
where $\beta = \frac{\mu C_o W}{L}$, μ is the mobility of majority carriers and C_o is the capacitance of the gate per unit area

$$I_{DS} = \pm \frac{\beta}{2} \left[(V_{GS} - V_T)^2 \right]$$ in the saturation region when $V_{DS} \geq V_{DS\,saturation}$

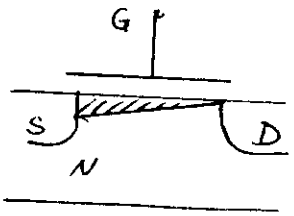
and $V_{DS\,sat} = V_{GS} - V_T$. The (+) sign in I_{DS} is for N-channel, the (−) sign for p-channel. Polarities are as follows:

	V_{GS}	V_T	V_{DS}	I_{DS}
N-channel	>0	>0	>0	>0 (into D)
Pchannel	<0	<0	<0	<0 (out of D)

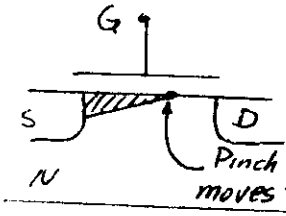
Equations for I_{DS} predict constant current in saturation which is not what you get in practice. The channel takes different shapes as shown:



$V_{DS} = 0$, $I_{DS} = 0$
$V_{GS} > V_T$

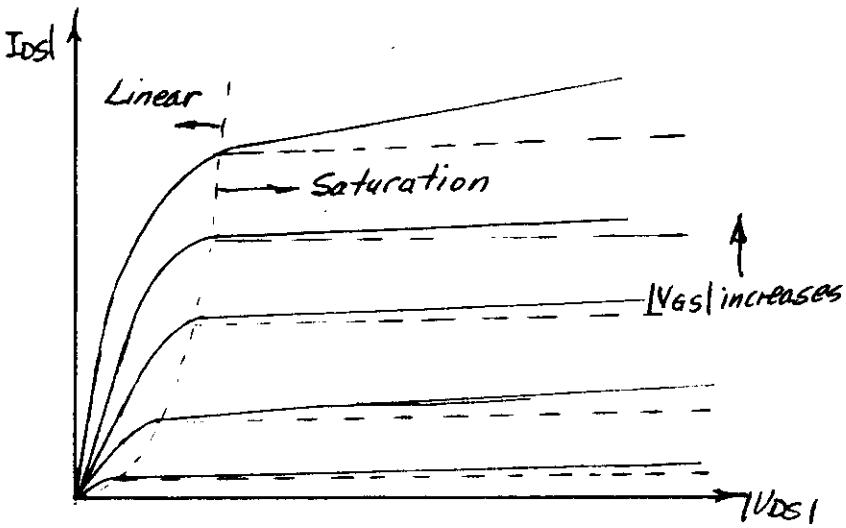


$V_{DS} > 0$
$V_{DS} = V_{DS\,sat}$
$I_{DS} \neq 0$



$V_{DS} > V_{DS\,sat}$
$I_{DS} \neq 0$

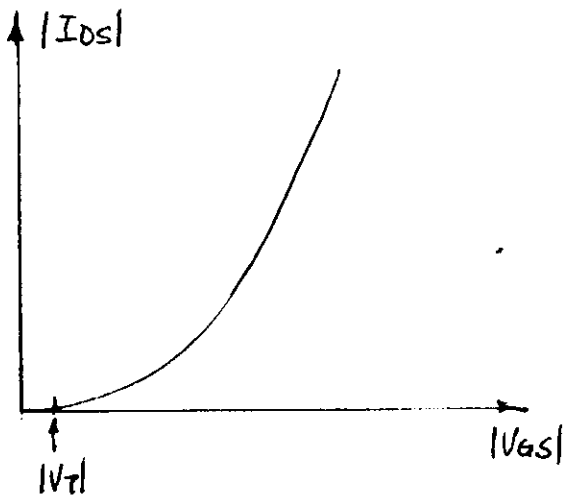
Output characteristics

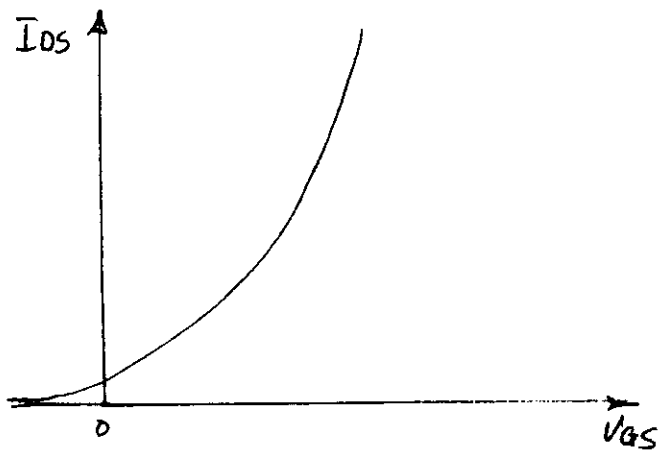


In the set of curves at left dotted lines represent ideal characteristics. Full lines are real output curves. The difference is accounted for by the so-called channel-shortening effect.

This effect consists in a movement of the pinch-off point (see figure in previous page) towards the Source. As I_{DS} increases, the pinch-off point moves due to the voltage drop along the channel.

Transfer characteristics





N-type Depletion device

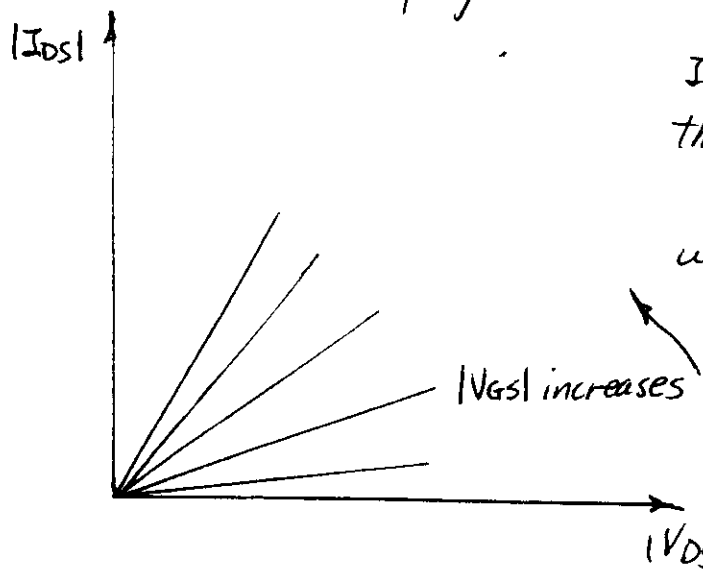
Remember that, for enhancement devices the D-S current is turned off if $|V_{GS}| < |V_T|$. For instance, you might think of the MOS transistor being a switch, whose OFF state is controlled by V_{GS}.

This, in fact, is a weird switch, which shows an ON resistance that is a function of applied voltages.

In analog circuits this is, of course, an important characteristic since devices don't usually function like switches but as variable resistors (for enough low values of V_{DS}, while in the linear region) or as a transistor, with corresponding small-signal A.C parameters, in the saturation region.

For digital circuits, where you expect the device to "move" from a non-conducting to a conducting state and viceversa This varying ON resistance influences the time response of the device.

Finally, to illustrate the use of an MOS device as a variable resistor let's make a zoom of the region around the origin of coordinates for the output characteristics shown in the previous page.



If $V_{DS} \ll V_{GS}-V_T$

then

$I_{DS} \approx \pm (V_{GS}-V_T) V_{DS}$

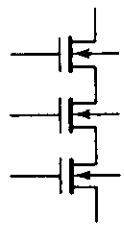
which plots like the straight lines in the figure at left.

A brief abstract on what we've seen up to now:

- Four-terminal device: source (the reference point), drain, and gate (the input)
- Source and drain are completely interchangeable physically
- Source and drain exist at opposite ends of the "channel"
- Whether the channel conducts or not depends on the gate (input) voltage relative to the source voltage
- Source and drain can be shared with adjoining XRs if done properly
- V_{dd}, drain voltage; V_{ss}, source voltage; V_{ds}, drain-to-source voltage; S, source; D, drain; XR, transistor

(a)

- "n-Channel string": the series connection of two or more NMOS transistors (XRs)



- Analogous statement holds for "p-channel string"

(b)

Figure 2.9 (a) Properties of MOS transistors; (b) the term "string."

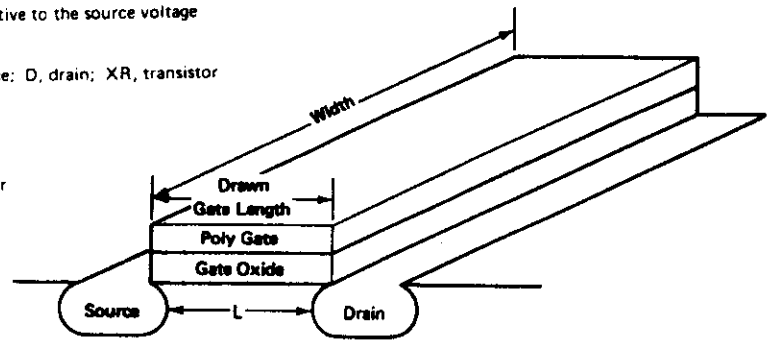


Figure 2.11 Three-dimensional picture of NFET showing width and lengths. L, Electrical length. Effective gate length (BGL, not shown) is a mathematical artifice used to account for parasitic capacitance. For each FET described in TRACAP, W, L, and BGL are specified. BGL depends on W and gate length. Current flows from source to drain if gate voltage is proper for that type of MOS transmitter. In current usage, effective gate length normally denotes electrical length.

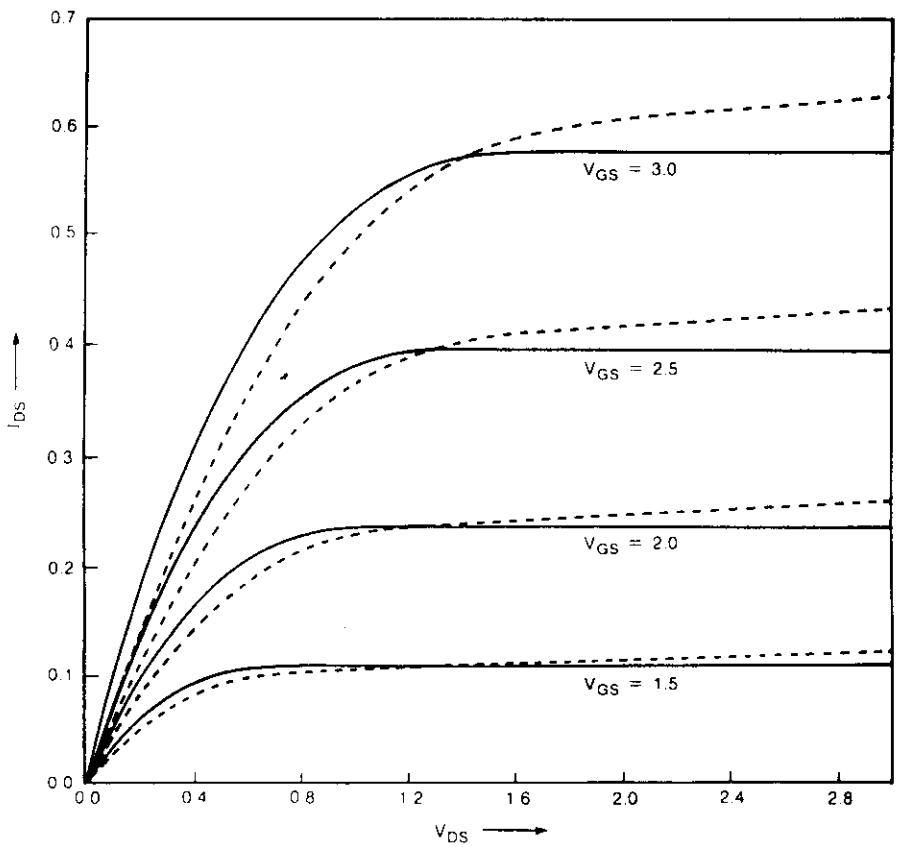
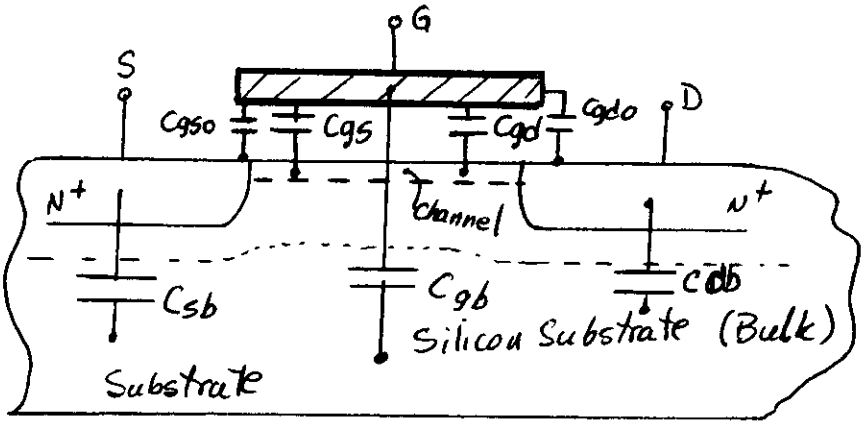


Figure 19. A typical family of I_{DS} as a function of V_{DS} with V_{GS} as a parameter. The solid curves are obtained from the model. The dashed curves represent actual data.

Parasitic elements in the MOS transistor

In the figure at right a representation is shown of the parasitic capacitances in an MOS transistor.



Capacitance values are not constant, since they vary with the condition in which operates the transistor. This fact is illustrated below

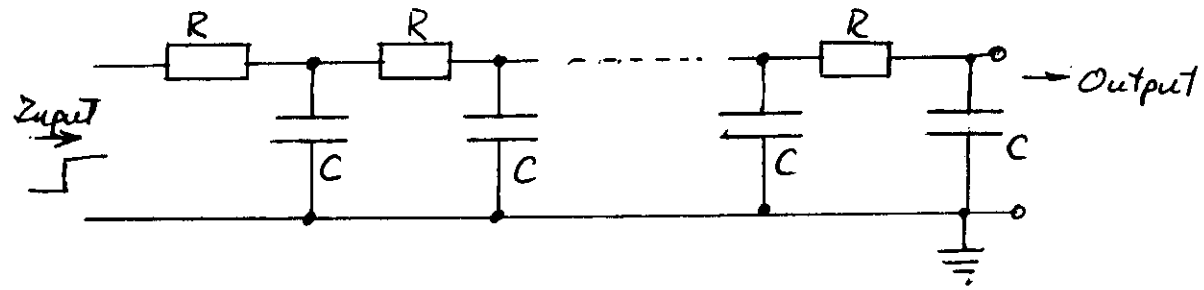
	OFF	LINEAR	SATURATION
C_{gb}	$\frac{\varepsilon A}{t_{ox}}$	0	0
C_{gs}	0	$\frac{1}{2}\frac{\varepsilon A}{t_{ox}}$	$\frac{2}{3}\frac{\varepsilon A}{t_{ox}}$
C_{gd}	0	$\frac{1}{2}\frac{\varepsilon A}{t_{ox}}$	0
C_{gso}	$\frac{\varepsilon A_{overlap}}{t_{ox}}$	same	same
C_{gdo}	$\frac{\varepsilon A_{overlap}}{t_{ox}}$	same	same

$C_g = \Sigma C_g$

C_{db} and C_{sb} are the junction capacitances to the substrate and can be calculated taking into account the total area junction, including lateral area and junction depth.

There's another capacitance which "clings" to everyone of the electrodes of a device. Since devices must be interconnected among themselves in the integrated circuit you must use some kind of interconnecting conducting line. Two general solutions are used: metal lines and

other conducting lines i.e.: polysilicon or silicides.
Routing capacitance is, thus, formed by these lines, a dielectric (usually thick oxide) and the substrate.
But keep in mind that your interconnect has also resistance and, if your line is long (as may be for clock lines and buses) you get an RC transmission line, which can be represented as an RC distributed line like illustrated below.



The RC line introduces a delay between output and input which can be approximated to

$$t_d \approx \frac{RC \cdot l^2}{2}$$

if the number of sections is large.

In the above expression:

R = Resistance per unit length
C = Capacitance per unit length
l = Total conductor length.

Second order effects

They introduce additional corrections in the I_{DS} expression.
Short and narrow channel effects are typical examples and need proper modelling.

Simulation

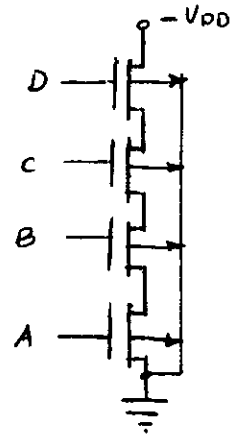
Perhaps the most widely-used electrical simulation CAD tool is SPICE from Berkeley (U of California). The name stands for Simulation Program with Integrated Circuit Emphasis. Versions running on PCs and workstations, like PSPICE, are a great help in electrical simulation.

Electrical simulation is a must if you are designing on a full-custom basis. Timing simulations must include routing capacitances and all parasitic capacitances if you don't want to be too deceived at the end.

Bulk effect

Biasing the substrate or bulk alters the threshold voltage.

In a combination like the one shown at right for P-N-P transistors, transistor A shows no bulk effect, while transistors B, C and D will show Bulk effect, due to their common substrate.



Thus, for the same driving signals and transistors with the same dimensions transistor D will show the less current. So you can't add transistors in series indefinitely

Fortunately enough, if you are on a semi-custom or a standard-cell approach, you won't have to worry about this: someone did it already.

Mobility:

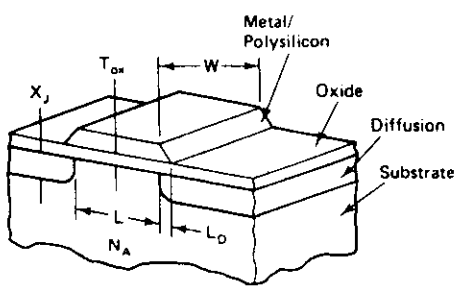
Keep in mind that factor β in the expressions for I_{DS} include carrier mobility μ. The mobility of holes is roughly one half that of electrons. Thus for equal drives and equal current an MOS transistor of p-type should be twice the $\frac{W}{L}$ ratio of the n-type you have.

Scaling

As time has elapsed, device density in integrated circuits has increased, supported by advances in technology and a better knowledge of device physics. This increase in device density has led to transistors with smaller dimensions. Second order effects have started to play their role, influencing parameters as well as characteristics of individual transistors.
Anyhow, scaling must be done according to certain rules which are included in the text that follows.

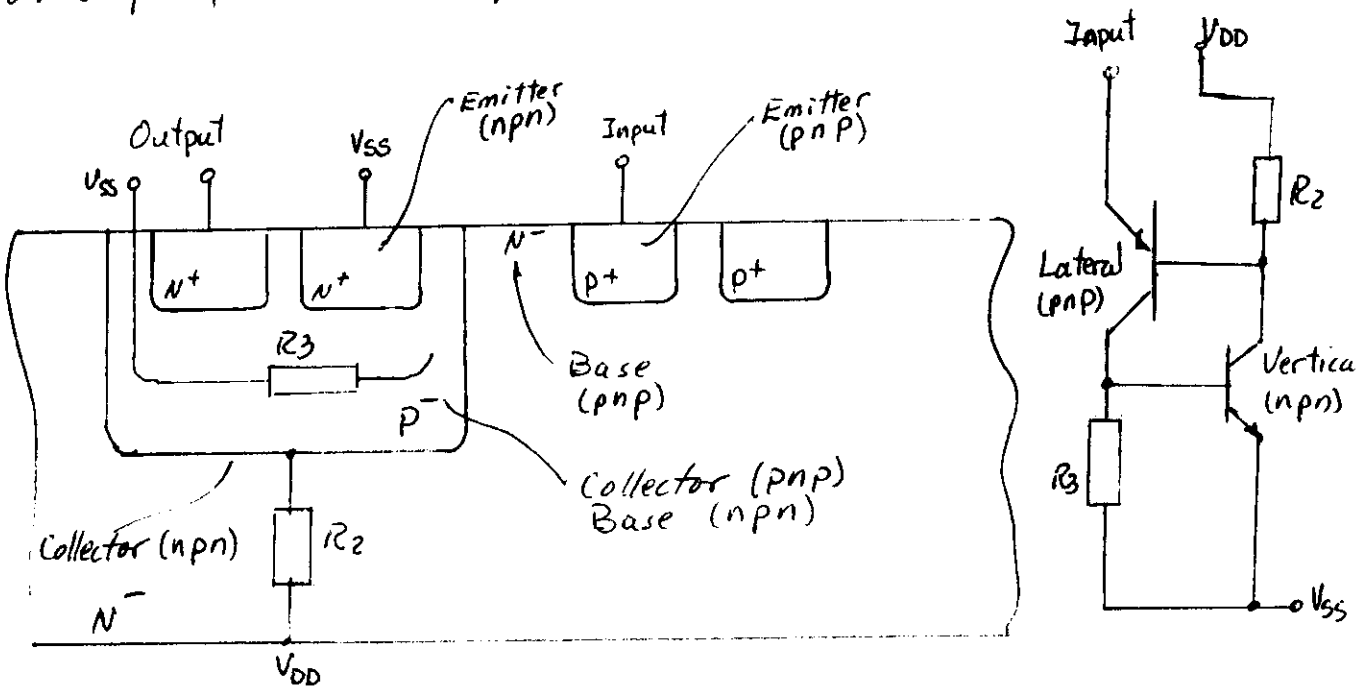
	Classical	Modified ($\alpha > 1$)
Substrate Doping N	S	S
Dimensions t_{ox}, L, W, L_D, x_j	$1/S$	$1/S$
Supply Voltage, V	$1/S$	1
Supply current	$1/S$	
Parasitic capacitance $\frac{WL}{t_{ox}}$	$1/S$	$1/\alpha S$
Gate Delay VC/I	$1/S$	$1/\alpha S$
Power Dissipation VI	$1/S^2$	1
Critical Charge $Q=CV$	$1/S^2$	$1/\alpha S$
Power-Delay Product	$-1/S^3$	$1/\alpha S$
Power density VI/A	1	S^2

Problems associated: Decreasing mobility μ as N increases
Hot electron emission into gate oxide
Punchthrough due to short channel



Latch-up

Latch-up is a property of CMOS by which the output signal stays "latched" (stuck) in its high state. It is caused by the two substrates of CMOS (the substrate and the well of opposite conductivity) creating a silicon-controlled rectifier (SCR) in the bulk of the CMOS device. An SCR can be regarded as a combination of a p-n-p and an n-p-n transistor.



The structure latches up due to increased currents through R_2 or R_3. Due to positive feedback. as current through R_3 increases the npn transistor turns on. If current through R_2 increases it can turn on the pnp transistor which will further turn on the npn vertical transistor. Also, if the lead input goes higher than V_{DD} then latch-up can occur, creating a path between V_{DD} and V_{SS}. Fortunately enough the manufacturer has taken care already of latch-up, eliminating most of the possibilities of latch-up occurence

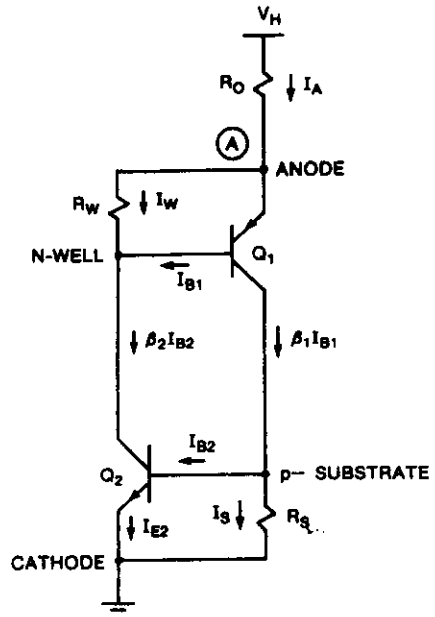
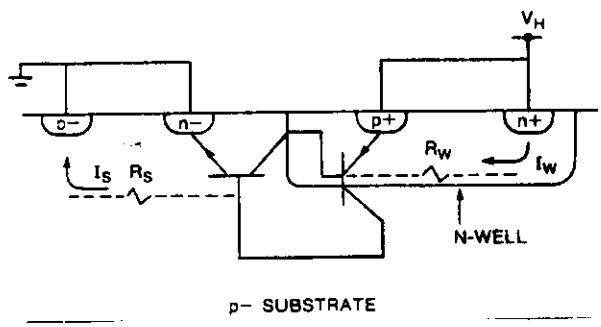


Figure 18. CMOS latchup SCR structure.

The total current I_A is the sum of I_S and I_{E2}:

$$\begin{aligned} I_A &= \frac{V_{BE(on)}}{R_S} + (1 + \beta_2)I_{B2} \\ &= \frac{V_{BE(on)}}{\beta_1\beta_2 - 1}\left[\frac{1}{R_S}(\beta_1 + 1)\beta_2 + \frac{1}{R_W}\beta_1(\beta_2 + 1)\right] \\ &= I_H \end{aligned} \tag{5}$$

Thus I_H, called the *holding current*, is the minimum current generated from the power supply to sustain the on state.



ESD : Electrostatic discharge

Electrostatic discharge, often called also ESD is a cause of failure in many integrated circuits. In order to avoid this failure cause, usually a punctured thin gate oxide in the input transistors, protections are added at the inputs. It's mandatory to do so, since unprotected inputs represent a potential risk of oxide breakdown.

2.5.4 Protection of Inputs from Electrostatic Discharge and Other Voltages

A person walking across a wool rug can develop as much as 10,000 V of static electricity if the air is dry enough. Similar voltages can be developed in a variety of other circumstances.

The effects on MOS devices of such voltages are disasterous. The gate oxides of 5-μm (drawn) gate length devices are only typically 750 Å thick. Those of 2-μm gate-length devices range from 150 to 250 Å in thickness. The tetrahederal radius of the silicon atom is 1.18 Å. Thus the gate oxides of current CMOS and other MOS devices are only 100 or more atoms thick and can easily be severely damaged by electrostatic discharge (ESD).

For this reason, virtually all MOS devices have some form of ESD protection built into them, Typically, this takes the form of diode voltage clamps in conjunction with a resistance of some kind. A cross-sectional view and a schematic of a typical input protection are shown in Fig. 2.24(a) and (b), respectively.

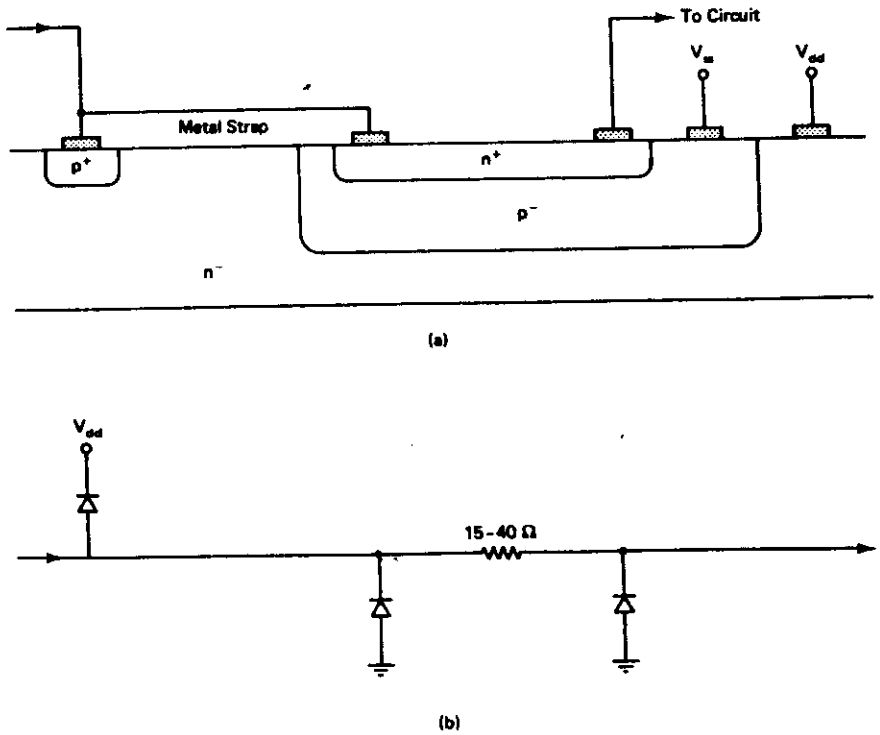


Figure 2.24 Input protection circuit: (a) cross section; (b) schematic.

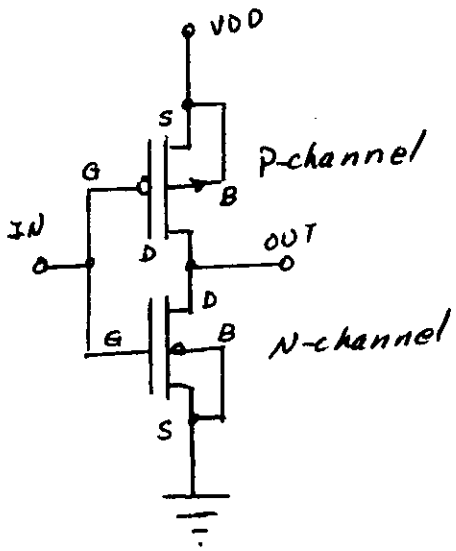
Inverters and Transmission Gates

The inverter is one of the simplest structures from the circuit point of view, so let's have a look at inverters, their properties and characteristics.

Inverters

In the figure at right a CMOS inverter is shown to be composed of an N-channel and a P-channel transistors.

Note that, for each one of the transistors its Substrate terminal is connected to its sourse, so $V_{BS}=0$ and there's no bulk effect.



When $V_{in}=0$ the N-channel transistor is cut-off, while the P-channel is ON. This means that, although in steady state there's no current, V_{out} rises very near V_{DD}, so after any capacitance clinging to the output is charged no current flows.

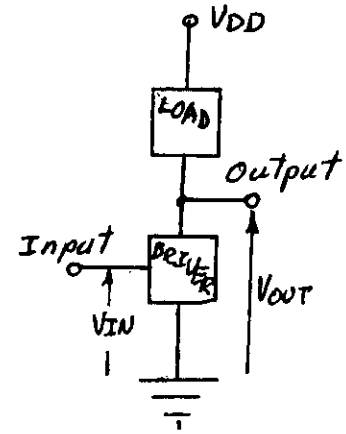
When $V_{IN}=V_{DD}$ the p-channel is cut-off and the n-channel transistor is ON, so $V_{out}\approx 0$.

Important to note: There is no current flow from V_{DD} to GND in steady state.

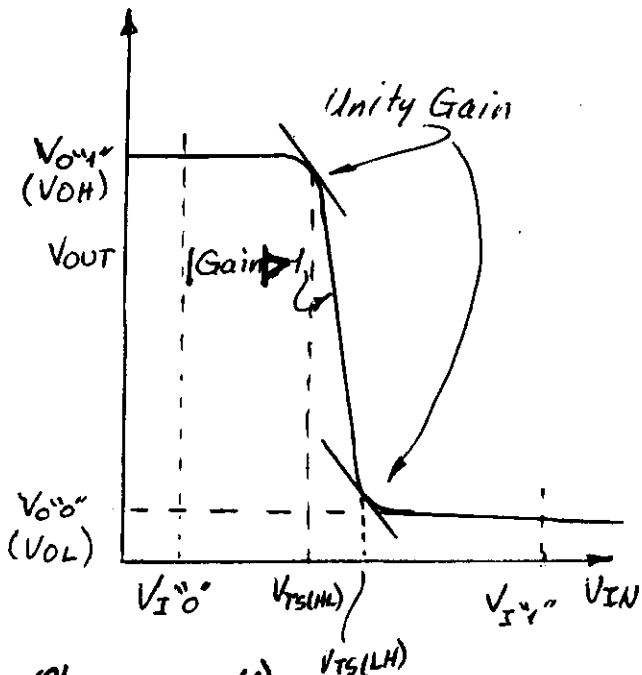
But in the process of going from $V_{out}=0$ to $V_{out}\approx V_{DD}$ both N and P-type transistors will conduct, thus being a direct path for V_{DD} to ground. This current from V_{DD} to GND during transitions must be decreased by designing the inverter appropiately.

Important facts on an inverter

In the inverter represented at right the driver and load devices may be of any type; for example the driver may be an N-type enhancement device and the load an N-type depletion device Although not illustrated, the necessary connections are supposed to exist between the load and the rest of the circuit as to allow for proper functioning.



In the next figure a transfer characteristic (or curve), supposedly corresponding to the said inverter. Unity gain points have been marked, as well as the input and output logic levels. Bear in mind that this curve doesn't have any information regarding time.
Noise Margins (NM) have also been indicated as you can notice. (Please see p.41)



Although for different types of inverters transfer characteristics look more or less alike they have distinct differences.

Current I_{DS} in an MOS transistor depends on factor β, which was defined before to be equal to:

$$\beta = \mu C_o \frac{W}{L}$$

where: μ = carrier mobility
C_o = capacitance/unitarea (Gate)
W = channel width
L = channel length.

Considering that W and L (final values) after fabrication are no longer what we had in the masks at the beginning, this W and L values used to evaluate β must be understood to be the final or effective values. For the sake of simplicity the same symbols will be used further on for these effective values.

Let's define the following ratio:

$$\beta_R = \frac{\beta_D}{\beta_L} = \frac{\beta_A}{\beta_L}$$

where: β_D refers to the driver and β_L to the load devices. Also $\beta_D \neq \beta_A$ (renamed)

Also note in the transfer characteristic that the transition from V_{OH} to V_{OL}, after a certain value of V_{IN} is attained, is accomplished in a more or less steep fashion. Since the slope of this curve is nothing but the voltage gain of the device, care should be exercised to make the change in V_{IN} fast enough as to avoid being in this transition region for a long time; your inverter behaves itself like an amplifier!

A comparison among different inverters

Inverters differ from one another depending of the types of the transistor nearer the GND terminal (called driver) and the transistor sitting on top of it (called load).

Let's consider a very common case where the driver is an N-channel Enhancement Device. So let's compare the inverters formed when changing the load device. regarding their static transfer characteristics.

1 E/E Inverter

This inverter, where E/E stands for enhancement/enhancement is illustrated in the figures that follows and accompanied by its transfer characteristic

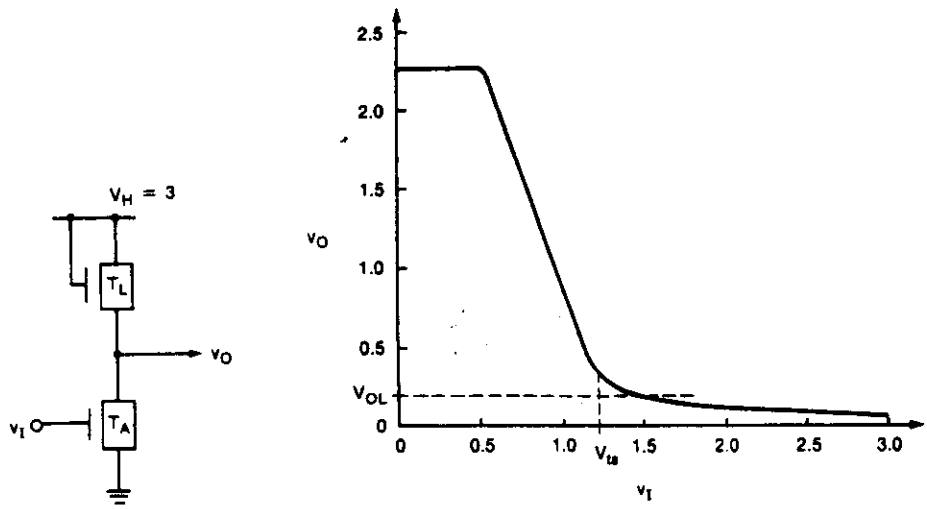


Figure 3. An E/E inverter with a typical transfer characteristic.

Note That V_{OH} doesn't rise to V_H (3V in This example), but only to some 2,25V, This is due to the bulk effect (V_T increase) for transistor T_L due to V_{OUT}. Recall That There's a common bulk or substrate connection for both T_L and T_A! Vout starts to fall out with a discontinuous change of slope at $V_{IN} = V_{TO}(T_A)$. A meaningful V_{TS} can only be defined for the transition from Low to High (see figure).

2 E/D Inverter

For a depletion load we get The transfer characteristic below, Note That when $V_{IN} \leq V_{TO}(T_A)$ Vout rises up to 3V, The power supply value. For $V_{IN} > V_{TO}(T_A)$ T_A draws current from T_L pulling Vout toward ground. A change in slope is produced after the steep change in Vout until Vout levels off.

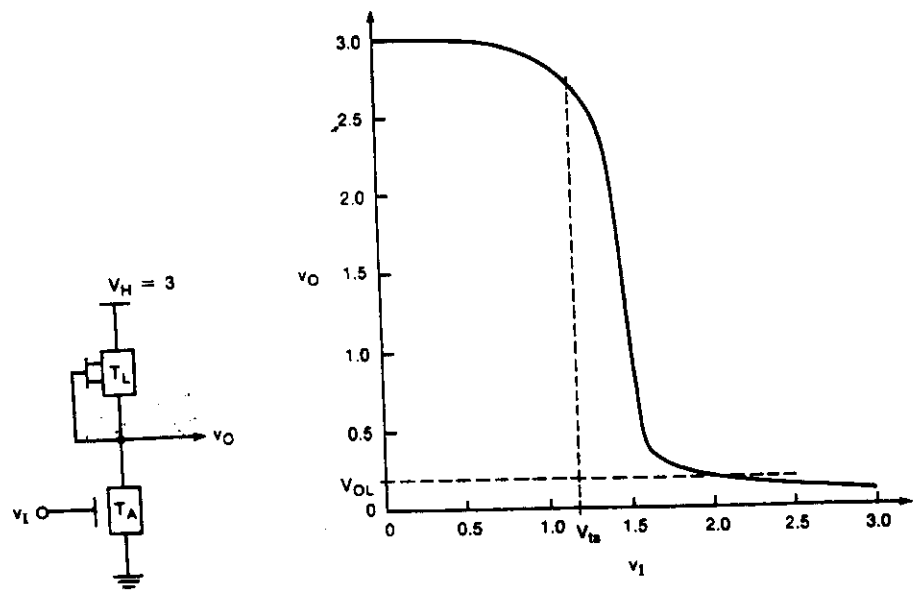


Figure 2. An E/D inverter with a typical transfer characteristic.

CMOS inverter:

In the CMOS inverter shown in the next figure the output voltage rises up to the power supply and falls to ground when H or L respectively. Remember that no static current flows, since either T_A and T_L conduct together only during transitions. This means that, in CMOS, static power (power dissipated when signals have stabilized) is very small - in fact is negligible - but dynamic power, due to current through both T_L and T_A from V_{DD} to GND, and to charging and discharging of circuit capacitances, is important and depends on switching frequency.

$$\beta_r \triangleq \frac{\beta(T_A)}{\beta(T_L)} = \frac{\beta_A}{\beta_L}$$

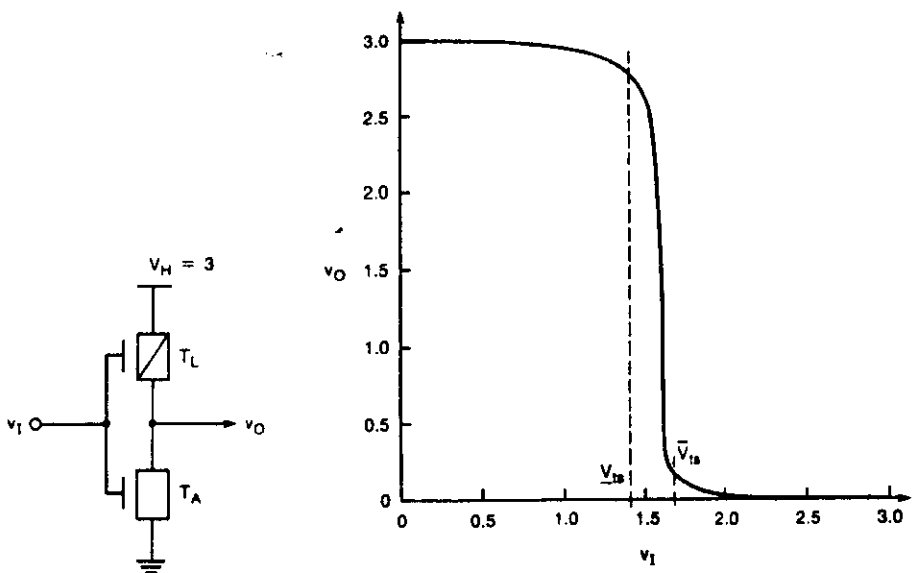


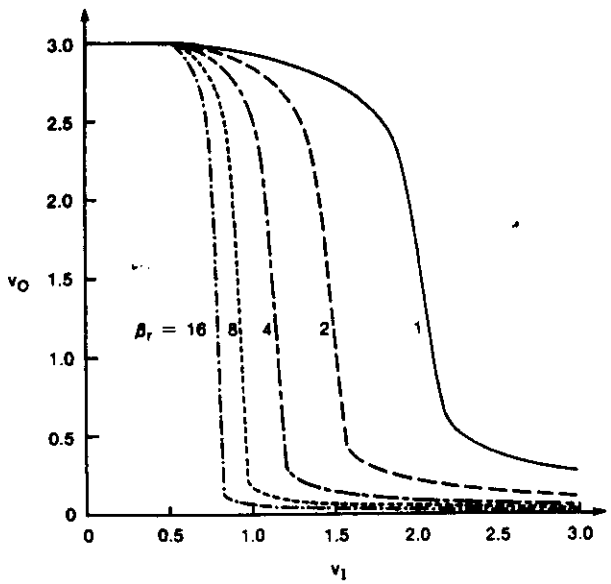
Figure 4. A CMOS inverter with a typical transfer characteristic.

Influence of β_R

β_R was previously defined to be equal to:

$$\beta_R = \frac{\beta_A}{\beta_L}$$

Below the influence of β_R on the transfer characteristic of an E/D inverter and a CMOS inverter are illustrated. Increasing β_R increases the gate capacitance of T_L, thus increasing the capacitive load. For CMOS and any other type of inverter this reflects at the input as an increased capacitance, loading the previous stage.



jure 5. A family of TC plots of E/D inverters of different β ratios.

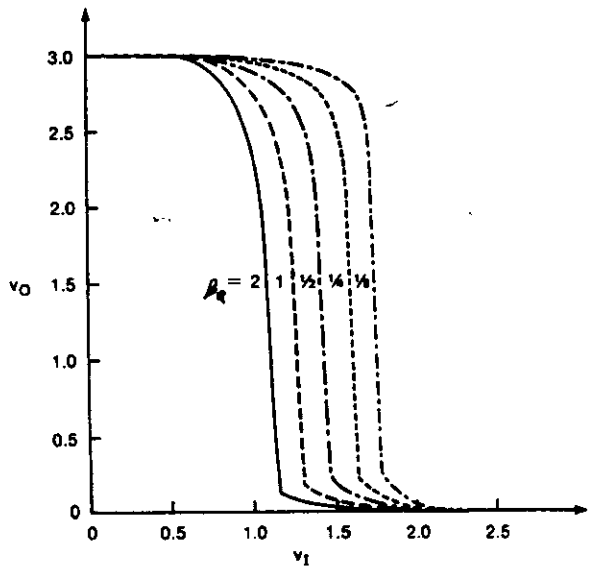


Figure 12. The TC plots of a family of CMOS inverters.

$$\beta_r \triangleq \frac{\beta(T_A)}{\beta(T_L)} = \frac{\beta_A}{\beta_L}$$

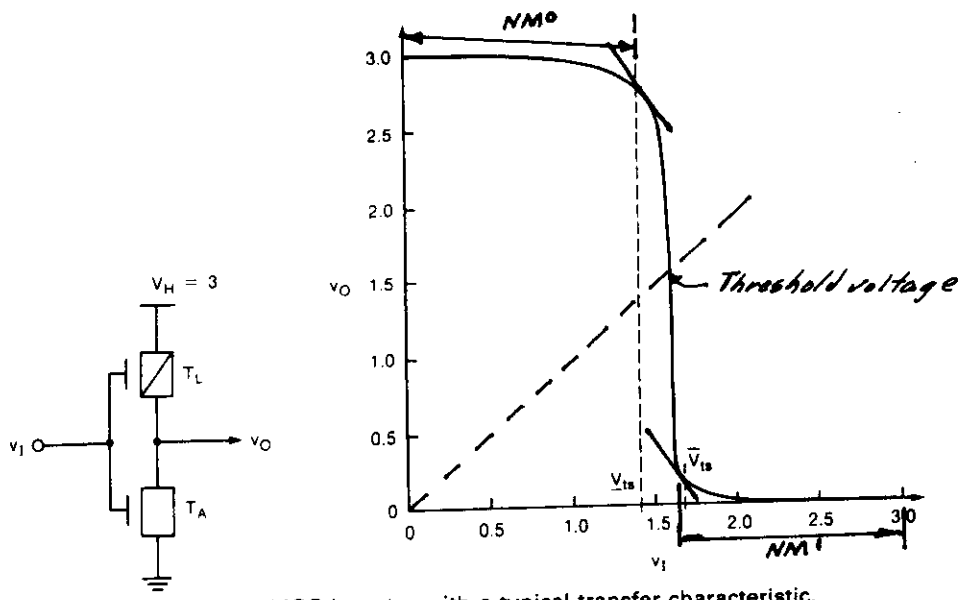


Figure 4. A CMOS inverter with a typical transfer characteristic.

Noise margins NM^0 and NM^1 are the allowable changes in V_I for obtaining a certain output V_O. NM^0 Thus represents the possible variation of V_I when it has an applied zero (0), That is to say, to maintain a "1" in V_O. CMOS ≡ Good Noise Margins

The threshold voltage V_t (see figure above) can be calculated recalling That in That region both transistors are saturated, so

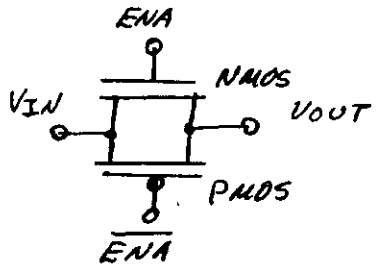
$$I_{DSN} = \frac{\beta_N}{2}(V_I - V_{TN})^2 \quad \text{and} \quad I_{DSP} = \frac{\beta_P}{2}(V_H - V_I - V_{TP})^2$$

$$\Rightarrow \sqrt{\frac{\beta_N}{\beta_P}} = \frac{V_H - V_I - V_{TP}}{V_I - V_{TN}} \quad \text{and} \quad V_I = \frac{V_H + V_{TP} + V_{TN}\sqrt{\frac{\beta_N}{\beta_P}}}{1 + \sqrt{\frac{\beta_N}{\beta_P}}}$$

Let's suppose we would like That the transition voltage were in between V_H and GND, just at $\frac{V_H}{2}$ then, if $\beta_N = \beta_P$ and $V_{TN} = V_{TP}$ we get the P device should be two times as large as The N device. Recall That $\beta = \frac{\mu\varepsilon}{t_{ox}} \cdot \frac{W}{L}$ and $\mu_N \approx 2\mu_P$

Transmission Gates

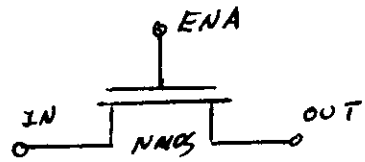
A very useful circuit element is the so-called transmission gate, which is formed by a PMOS connected with an NMOS as shown.



Note that ENA="1" turns ON both transistors, while ENA="0" turns them OFF.

The result is passing to VOUT the value set at V_{IN}.

The transmission gate is bidirectional, as can be easily seen. The configuration above is typical for CMOS. For NMOS (and correspondingly PMOS) a single transistor might be used:



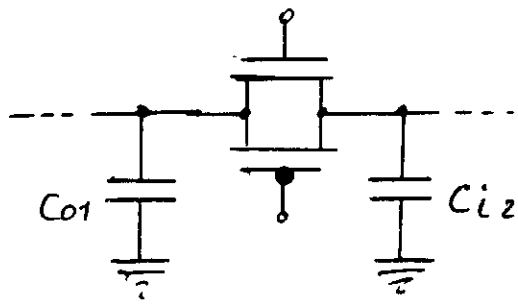
The drawback of this approach, which has been used in the so-called quasi-static logic, is that for V_{IN}="1" the voltage at VOUT is V_{IN} minus one Threshold voltage drop (bulk effect).

If you connect several transmission gates in an adequate fashion you can make logic operations. This is called "steering" logic. Due to bulk effect, you cannot connect several one-transistor transmission gates in series. Each one, introducing a V_T drop in its output might make that the information at V_{IN} be lost. A "1" at V_{IN} turns out, roughly, in: $V_{OUT} \simeq V_{IN} - nV_T$, where n is the number of series one-transistor gates.

Charge sharing:

Besides all mentioned about transmission gates There's an additional problem related to charge redistribution when the gate is on.

For instance, let's admit we have a capacitance C_{o1}, the capacitance connected to the output of an inverter including parasitic elements. Suppose this node, and its associated capacitance are to be connected to the input of a circuit, with input capacitance C_{i2}, via a transmission gate as shown.



What final voltage will there be in C_{i2} if C_{o1} is loaded at V_{o1}?

$Q = V_{o1} C_{o1}$

When the gate is on:

$Q = V_{final} (C_{o1} + C_{i2})$

and $V_{final} = \frac{V_{o1} C_{o1}}{C_{o1} + C_{i2}}$

So you may turn out with a final voltage much less than V_{o1}!

Applications I: A start point

An Introduction

Some classification must be made regarding "basic" circuitry. A reasonable one may be:

- Static circuitry
- Quasi-static
- Dynamic.

And now the question arises as to, what is "basic" standing for?
First of all, logic gates, flip-flops, etc.

Static circuitry and quasi-static circuitry look similar to each other as we'll see in a short while.

Dynamic circuitry is a little more "itchy" regarding operating frequency and, most of all, it's minimum value. Refreshing is commonly used to restore information that, otherwise, would be lost due to leakage currents.

Static circuitry

Basic, in this context, means elementary gates:

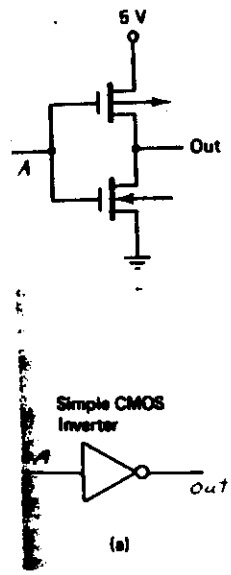
- inverter
- and
- or
- nand
- nor
- and-or-invert

Many others might be also considered basic, but they're formed by combinations of these few gates listed at the left.
Gate implementation is not a difficult matter in CMOS, as we'll see in the following pages.

Inverter :

Inverters differ from each other in their specific capabilities to accomplish a certain characteristic. For instance, propagation delays, Fan-Out and so on might be different among different inverters. This may seem an obvious remark, but what we try to point out is the fact that they might be quite different.

The basic inverter is shown in the figure at right. Please note that the P-type transistor "sits" on top of the n-type one, and that its substrate is connected to its source (and to the positive power supply rail).



Main characteristics are :

- Logic "1" (or High) output level
- Logic "0" (or Low) output level
- Propagation delay time and relate L-H and H-L transition times
- Static current drawn from the power supply
- Maximum operating frequency.

For an inverter like the one shown in the figure above we have :

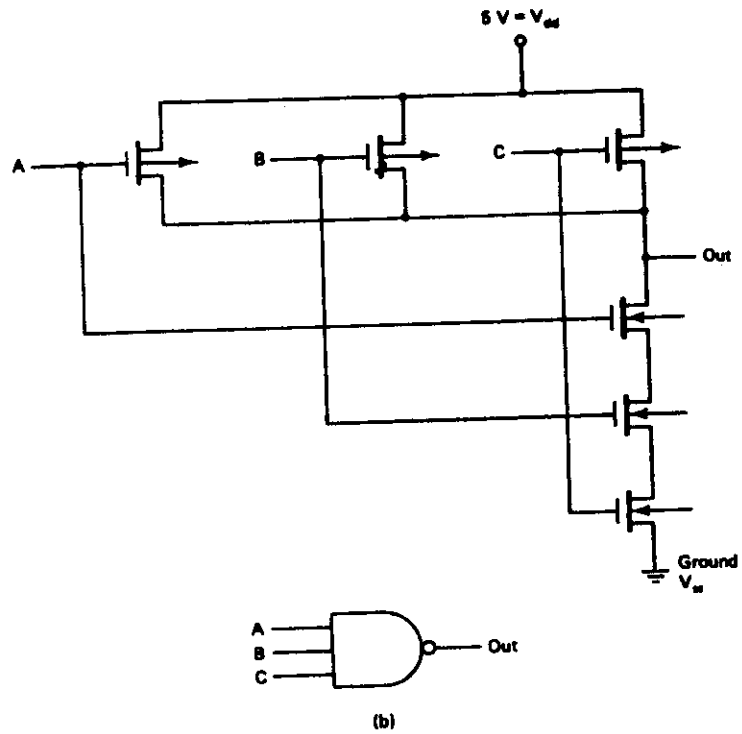
$Out = \bar{A}$

A	Out
0	1
1	0

NAND Gates

In the figure at right a 3-input NAND gate is shown with the following truth table

A	B	C	Out
0	0	0	1
0	0	1	1
⋮	⋮	⋮	⋮
1	1	1	0



(b)

which can be represented by

$$Out = \overline{A \cdot B \cdot C}$$

Please note that, like in any CMOS gate, there's no static current from V_{DD} to V_{SS}. Current only flows during transitions from H to L or viceversa. Current also flows, like in any gate, into and out of the Out terminal, in order to discharge/charge capacitive loads.

A 5-input NAND would have a similar, although a little bit more complicate, structure.

AND gates can be easily implemented recalling that

$$A \cdot B \cdot C = \overline{\overline{A \cdot B \cdot C}}$$

Actually, Inverters, NAND Gates and NOR gates are the "bricks" that you use for implementing more complex functions.

NOR Gates:

Referring to the inserted figure we can note distinct differences between a NOR and a NAND gate. First, note that all N-type transistors are in parallel and not in series. Secondly, P-type transistors are connected in series and not in parallel.

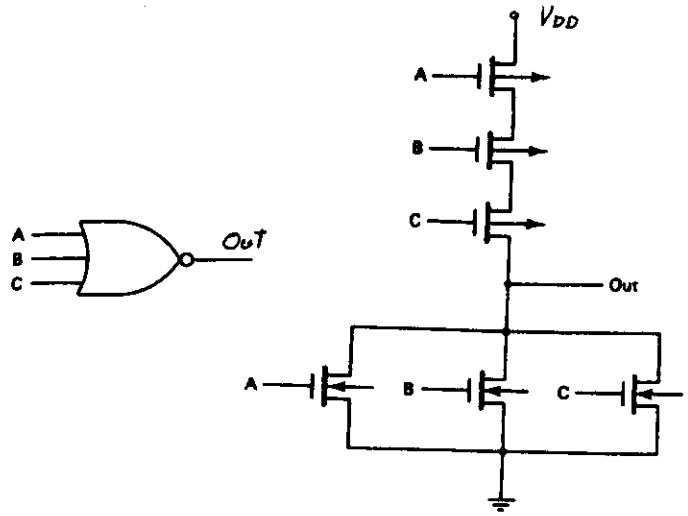


Figure 2.26 CMOS NOR gates.

So, we get for the truth table:

A	B	C	Out
0	0	0	1
0	0	1	0
⋮	⋮	⋮	⋮
1	0	0	0
⋮	⋮	⋮	⋮
1	1	1	0

$$\overline{Out = A+B+C}$$

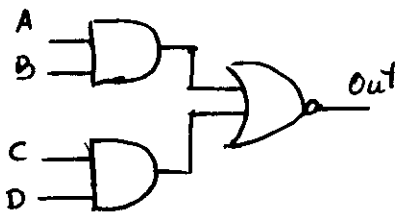
There's a quite popular type of gate which is called And-Or-Invert, that we'll discuss next, but before that: you can achieve an OR gate recalling that:

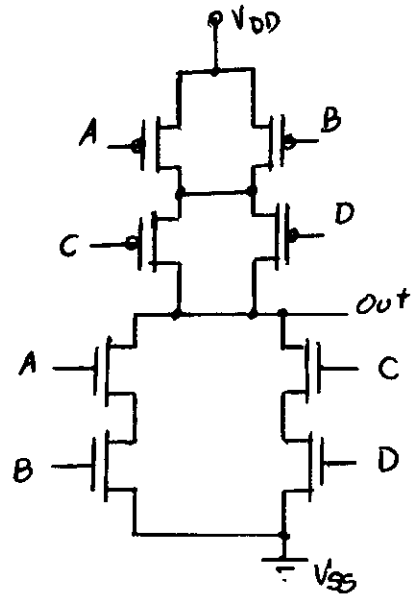
$$A+B = \overline{\overline{A+B}}.$$

That is, using an inverter and a NOR gate.
Generally speaking, ANDs and ORs can be implemented with NANDs and NORs using an extra inverter.

And-Or-Invert (AOI) and OAI : Two-level logic blocks

The And-Or-Invert is a useful combination of ANDs and OR. A 4-input AOI and its corresponding truth table and CMOS implementation are shown below.





A	B	C	D	Out
1	1	0	0	0
0	0	1	1	0
1	1	1	1	0
⋮	⋮	⋮	⋮	1
⋮	⋮	⋮	⋮	⋮

$$Out = \overline{A \cdot B + C \cdot D}$$

Note That the only input combinations making Out = "1" are those where [A B], [C D] are "1".
Please, note also That the CMOS implementation produces The desired response in Vout, which can be easily checked-out in the figure.

AOIs are talked-about as two-level logic functional blocks. An example of another two-level logic functional block is The OAI or Or-And-Invert which is shown in the following figure.

Please, check out that $y = \overline{(X1+X2)\cdot(X3+X4)}$

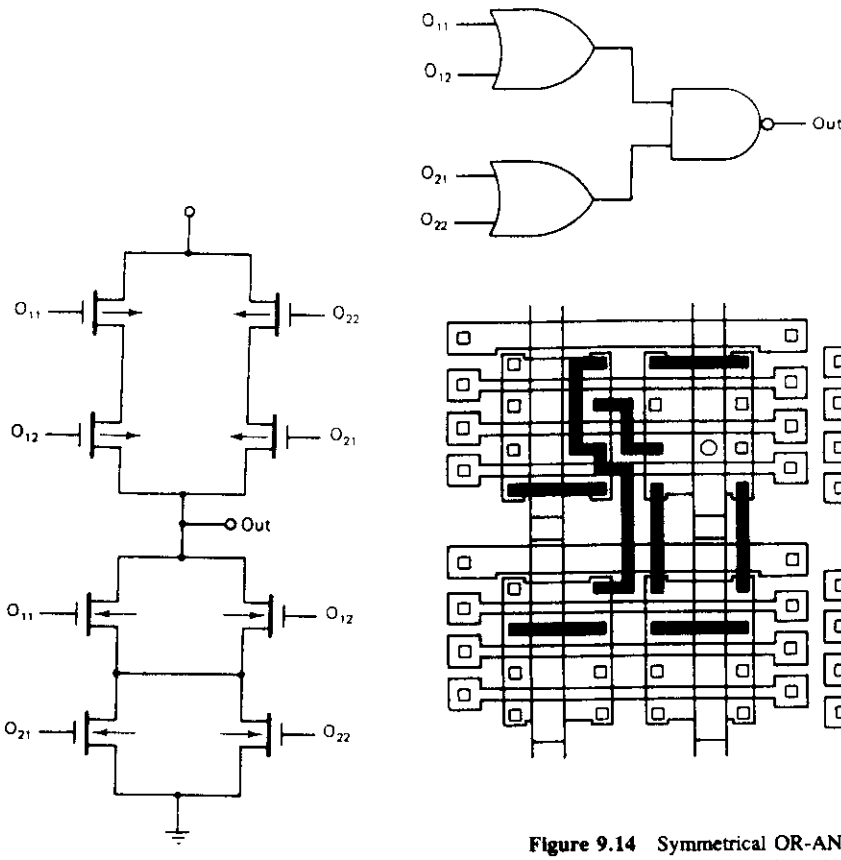
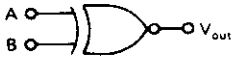


Figure 9.14 Symmetrical OR-AND-INVERT OAI: (a) FET structure (b) layout. (Courtesy of International Microcircuits, Inc.)

Some tasks AOI and OAI can do are MUXes (Multiplexors) XORs, XNORs etc. Of course, some slight change might be necessary, like shown in the figures that follow

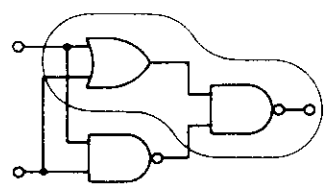


(c)

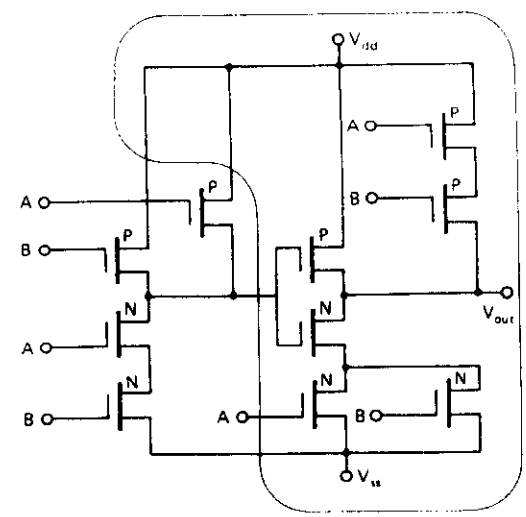
A	B	V_{out}
0	0	1
0	1	0
1	0	0
1	1	1

(d)

Figure 9.19 EX-NOR made from degenerate OAI (degenerate OAI is encircled): (a) logic circuit; (b) circuit diagram; (c) symbol; (d) truth table. (Courtesy of Interdesign; modifcations to FET symbols by the author.)



(a)



(b)

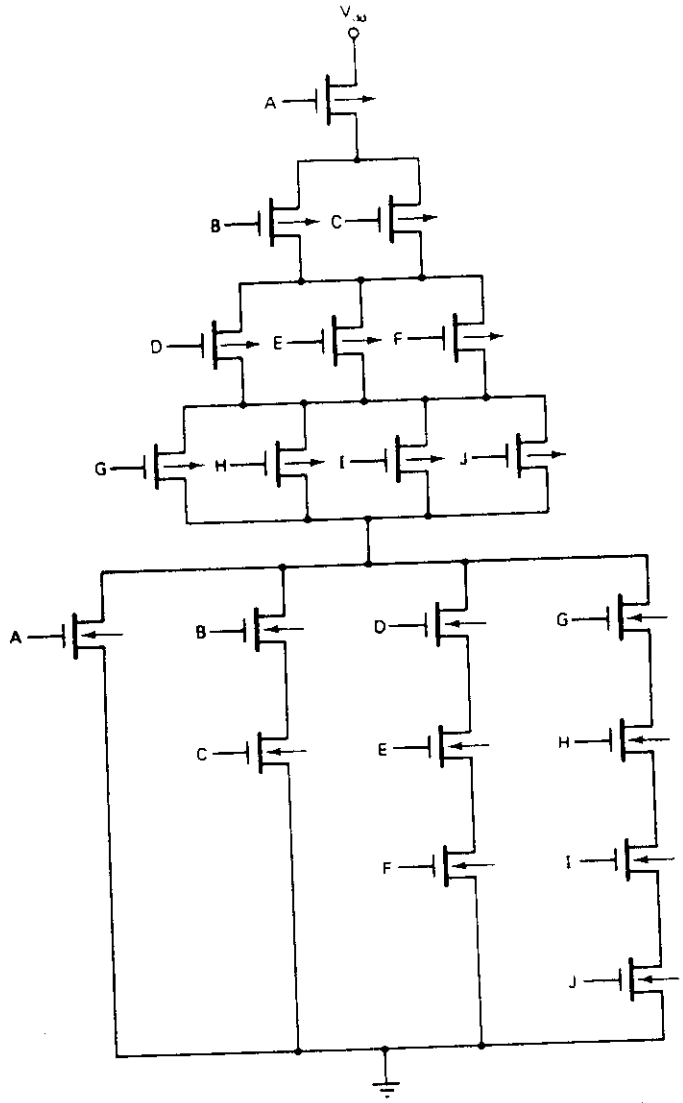


Figure 9.16 (a) FET diagram of asymmetrical AOI in Figure 9.15; (b) corresponding FET structure.

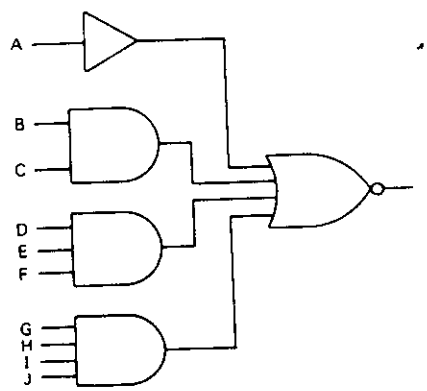


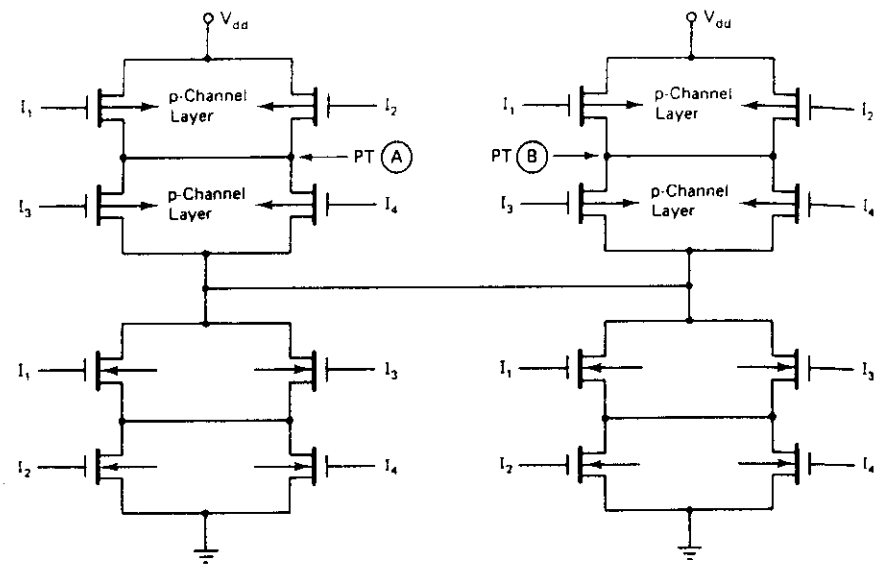
Figure 9.15 Asymmetrical AOI

9.9.4 Importance of Tying Node Capacitance to Power Supply Rails

Figure 9.16 has been drawn with the PFETs in the form of a "tree" in order to better exhibit the layering effect of the PFETs. In reality, the structure would be laid out with the PFETs in an inverted tree. The object is to tie as much source/drain capacitance as possible to the power supply rails. Taking the output from where it is shown would slow the signal considerably because of the high driving-point capacitance. When laying out asymmetrical OAIs, the NFET layer containing the most NFETs would have its common source connected to V_{ss}.

Paralleling of structures

To decrease the fan-out seen by a given device structures can be paralleled as is shown in the figures that follow.



It is not necessary to connect PTA to PTB but it is OK to do so ONLY IF I_1 and I_2 are on the same "p-channel layer" (in this case the topmost one) in BOTH of the paralleled AOIs

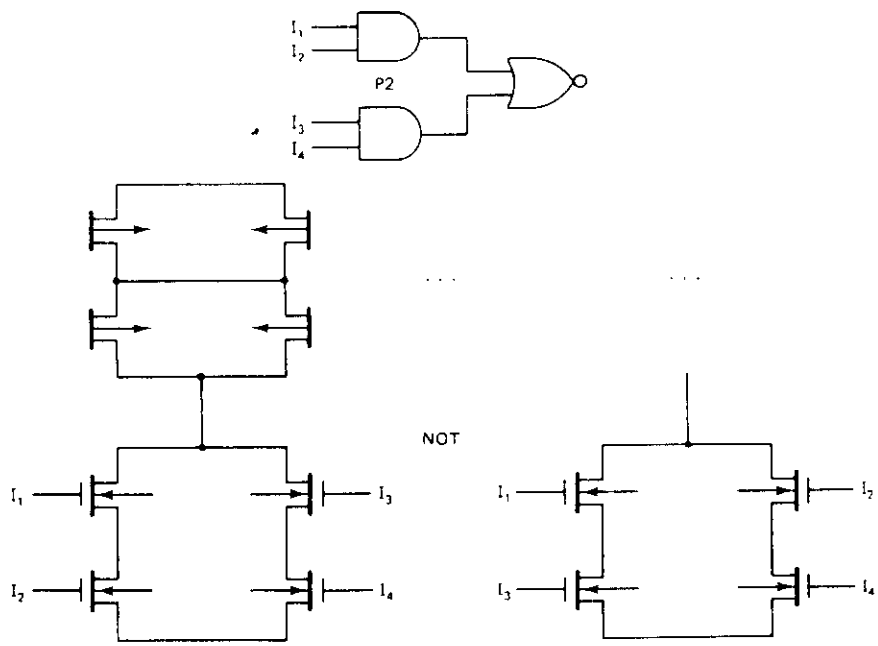


Figure 9.46 Paralleling AOIs.

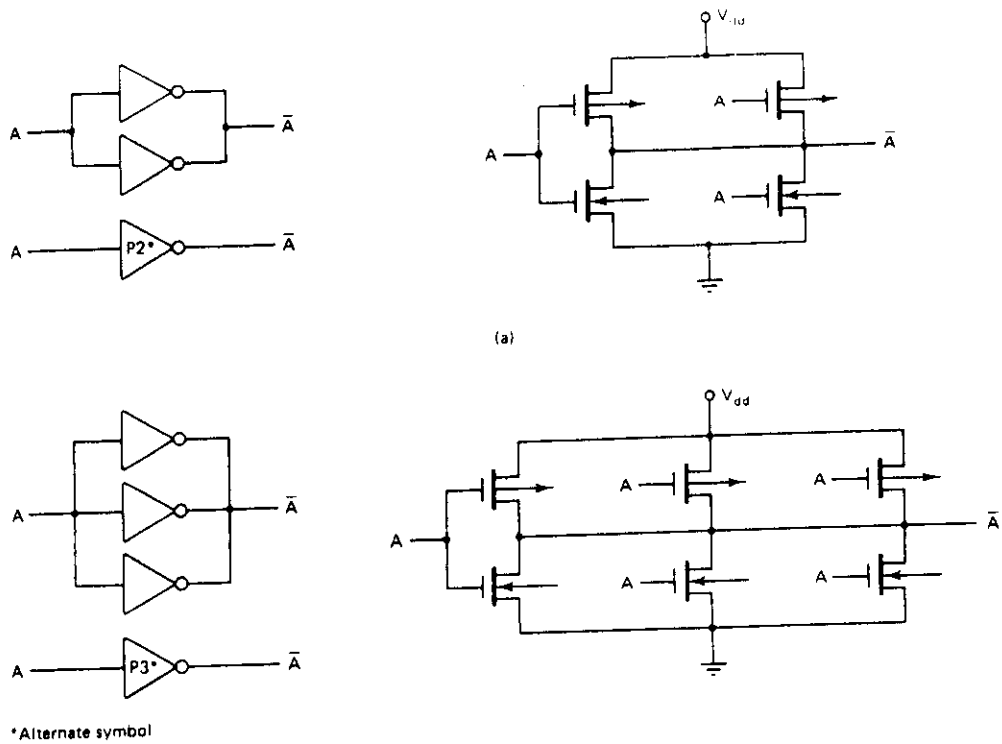


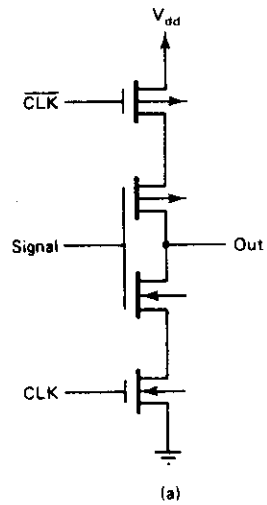
Figure 9.44 Paralleling transistors for greater drive: (a) double paralleled (designated P2); (b) triple paralleled (designated P3).

Floatable Drivers (Clocked Inverters)

A floatable driver, FD, is an interesting structure shown in the figure at right.

Note that the signal goes into a gate when CLK = "1" the structure behaves itself like an inverter.

When CLK = "0" the output floats. This allows several outputs to be wired-or.



Output and buffered structures

Buffered structures are used when a highly capacitive load is to be driven. For instance, this may happen when a clock or other signal line must be routed around the chip to many nodes.

To drive large capacitances what you use is a chain of inverters appropiately dimensioned to charge or discharge the said capacitance in the smaller possible time

Recalling that, to a first approximation, the transit time (for carriers to go from S to D) in an NOS transistor is given by:

$$\tau_t = \frac{L^2}{\mu(V_{GS}-V_T)}.$$

It can be shown that the total delay in charging a capacitor C_L is:

$$\tau_D \simeq 2.\tau_t \cdot \frac{C_L}{C_g},$$

where C_g is the total gate capacitance of the transistor that contributes to current (no parasitics considered).

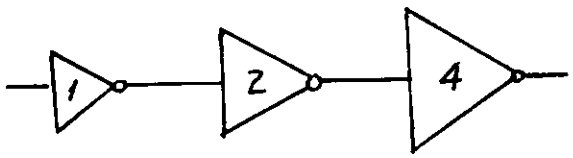
Not only will time grow if C_L is large, but also power consumption will increase due to both N and P type transistors being ON at the same time. Remember that rise and fall times become "slower".

The approach is to use a chain of inverters with increasing sizes (W/L). If the size ratio from one inverter to the previous one is $\underline{e}$ (remember $e = 2,71828\ldots$, is the base of natural logarithms) it has been shown that total delay is then given by:

$$\tau_D \simeq 2 \cdot e \cdot \tau_t \cdot \ln \frac{C_L}{C_g}$$

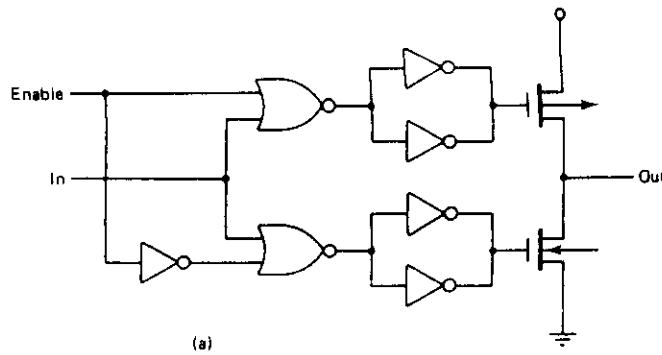
with $\ln \frac{C_L}{C_g}$ representing the number of stages N.

In practice what you use as a proportionality factor is not $\underline{e}$ but a power of 2. So that, if your basic inverter is represented by a size factor 1 then your inverter chain might look something like this.



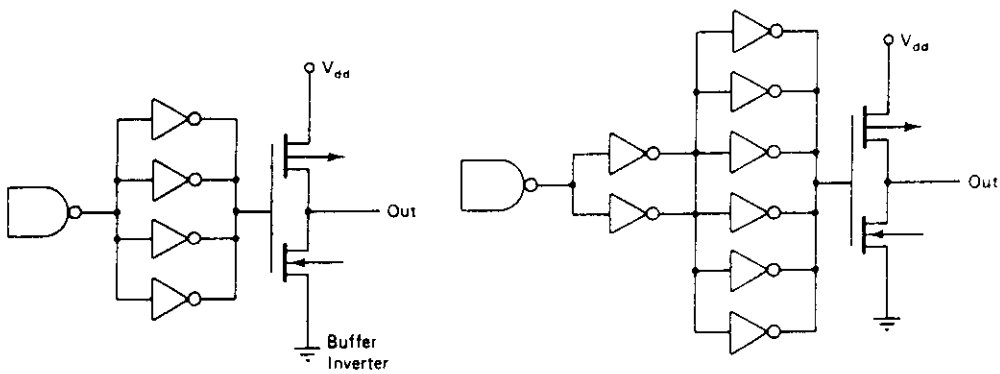
Output buffers are designed to drive large capacitances like those to be encountered when the output pin is connected to a bus line. At times they are called super buffers (just an elegant name and impressive also), which also include tri-state capabilities (high impedance or high Z state) and may be inverting or not inverting. To drive a superbuffer it's mandatory to use a good drivingstage, like the one just mentioned above.

Enable	In	Out
L	H	L
L	L	H
H	X	HI-Z



(a)

Figure 9.43 (a) Tristate output buffer configuration; (b) layout; (c) layouts of various other buffer combinations. Student exercise: one of the NOR gates should be a NAND gate. Which one is it?



(a) (b)

Figure 9.45 Driving output buffers: (a) inverting; (b) noninverting. (Note Third-generation gate arrays have drivers for the buffers built into the macros. Check with vendor.) Each buffer inverter pair is 12 unit loads. Drive each buffer FET with at least two paralleled inverters. The buffer shown in part (b) is 2 to 5 ns slower than the buffer shown in part (a).

Applications II

More complex blocks! "Flip-Flops and relatives"

Flip-Flops are made out of gates, of the simpler gates we already had a look at.
The gate array vendor supplies Flip-Flops, complex gates and so on for you to use without having to worry about their practical implementation. He'll supply delays and other time parameters so: What else should you desire?

In the figures that follow you'll find the implementation of some popular, and others not popular, Flip-Flops and counters.

The T-type (Toggle) Flip-Flop shown might look a little awkward, but will function properly as a T-type despite the small time delays to be expected from the gates

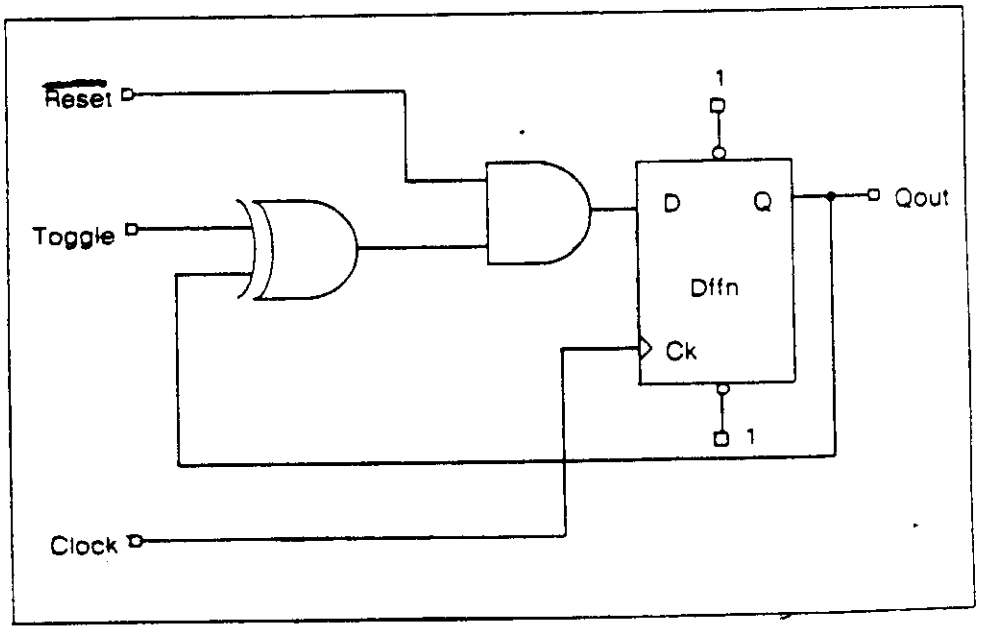


Figure 8.26: T-type Flip-flop

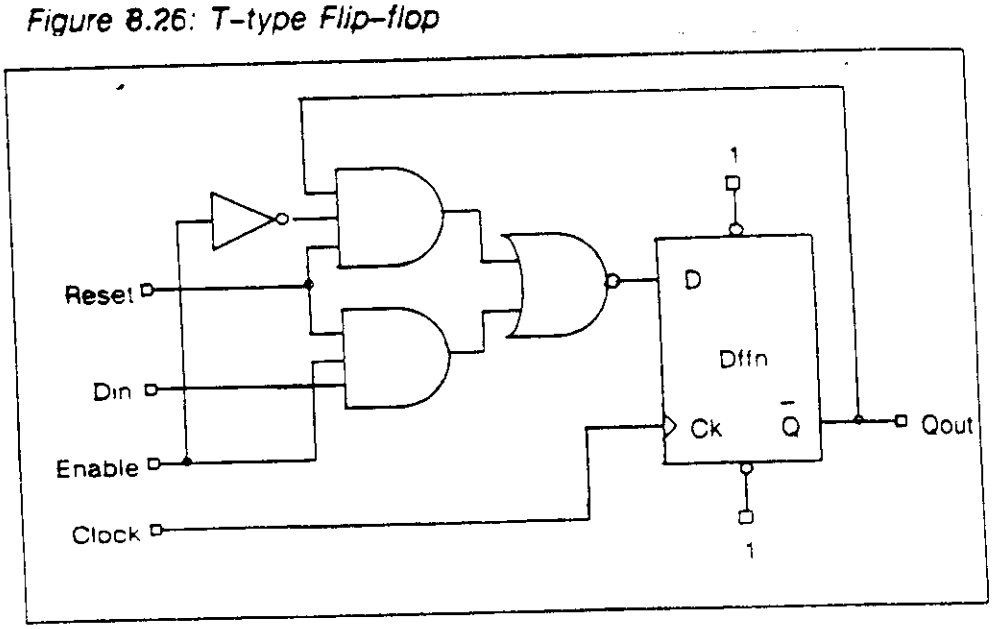


Figure 8.25: Optimised E-type Flip-flop

8.4.4 E-type Flip-flop

The Enable or E-type flip-flop (effn), also known as a multiplexed flip-flop, is made from a D-type and a multiplexer. See Figure 8.23 for the principle of the circuit, and Figure 8.25 for an optimised version. The multiplexer in Figure 8.23 uses the ANDNOR part described in Section 8.1.2. The Enable input chooses whether to gate a new data input into the latch, or to hold the previous data.

Note that the E-type flip-flop described here is not the same as a transparent latch, where the input is strobed directly to the output.

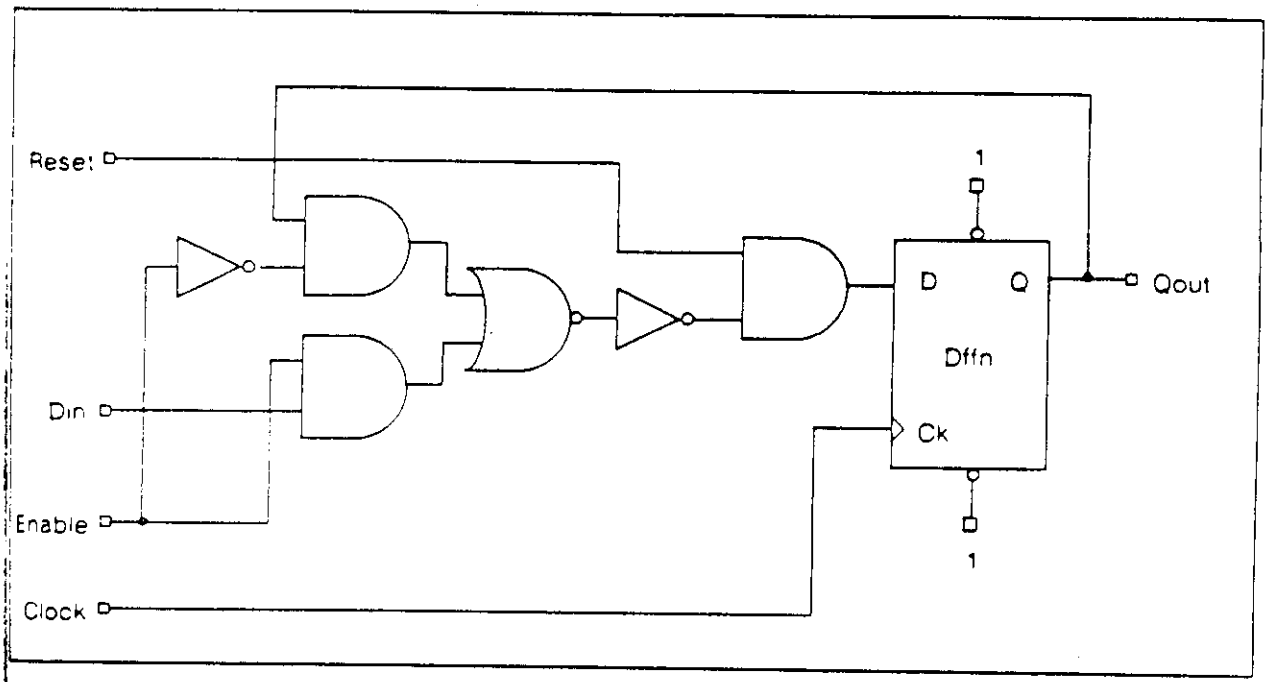


Figure 8.23: E-type Flip-flop

The operation of the circuit is as follows:

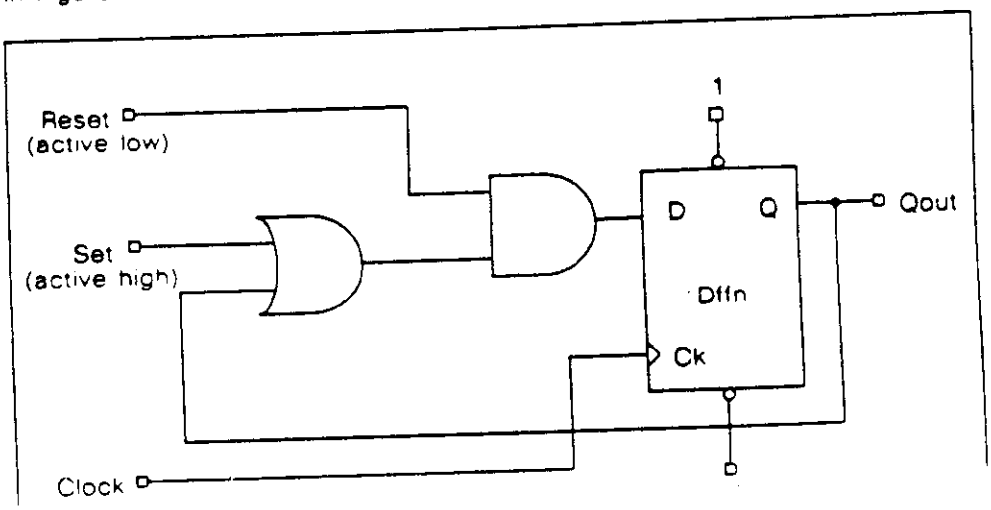
Enable	Reset	Qout
0	0	0
1	0	0
0	1	Qout (previous clock)
1	1	Din (previous clock)

Reset is active low: when it is low the next output is low.

Enable is active high: when it is low, the output data state is maintained from one clock cycle to the next. When Enable is high, the data input passes to the data output after one clock cycle.

If we recall that the simple RS Flip-Flop has an ambiguous and hard to define state, we'll welcome an implementation with no such a difficulty. Please, find it immediately below.

The asynchronous RS flip-flop is not suited for inclusion into an integrated circuit for a number of reasons, in particular its state is unknown when R and S are high. However, its form may be adapted to the synchronous version shown in Figure 8.31.



This form of the flip-flop is synchronous, and there is no ambiguity over its state. Reset takes priority over Set.

Counters

The next page illustrates several counters used in many applications in IC design
Although they can not be classified as counters, shift registers (SR) are also important and useful complex cells supplied usually in vendors' libraries

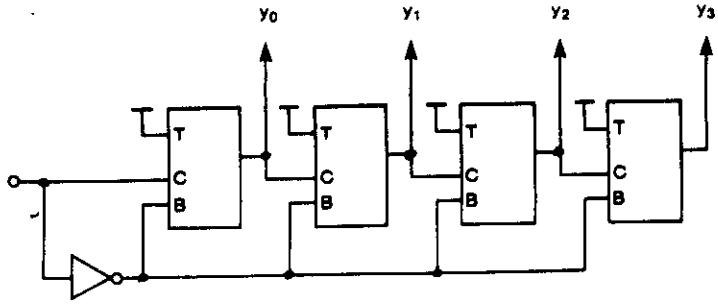


Figure 26(a). A 4-bit ripple counter.

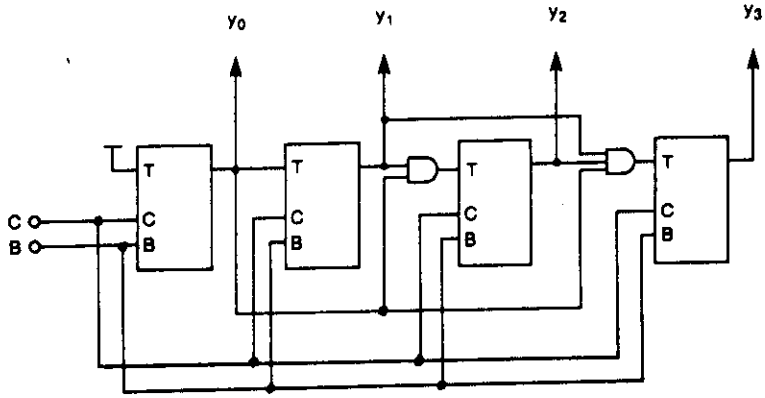


Figure 26(b). A 4-bit synchronous binary counter.

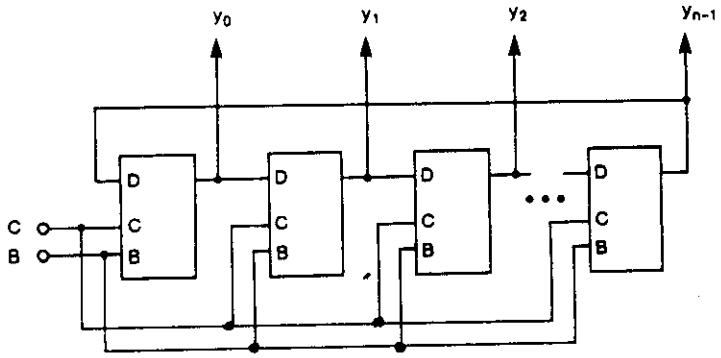


Figure 27(a). An *n*-stage ring counter.

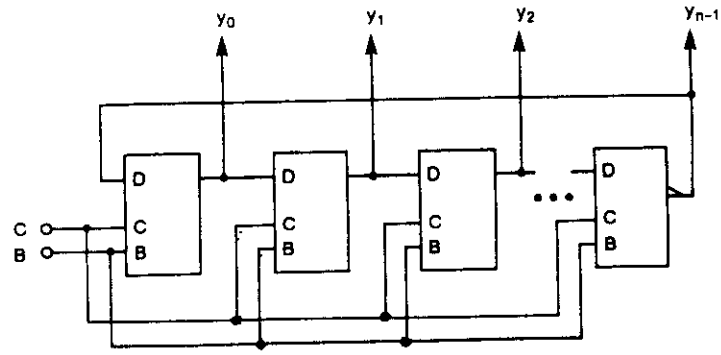


Figure 27(b). An *n*-stage Johnson counter.

In the 4-bit ripple counter a "1" is added to the binary count every time the input gets a "1". Resets to $y_3y_2y_1y_0 = 0000$ after $2^4 = 16$ "1s". Also called modulo n counter, n being number of FFs

The 4-bit synchronous binary counter advances the count in synchronism with the clock. Notice there's a single clock signal C.

Next counter is an n-stage ring counter. After initializing to (y_0 = "1")

1 00 ... 0

↑ MSB ↑ LSB

it propagates the "1" down the chain. Notice feedback from y_{n-1} to the input. By feeding the complement of the output at the last stage to the input of the first one the Johnson counter or twisted-ring counter is created. Counts up to 2n, where n is the number of stages. The output changes only one bit at a time.

Transmission Gates II

Again transmission gates ?
Well, as a matter of fact yes! A little bit more again. Since we already introduced transmission gates it won't do any harm to have a look at some symbols for representing it.

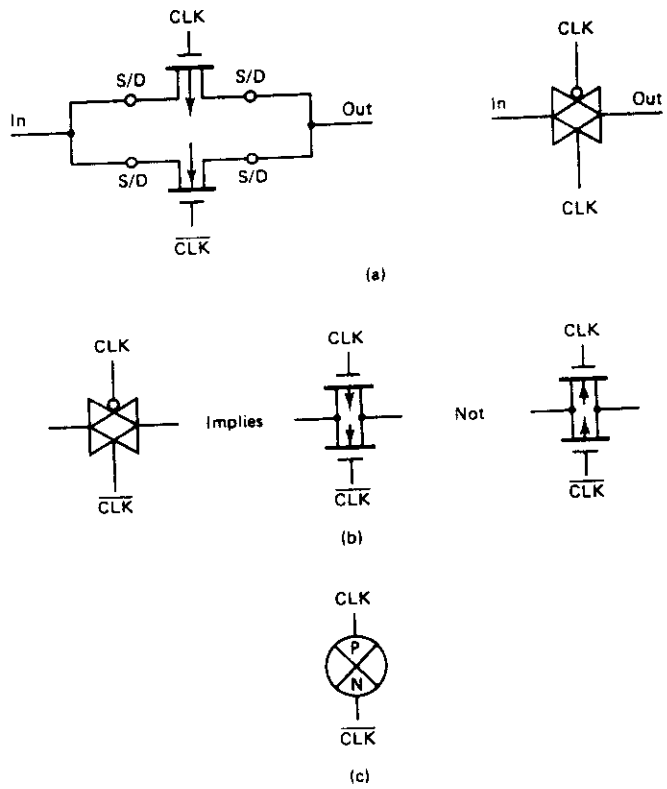
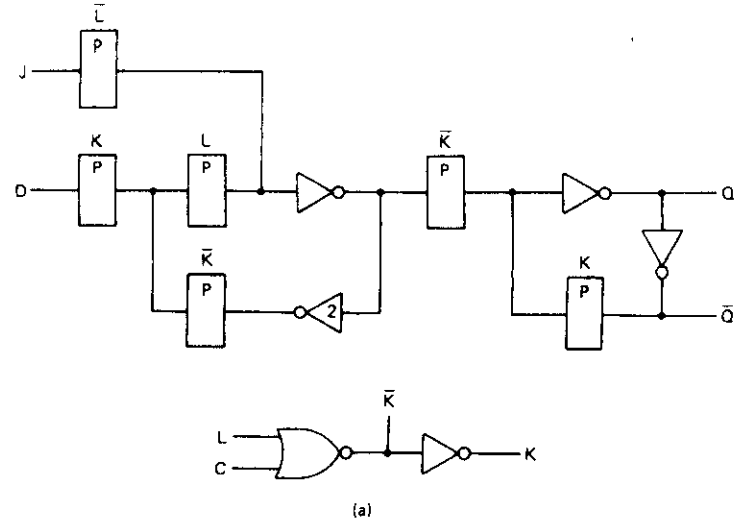
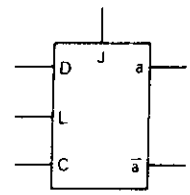


Figure 9.25 (a) Transmission gate (TG); (b) standard symbols; (c) alternative symbol.

Applications of transmission gates range from switches in latches and flips flops to implementation of logic with the so-called steering logic
In the figures that follow several applications are shown which illustrate this statement.



(a)

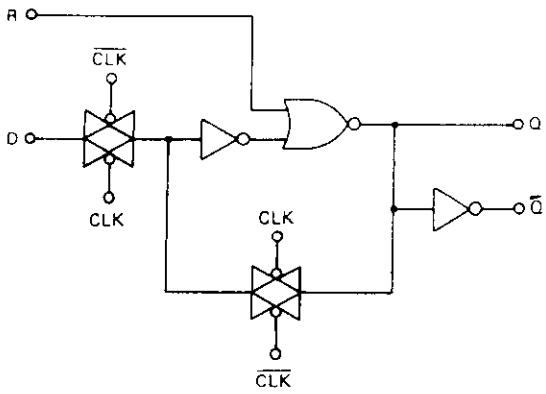


(b)

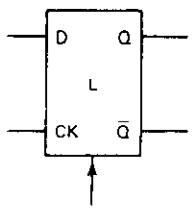
D	J	C	L	QN
X	0	X	1	0
X	1	X	1	1
0	X	↑	0	0
1	X	↑	0	1
X	X	↓	0	QN − 1

(c)

Figure 9.34 D flip-flop with jam data input and low active reset: (a) circuit schematic: (b) symbol; (c) truth table. (Courtesy of International Microcircuits, Inc.)



(a)



(b)

A	CK	D	Q	$\overline{Q}$
1	X	X	0	1
0	1	1	1	0
0	1	0	0	1
0	1	X	No Change	
0	0	X	No Change	
0	↑	1	1	0
0	↑	0	0	1

(c)

Figure 9.32 Clocked latch with reset: (Courtesy of Interdesign.)

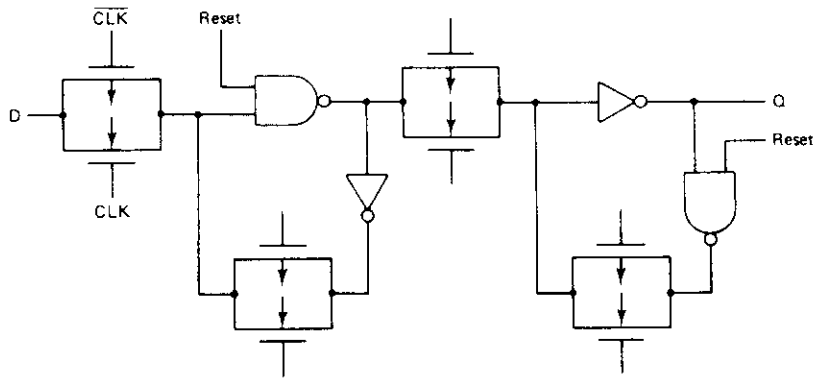
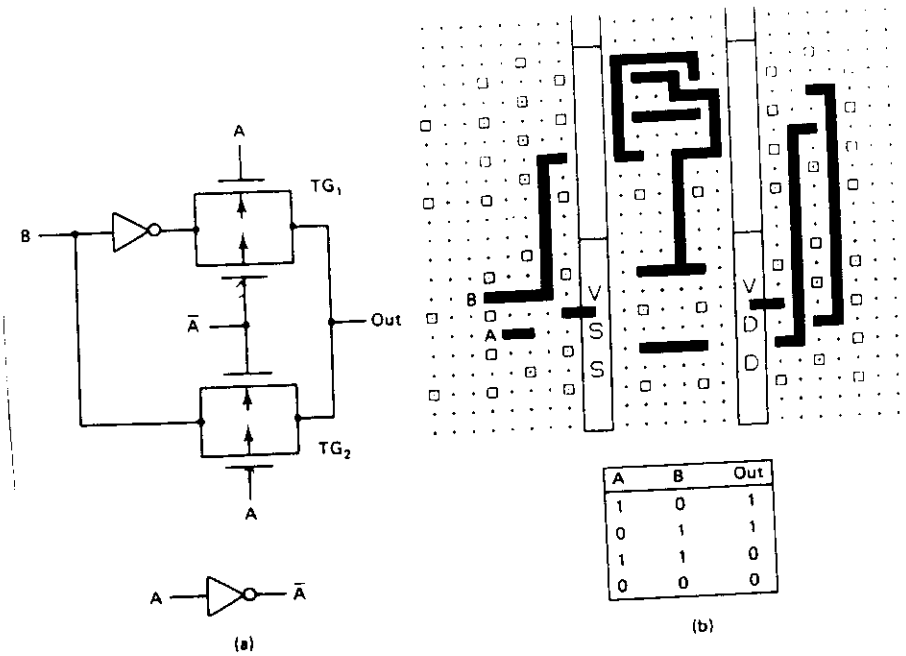


Figure 9.33 D flip-flop, which holds reset state after reset goes high regardless of clock polarities. Student exercises: (1) label remaining TG CLK and $\overline{CLK}$ signals; (2) prove above statement; (3) Does Q change state on high or low clock transition?; (4) How could you make Q change state on the clock phase, which is opposite to that in number 3?



A	B	Out
1	0	1
0	1	1
1	1	0
0	0	0

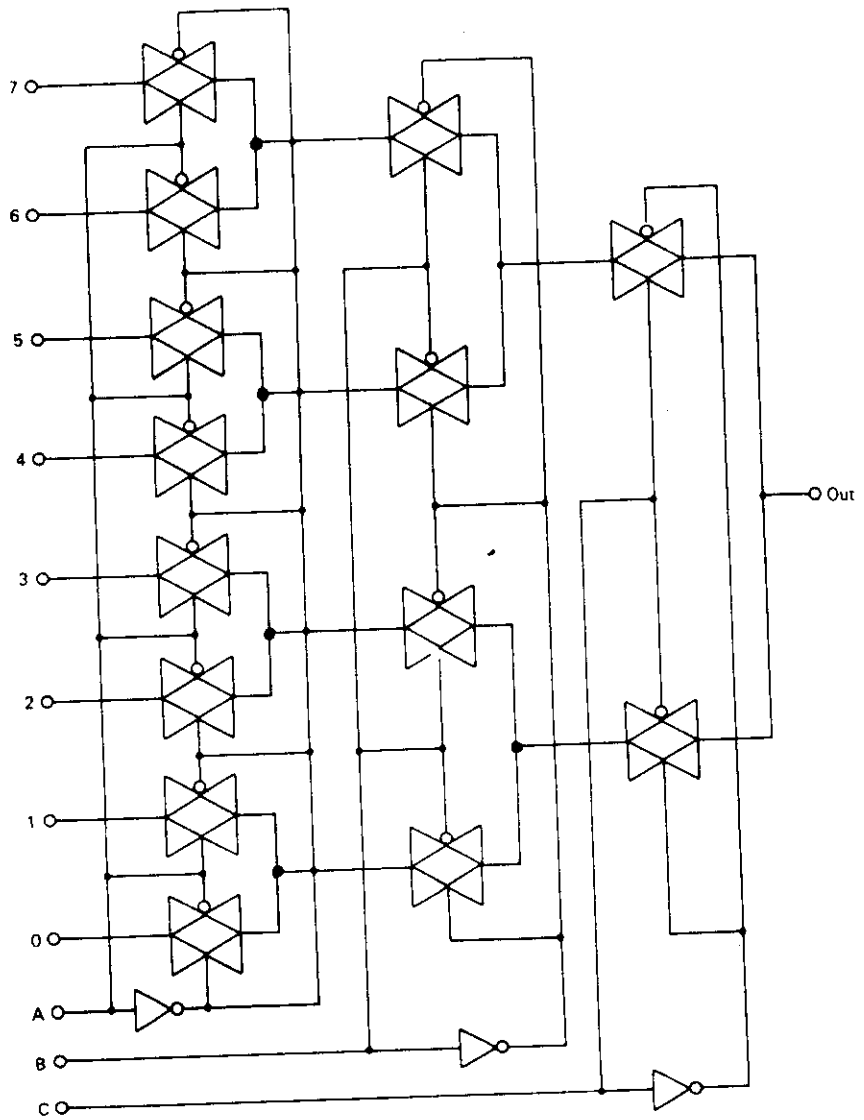


Figure 9.35 Decoder using transmission gates: 8-to-1 decode tree. (Courtesy of Interdesign.)

In the figure at left an XOR gate, implemented with transmission gates is illustrated.

Please, verify that

$Out = A \oplus B$

The next figure shows an 8-to-1 decode tree which uses 14 transmission gates and 3 inverters.

Just verify that Out accomplishes this function

Domino logic

Domino logic is a simplified version of dynamic logic, which is shown at right. Inputs and outputs are latched in registers and the random logic is grouped in blocks.

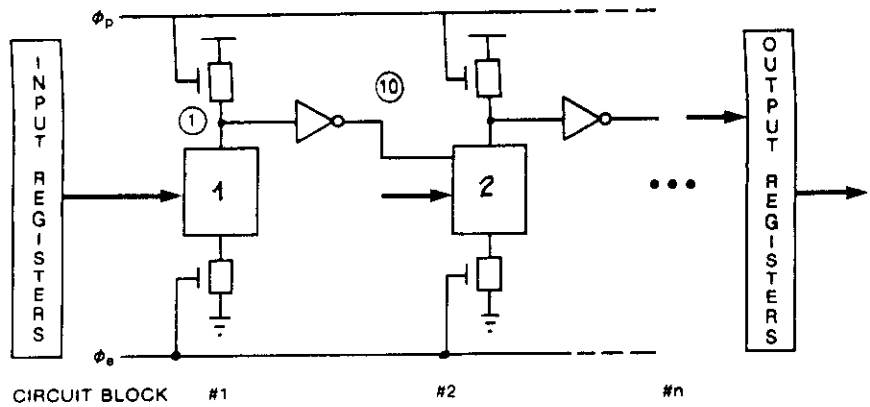


Figure 47. Domino logic with registers.

Two nonoverlapping clocks are used:

ϕ_e : evaluation clock.

ϕ_p : precharging clock.

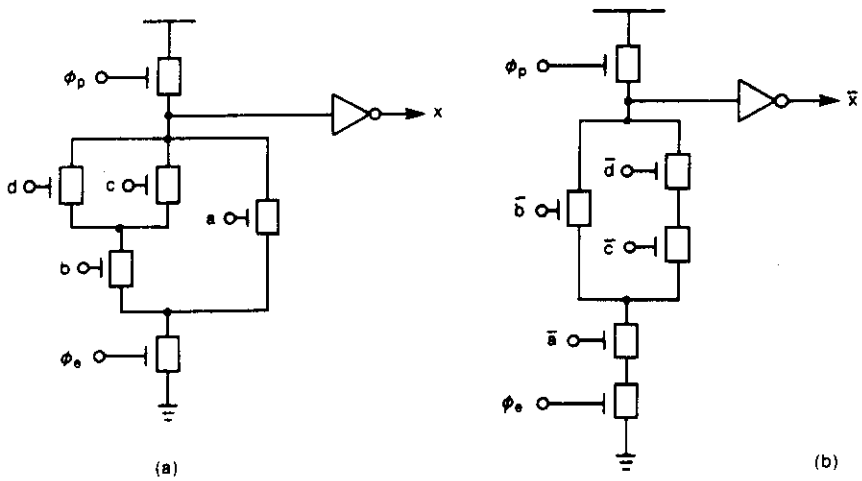


Figure 48.

The precharging clock precharges node 1 in the figure to High. The evaluation clock ϕ_e turns on the bottom transistor and Node 1 is discharged if the logic creates a conducting path from Node 1 to GND. If this happens Node 10 will go to High, otherwise staying in Low.

As Node 1 is precharged to High it is important that all inputs coming from another blocks switch only from Low to High and all external inputs stay constant during the evaluation phase. The block at a) generates the function $x = a + b(c+d)$ while generating $\bar{x} = a + b(c+d)$ demands the block at b). Notice the need for this new block and not a single inverter added to block 1, because Node 10 must switch from Low to High.

Now $\bar{x} = \overline{a + b(c+d)} = \bar{a} \cdot (\bar{b} + \bar{c} \cdot \bar{d})$

Notice also That De Morgan's theorem changes all inputs to Their complements and transforms The original graph to its complement.

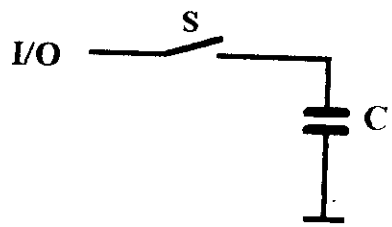
In NMOS, Domino Logic leads to very poor density as compared with static logic circuits. Used mostly in CMOS technology it renders a density comparable to ordinary CMOS static logic.

Notice finally that, if P devices are used for precharging, a single clock will work for precharging and evaluation.

Dynamic logic

Dynamic circuits store charge in a capacitor to specify a logic state. Let's suppose a "1" is stored when your capacitor is charged. Thus, when the said capacitor is discharged you'll have a "0" stored.

A simple circuit to accomplish This is The one shown at right:
Switch S connects capacitor C to an I/O terminal, thus allowing to charge and discharge it.
Charge will remain stored in C with S being open for a time which depends on capacitor leakage currents. These currents are usually reverse junction currents, depending on temperature and fabrication technology.



The small values of capacitance impose a constraint to the circuit: charge must be replenished to maintain a "1" in C, since it "fades away" with time. This means you must "refresh" the circuit with a minimum frequency, otherwise information will be lost.

In the figure at right a dynamic circuit is shown.
Notice the presence of two clock phases $\phi 1$ and $\phi 2$ which play the role of the usual power supply. Thus ϕ_1 and $\phi 2$ are expected to be "sturdy" clocks.

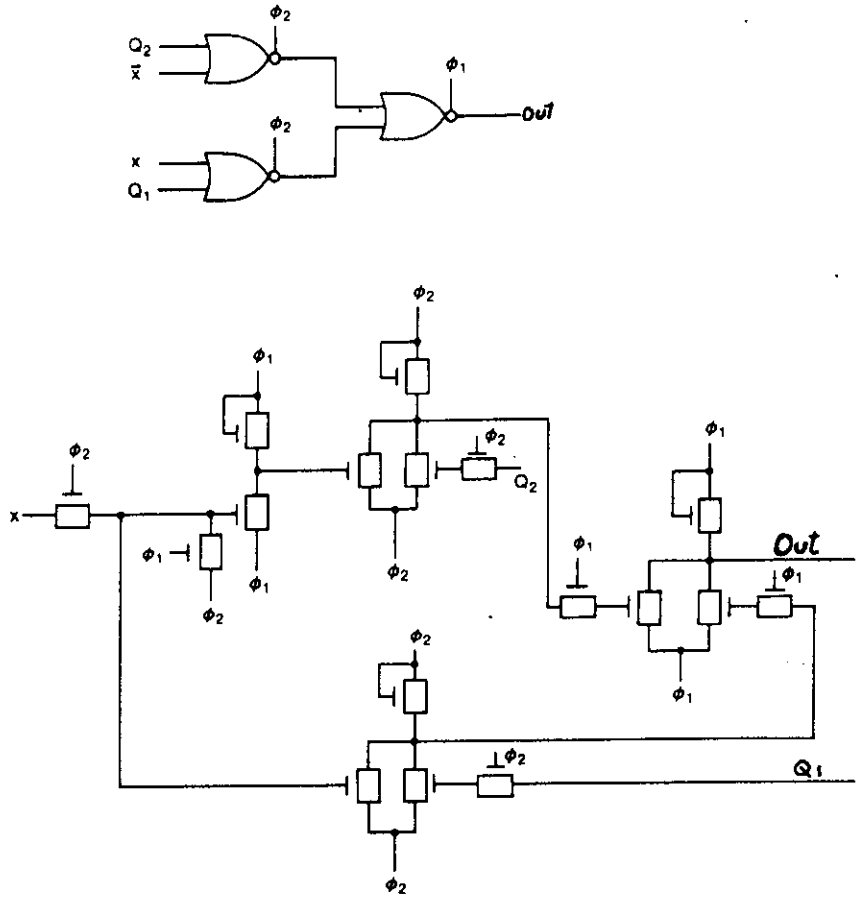


Figure 46.

Memories

You are surely familiar with memories and, for sure, you will recognize their main characteristic: they store information and are regular structures.

Be it a ROM, RAM, DRAM, EEPROM or EPROM, they are highly regular structures.

Using gate arrays you might be allowed to implement ROMs and RAMs, but not EEPROMs or EPROMs Your vendor will provide you with the layout and electrical connections for your "modest" in-chip memory. Nobody would think of a 1Mb RAM or ROM for in-chip uses.

In The figures that follow a DRAM cell circuit and two technology implementations are shown.

Notice The Bit Line (BL) nd Word Line (WL)

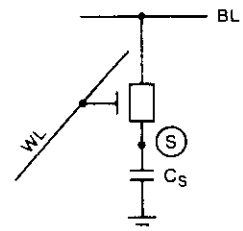


Figure 25. A DRAM cell circuit.

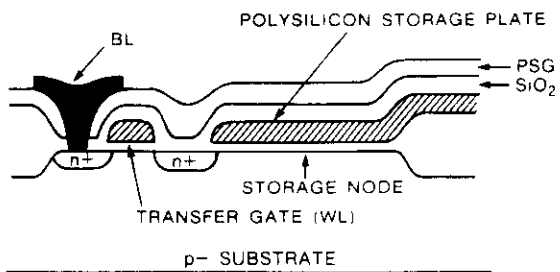


Figure 26. A single-poly DRAM cell.

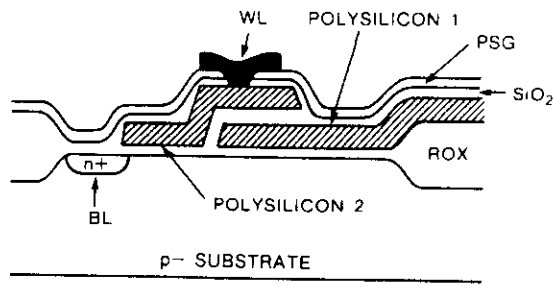


Figure 27. A double-poly DRAM cell.

In order to increase integration i.e, number of devices on the chip, manufacturers have figured out complicated and very ingenious technologies to build the capacitor "inside" silicon and not on its surface. Below you'll find some of these structures illustrated.

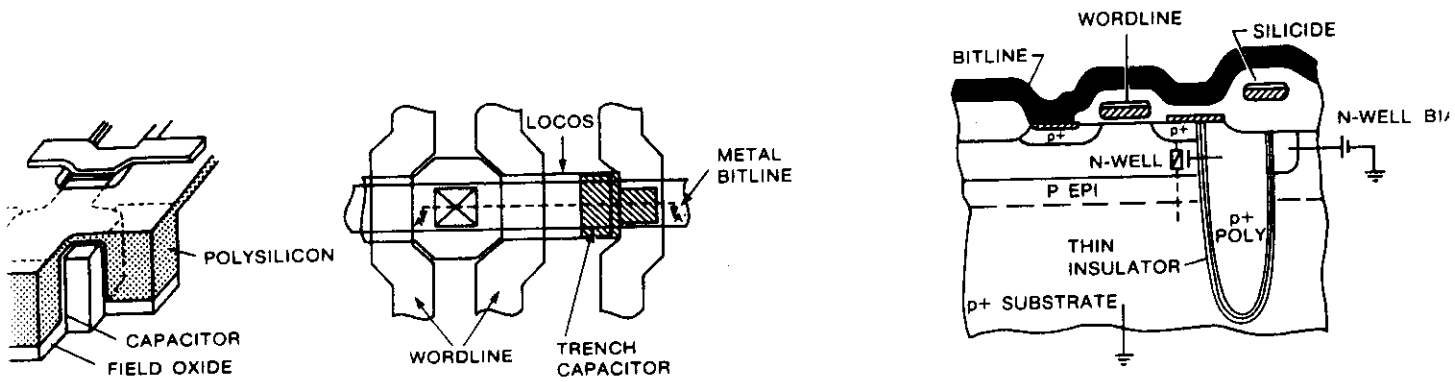


Figure 31. A substrate-plate trench capacitor cell © 1986 IEEE. [23]

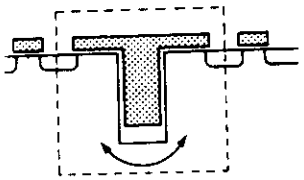


Figure 30. A folded capacitor cell
© 1984 IEEE. [19]

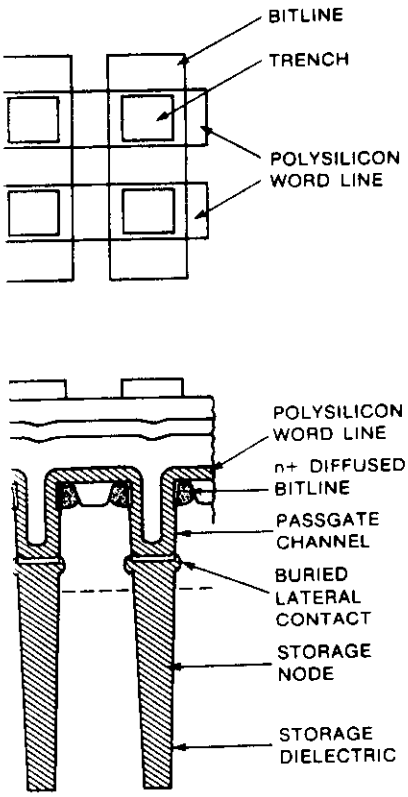


Figure 32. A trench transistor cell
© 1986 IEEE. [20]

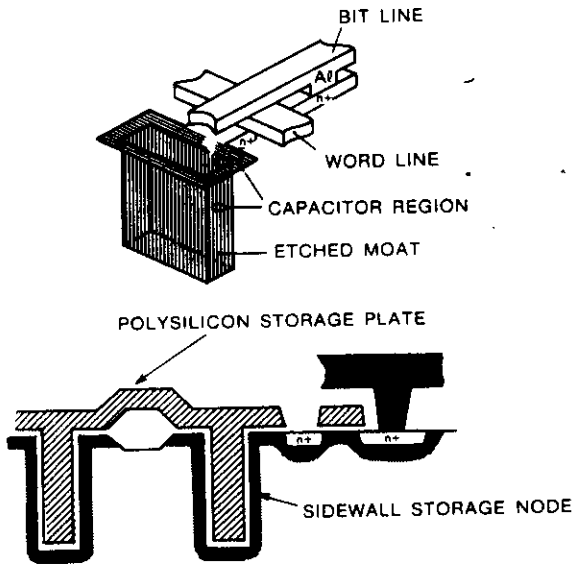


Figure 29. A corrugated capacitor cell © 1982 IEEE. [16]

Applications III

Gate Arrays

Early in these lectures Gate Arrays were introduced as a means of carrying out a design, implementing it and having chips operating correctly with a short turn around and a relatively low cost. Remember that gate arrays do not allow to obtain the most efficient solution in terms of real-state (silicon area) but this is not so important for a beginning. It might be so if you have to produce quite a large number of chips. Then you'd think of translating your gate array design into a full custom design.

As you know, gate arrays are organized and periodic structures comprising both N and P-type MOS transistors (when CMOS) as well as interconnects, crossovers and I/O cells (input/output). I/O cells are responsible for connecting the internal logic with "the outer world", thus their individual transistors differ radically in size as compared with the internal ones. But don't you worry about them: they're already designed so, within reasonable limits, you might consider them in a WYSIWYG (What You See Is What You Get) form.

Gate count has been increasing steadily and tens of thousands gates are quite common nowadays.
Before going deeper we should have a look at some aspects related to design methodology.

Designing with gate arrays

In the next figure, the block diagram apparently illustrates all the stages in a design. There are two main issues missing:

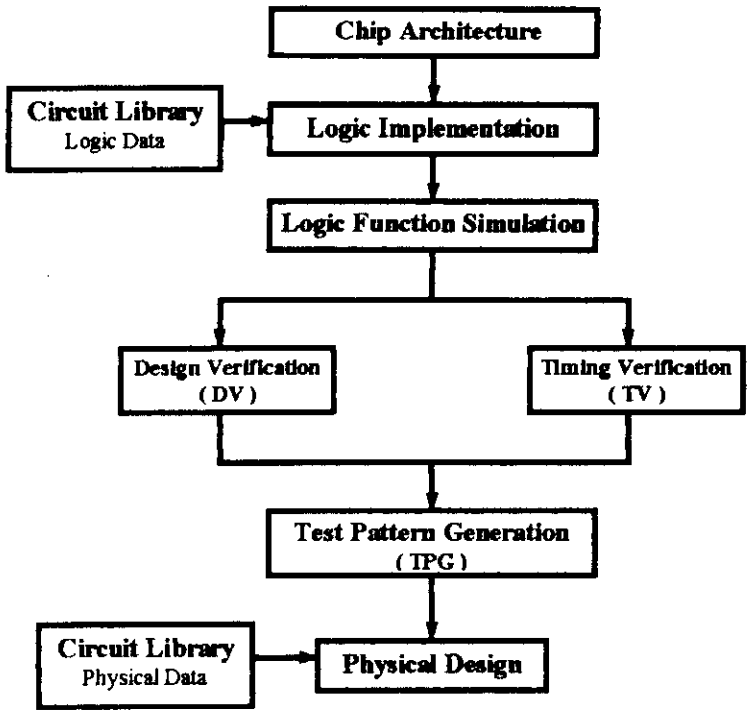
1 - Feedback among design stages

2 - Feasibility assessment.

Feedback is unavoidable: only seldom you get what you intended for at the starting point. Very seldom indeed! Sometimes you get quite a close result to your wishes and so you must "iterate" some design stages to get right to your goal.
Unfortunately, there are times when you don't even get anything that resembles what you intended to get. It should be quite clear to all of us that System Design must be "focused at" in a regular, organized and consistent approach. Success means money saved! Defeat means -even when temporary- lost time and money and the chance to loose your main goal: introducing your chip.

Feasibility assessment is also mandatory: What about if you "attacked" your design by brute force methods and, when manufactured, it proves to be uneconomical? No way to go around this issue either!

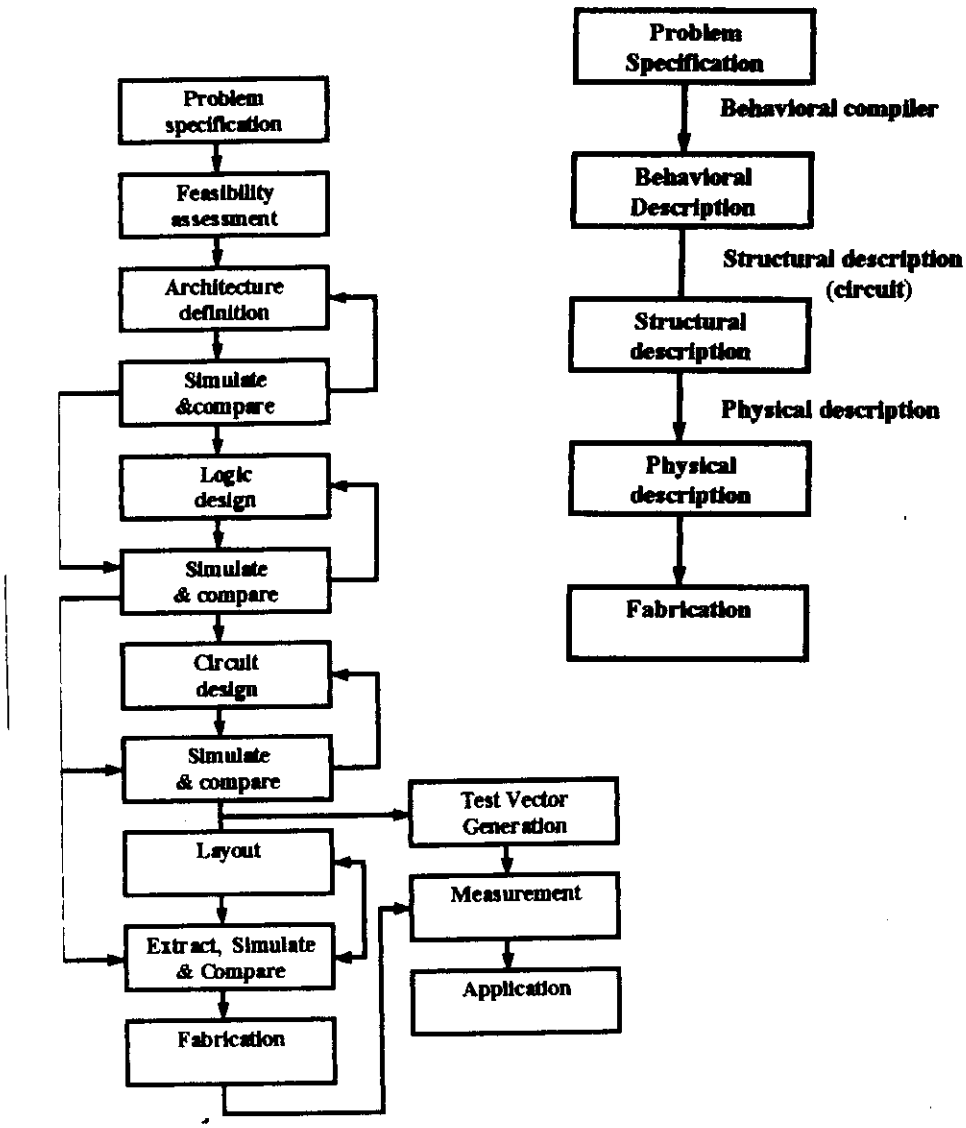
Let's have a look at the simplified design process:
Note that all stages are needed, but there's something lacking: feedback and feasibility assessment



SYSTEM DESIGN

System Design

In the next figure a more general representation of the process for system design is shown.
Note the several feedback paths in the diagram as well as the presence of an assessment stage and a final application stage. Remember that you still might have some nasty surprises when you plug-in your VLSI in your final PC Board.



The process illustrated in the figure above involves several degrees of abstraction:

The topmost refers to the behavioral or functional stage, followed by a lesser abstraction level: structural, where you specify architecture, and keep coming down to schematic design. The third abstraction level is the physical one, where dimensions and layout are accomplished.

TABLE 5.1 ASSESSING FEASIBILITY

INFORMATION NEEDED
Maximum clock/data transfer frequencies; through how many levels of logic?
Critical timing paths
Noncritical timing paths—how loose?
Loads that have to be driven, including expected PCB runs
Available input drivers
Restrictions on pin-outs
Prioritized list of technical, economic, and timing factors
Any analog circuitry needed
Are there especially difficult parts that can be put outside the gate array chip?
Max./min. temperature
Voltages
ANALYSIS AND PARTITIONING: THINGS TO LOOK FOR AND CONSIDER
High frequency clocks: often are divided down into manageable frequencies
Take slices out of a memory so that parts of it can be put on a common chip with random logic
Can two similar circuits be made into one composite design with mode select options?—decreases development costs
What is roughly the ratio of "global" path lengths (long chip runs) to local path lengths (< 2 cells distant)? Will selected array accommodate this?
Where should function cell (macro) paralleling be used, if at all?
High-fan-in and high-fan-out gates
Loads that must be driven
Number of I/Os
What are the prioritized reasons for using a gate array—speed, power, cost, etc. . . .
Where are the clocks? What do they drive? Must they be regenerated locally? How will they be distributed?
Are there one or two ICs that if left off the gate array can make the gate array chip feasible? Result might be one or two standard ICs plus one gate array IC replacing, say, 150 standard ICs.
If two or more PCBs are being combined onto a few gate array chips, partitioning across PCB boundaries can be done because now all PCBs are on chips on the same board
Any A/Ds or DACs or other analog circuit embedded?
What percent of array cells will be used?
Where are the critical timing paths? What possibilities exist for races and spikes? Do certain paths' timings have to track?
Can certain signals be multiplexed to save pin count?
What package type is required? Will the die selected fit it?
What cooling is available? What is maximum power dissipation? What are ambient temperatures (max./min.)? What is maximum junction temperature?
What voltages are available?
Must certain package pins be predetermined signals? (affects layout)
Any unusual EMC (electromagnetic compatibility) or other environmental conditions?

Feasibility assessment

Feasibility assessment has two major aspects to consider. economic feasibility and technical feasibility. Both of these aspects must shake hands with each other. Neglecting anyone of them might mean chaos, bankrupcy and being out of business.

Let's have some words on economics: No engineering task is divorced from economics. It is mandatory to recall that minimum production volumes are necessary to support for a VLSI design to be carried out. Minimum volumes will help you decide among Gate Arrays, Standard Cells or a Full-Custom approach.

An analysis carried out by Texas Instrument for a sample case is depicted below.

Cost Component	40 pin, Gate Array (877 used gates)	59 LS SSI/MSI (477 used gates)
- Component	$ 25,00	$ 14,10
- Insertion		
Buying	0,10	$0,03/unit — 1,77
In. Tests	0,25	$0,08/unit — 4,72
Inventory	0,10	$0,10/unit — 5,90
PCB test	0,25	$0,10/unit — 5,90
PCB redisign	0,55	$0,09/unit — 5,31
Insertion Sub Total	2,00	47,20
— PCB 3in² @ 0,50	1,50	88,5in² @ 0,238 — 21,12
— Energy 0,5W @ $3	1,50	1W @ 1$ — 1,00
Sub Total PCB+Energy	3,00	22,12
- Reliability Costs 0,001%/1000h $400/h	1,00	0,001%/1000h $400/h 8,20
	$ 31,00	$ 91,62

The results of the comparison are quite clear, Don't you think so? Economic feasibility answer The question: Does it pay to do it on a VLSI basis? Technical feasibility, on the contrary, answers the question:

Can it be done?

You must have certain information available for discussing the issue. The next page shows a list of "Information Needed" and "Things to look uf and consider" in Analysis and Partitioning.

- Keepin mind That you shouldn't try to use more Than 80% of the total gate count of the array you choose. This, of course, depends of what your manufacturer offers. This 80% is a practical limit in order to be free from interconnection troubles, although some manufacturers claim to be abk to use more than 90% of their total gate counts in a given gate-array. Remember you should include additional logic in your chip (around 25% to 30% of your gates for the design). For instance: your design needs 4000 gates, which will require around 1000 additional gates for testing, so you'll start trying with a gate array of 5000/0,8 = 6250 equivalent gates or whichever is nearer.

A good design must take into account several techniques:

Hierarchy: Is implicit in the thought of having different levels of abstraction. Hierarchy means breaking down complex parts or modules of your design into less complex blocks or submodules. This not only eases understanding of the whole, but makes it easier for you to design. Everytime you divide into simpler blocks you tend to approach well-known and easy-to-use simple blocks which, for sure are waiting for you at your CAD library.

Modularity: Once you choose and form the "right" modules (which are not unique for a given design), their interactions with the rest of the modules can be well characterized. Characteristics, properties and what each module does should be well specified.

Regularity: An example of high regularity is a RAM (or a ROM). Gate arrays are also good examples of the use of regularity (from the topological point of view. Regular structures, like PLAs (programmable logic arrays) should be favoured during design.

* Remember that characterizing your interfaces allows for a better understanding of the module and is a must when a design is divided among several designers.

Packaging

Packaging is to be considered as an important step. Pin count, as well as the type of package are important and might prove to be relevant. Furthermore, you want to use your circuit in your application, Don't you? So you need to think if you're going to use sockets or you're going to solder your package directly to the PCB. If so, will it be a Surface-Mount or a Through-Pin type? Plastic or ceramic? These decisions usually mean money and might be different for prototyping and for production.

Power dissipation and chip temperature

The temperature at the junction of the device is calculated by the expression

$$T_j = T_{amb} + \Theta_{j.a} \cdot P$$

where T_{amb} = ambient Temperature
Θ_{ja} = Thermal resistance from junction to ambient
P = Power dissipation

Θ_{ja} is given for each package type and for equal pin counts and package type is smaller for ceramic than for plastic packages.

The expression for T_j can be applied to the whole chip supposing that the power source is a point in the chip.

This is not the case, as you may realize. Anyhow, you may carry on an initial calculation remembering that:

- $P = CV^2 f_{max}$ and assigning $C = 1pF$ to each node. This brings out larger values of C than the real ones
- From 10% to 30% of the total number of gates switch in CMOS at the same time.
- Output buffer power dissipation is considered by itself, since it may be quite large.

This gives an estimate of the total power dissipation, which is usually a worst case one.

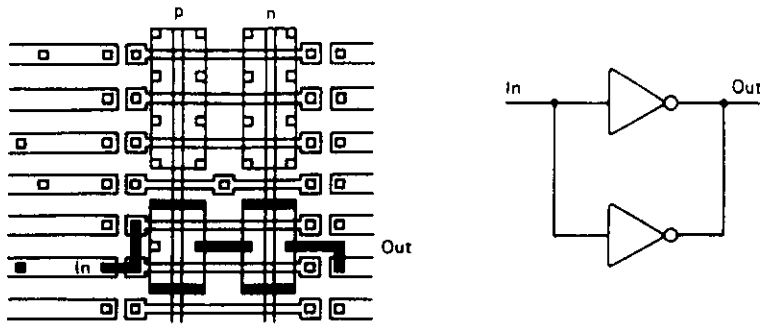
A better analysis demands special programs for calculating power distribution, power dissipation and chip temperature.

All gate arrays, regardless of technology and manufacturer, consist of several basic elements:

- *Pads to enable connection to hand from the "outside world" (package or hybrid).*
- *Buffer devices to drive the higher capacitance of the outside world.*
- *Distributed power and ground buses.*
- *Transistors and diodes.*
- *Underpasses to cross under the power and ground buses without contacting them or more commonly a second metal layer.*

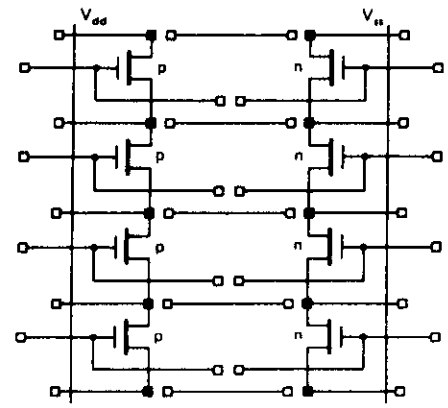
Many arrays dies have in addition to above:

- *Test devices.*
- *Voltage-level shifters to enable operation from only one external voltage source (as in the case of ECL and ISL/STL) or to make the devices compatible with TTL logic levels.*

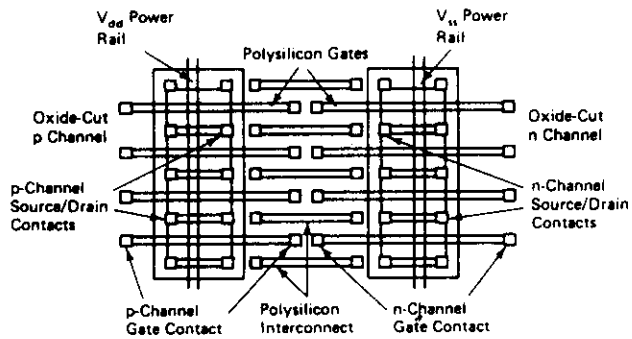


(d)

Figure 3.8 *(cont.)*

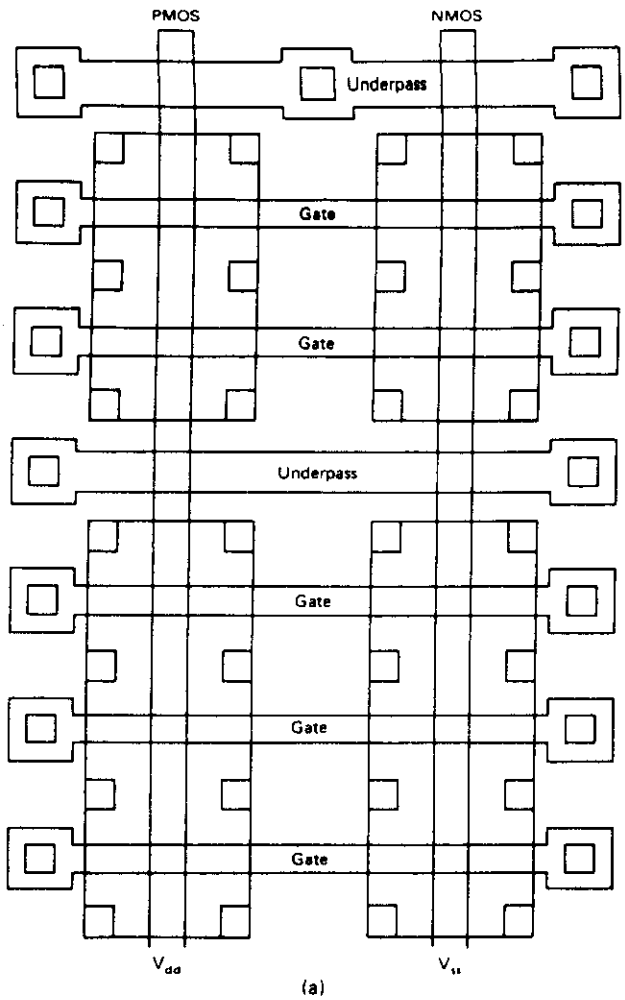


(a)



(b)

Figure 3.9 MITEL CMOS gate array structures: (a) array cell mask layout (unprogrammed); (b) a layout of basic array cell; (c) schematic of basic array cell. (Courtesy of MITEL.)



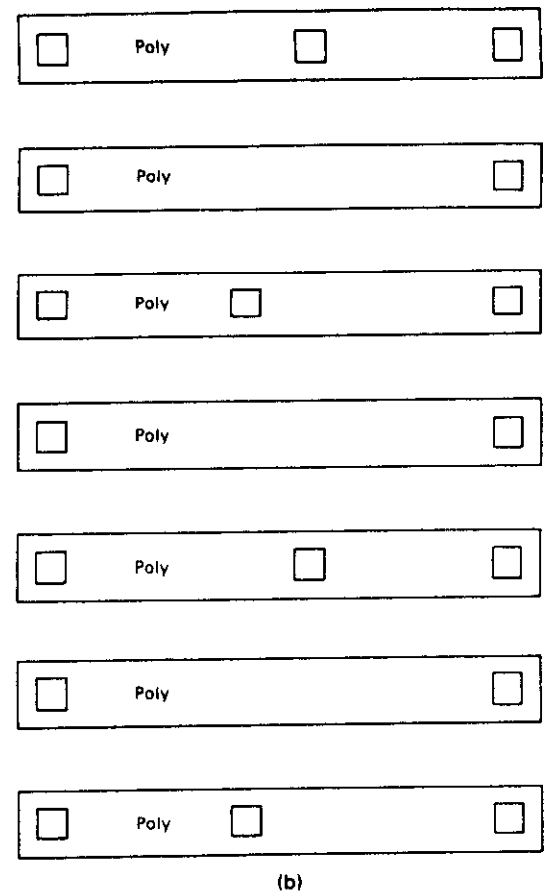
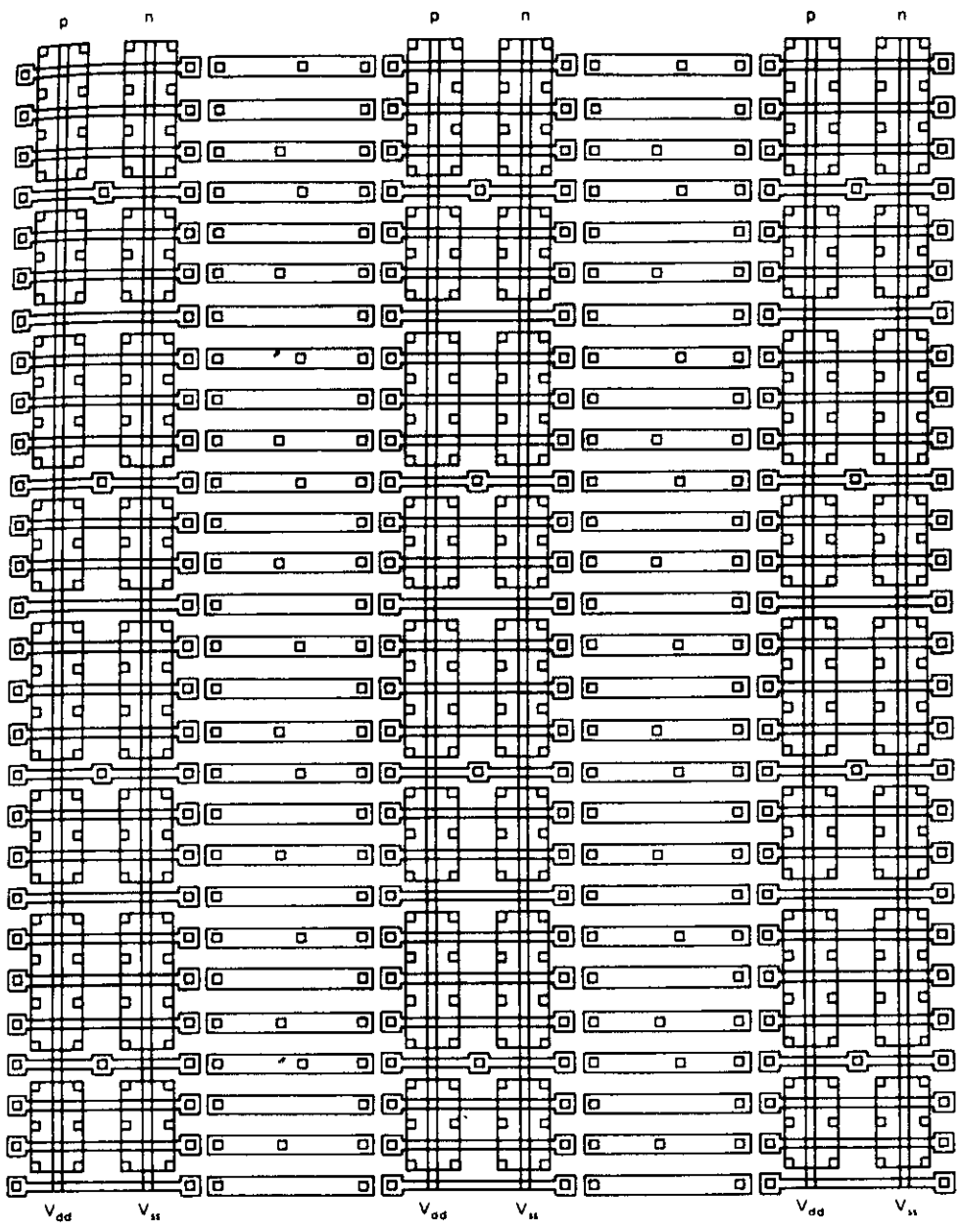


Figure 3.8 SPI's single-layer metal HS CMOS grid structure: (a) device cell; (b) crossover cell; (c) section of array at 200×; (d) double inverter. (Courtesy of Semi Processes, Inc.)

Figure 3.8 (*cont.*)



(c)

Figure 3.8 (*cont.*)

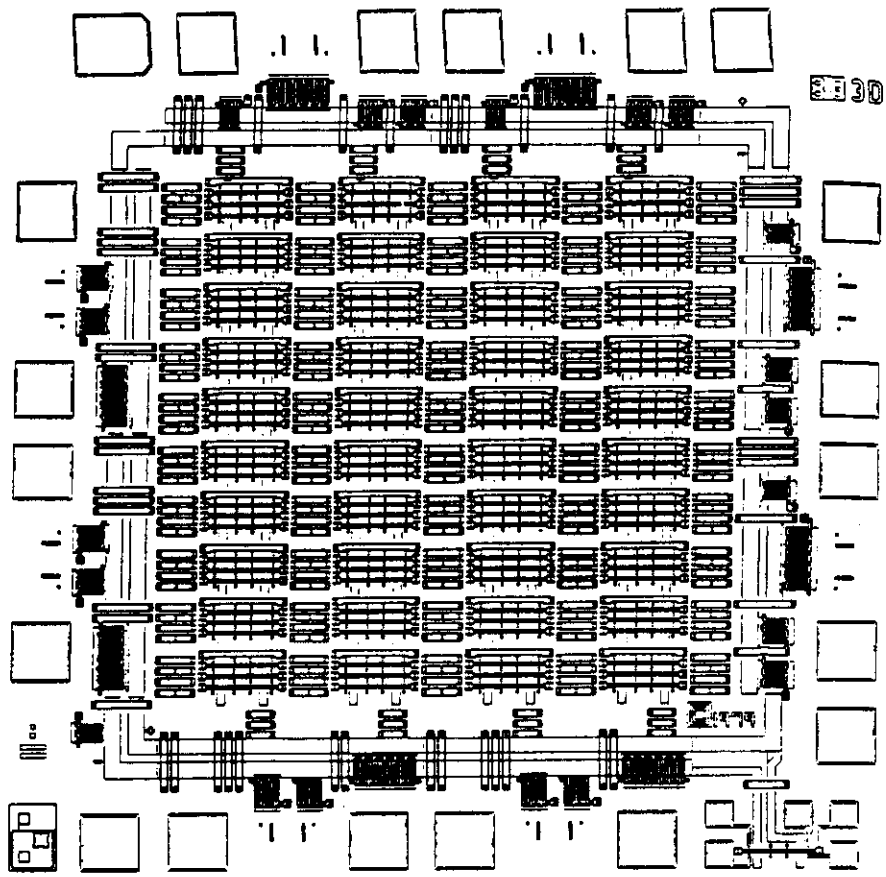


Figure 3.1 *Fabricated* picture of gate array die showing circuit elements. The figure was *created* by *removing* 1400 of the 1440 cells which are present in the layout of the *actual* die and similarly *reducing the number of I/Os so that a* complete die could be shown. *This is not a picture of an actual die.*

CAD Tools

Available CAD Tools usually respond to the block diagram previously shown for System Design: they include specific design tools. You may try to ignore some of them, one of them or all of them. It's no good to do so. Ignoring schematic editors in layout design is impossible, like ignoring simulation for determining the response of a block or whole design, be it electrical, switch-level or logical simulation is nonsense.
Let's have a look at the main design tools usually encountered in practice:

Switch level simulators simulates transistors as switches where RC effects in the design are considered by lumped RC in the nodes.
Although bringing out some timing information, it is not as accurate as might be the one resulting from an electrical simulation.

Electrical simulators substitute each device by the corresponding model and forms a conductance matrix which must be solved to get currents and voltages for each point of analysis.
Timing results are more accurate but take time to calculate, the more so as circuit complexity increases.

Layout editors: Make it possible to "materialize" in a floor plan the circuit in the form of geometrical shapes in different layers or levels. Individual devices are represented by figures, one for each photolitographic process used. Editors allow for the creation, transformation and manipulation of multi-sided, multi-level polygons. Layout rule checkers may be available off-line or they be on-line, signaling device layout rules violations as they are introduced during the design process.

Schematic editors provide the means for drawing and editng your schematic, including electrical rule checks. This is one of your early design stages, but is many times the best way to specify your circuit, since layout by itself is cumbersome to interpret. They usually provide netlists as outputs also, where a description of the electrical connections of individual devices is made.

Netlist comparison makes a comparison between your physical implementation (layout) and the design you intend to accomplish via a schematic description or a symbolic description of some sort. This is a must if layout is not generated automatically. In other cases, like in standard cells, gate arrays and silicon compilers, where you don't create layout by yourself you don't carry out a netlist comparison.

Circuit extractors go over your layout and extract the electrical circuit you've really got. This detects connectivity errors and evaluate real capacitances (interconnection and parasitic) in the nodes, so you have more accurate data to simulate for timing and delay purposes.

Other facilities are supplied for choosing the right type of package, power distribution etc. on the chip.

Good Design Practices

Any human action is prone to introduce some kind of mistake in the process of designing a VLSI. The more so if one tries to skip steps like simulation forgetting that the use of CAD tools is mandatory.
Well now, since we're not that kind of people let's admit we'll make use of CAD tools in order to get a design working in the shortest possible time. Does this guarantee your success?

Definitely not! It helps a lot, but still there are certain things you should do, as well as others you shouldn't do.

Let's have a look at the things you should do:

1- Use synchronous design. It will help to keep away some timing problems.
2- Use a general Reset whenever it's possible. It will help you when testing.
3. Don't use clock frequencies inside the chip higher than the maximum guaranteed by the manufacturer. Use a single clock for the whole circuit. Buffer it to achieve same gate delay per element.
4. Make a generous use of simulation. Don't believe you can skip it off.
5. Keep track of your timing and internal loads on your nodes.

6. When talkers (T) and listeners (L) are connected to a bus it is a principle of good chip design to have all talker with enable lines.

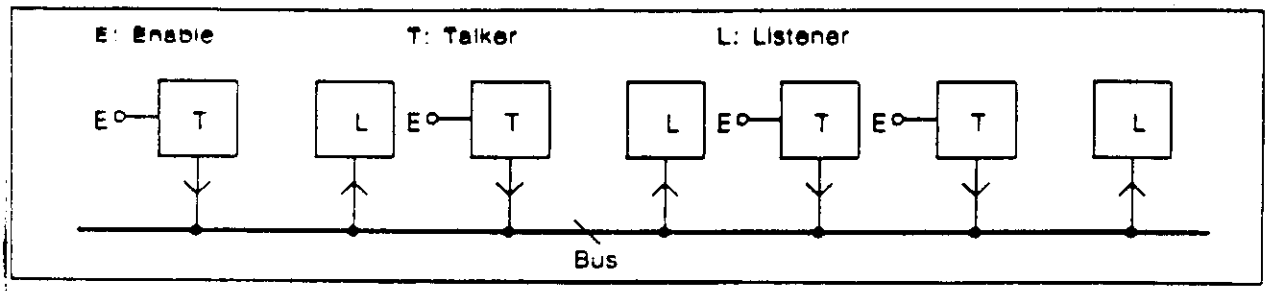


Figure 8.13: Talkers and Listeners on a Bus

You can lower the loading on buses by grouping all the talkers and all the listeners together and introducing a buffer on the bus between them.

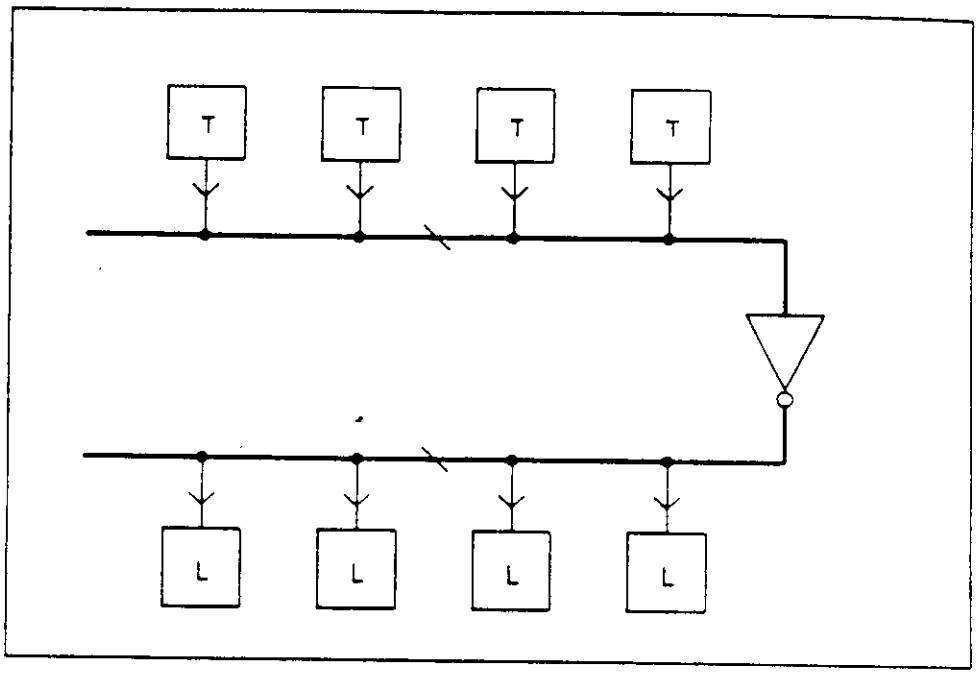


Figure 8.15: Buffering Talkers and Listeners on a Bus

7. Use the right size of package for your chip. Use the necessary number of pads. Keep in mind that you can use multiple function pads depending on your choice.

8.- Build up your chip from a reduced number of carefully designed and tested elements

9.- Control each element of the circuit by its own enable line

10.- Use a synchronous reset for each element

11.- If a single clock input is used there's no need for asynchronous elements in the design.

This principle can be extended by placing tristate buffers linking groups of talkers to the bus. See Figure 8.16 for an example.

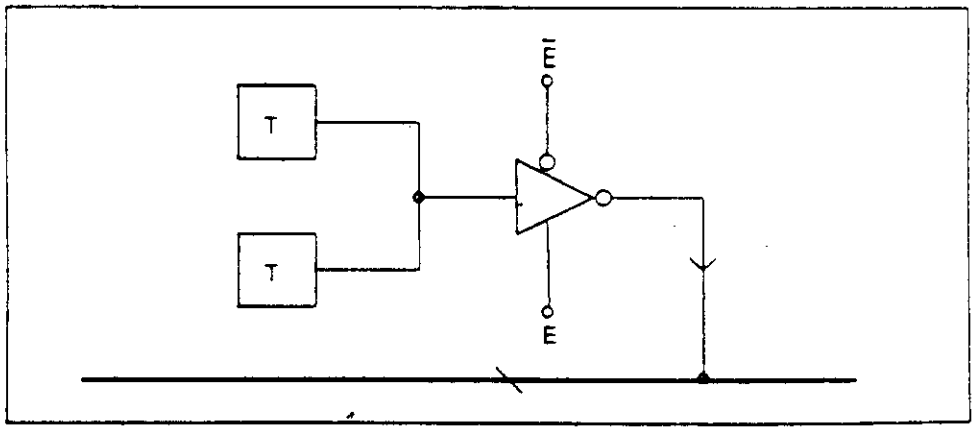


Figure 8.16: Buffering Talker Access to a Bus

Common sense, experience and, why not?, a little luck will also help in achieving a working design in the palm of your hand.
But, what if you make a mistake?

If You Make a Mistake ....

Obviously, no one purposely makes a mistake. But, as we said at the beginning of this lecture, sometimes they do occur even when using the newer CAD tools.

Reasons for this exist and are there to annoy you.

First, there are generally more nodes in a system that can possibly be simulated within a reasonable time and cost. Murphy's law says that the ones the designer thinks are trivial and/or not important are those that do not work and in which the program is most interested

Second, the increased densities of the available chips exacerbates the problem.

Third, on occasion, changes are required to the design even before finishing it. For example: The competition came out with a superior circuit, so you must introduce changes in yours

Fourth, on occasion, even gate array designers -and you- make mistakes.

So, what to do next?

- Find the problem
- Determine a fix for the problem. But be sure the fix works using simulation or other tool.
- Check to make sure something right previously has not been made wrong by the fix
- Remake mask(s) and a new set of parts. -go-

Clock distribution strategies

Delivering clock signals anywhere in a large chip demands special attention from the designer. Long lines introduce delays which may be considerable enough to influence, and not for good, circuit performance. Skew in clock signals may cause misoperation of the circuit. Slow clock signals make edge sensitive systems to work poorly, besides producing a high current flow when both P- and N-type transistors are ON. Spreads in threshold voltages can exacerbate skew problems in clocks since they introduce a change in the time a latch or counter samples.

Three approaches may be used:

1. Brute force, using huge buffers to drive the long lines with high capacitive loads. This is not a good approach because of the large currents in the lines

2. Geometrical Buffering, illustrated in the figure, is useful when fanout have a value up to 32.

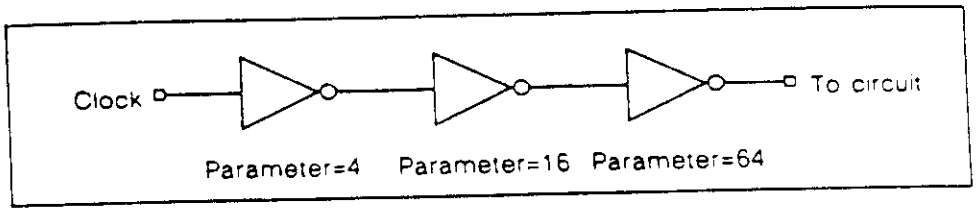


Figure 8.19: Geometrical Buffering on Clock Line

3. For fanouts greater than 64 high current in clock lines leads to a less reliable circuit. Use a tree fanout like the one illustrated in the next figure.

4 For fanouts between 32 and 64 choose any approach: geometrical or tree fanout.

5 Simulate to verify your skew and quantify its value.

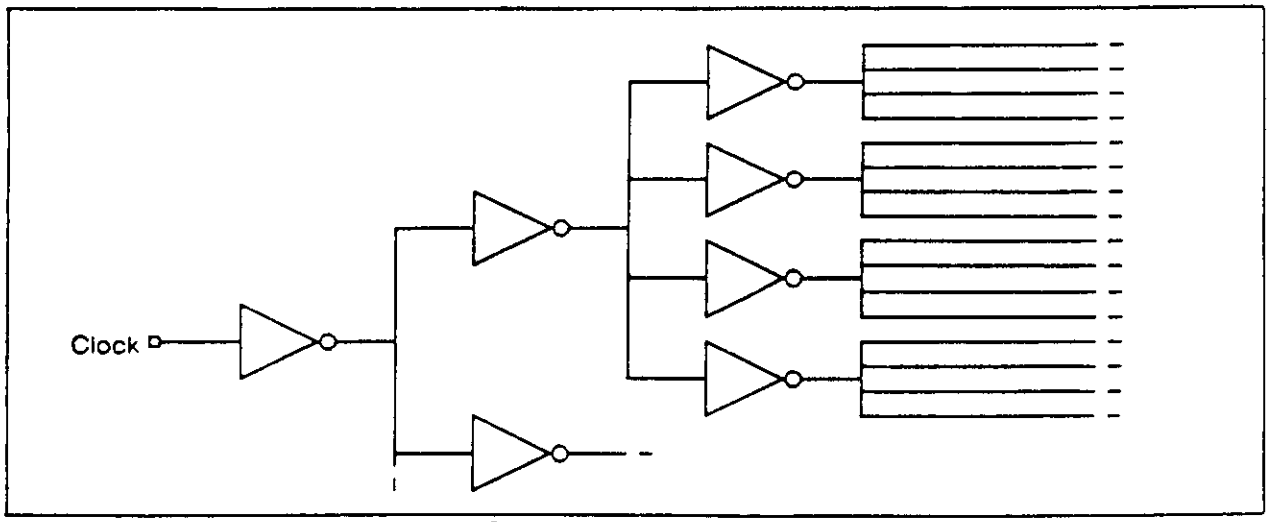


Figure 8.20: Tree Fanout on Clock Line

Enabling each individual block is useful, as illustrated below

Enabling Bus Talkers

In order to ensure that one (and only one) talker is driving the bus at any time:

The recommended way of generating the enable signals is via a decoder.

Using a decoder to enable talkers on a bus ensures that there is no contention on the bus, and also that it is never tristate. If necessary, unused output lines on the decoder can be ORed together, as shown in Figure 8.14. This circuit will enable talker 6 when (non-existent) talker 7 is enabled. This error is, however, preferable to the bus going tristate in this situation.

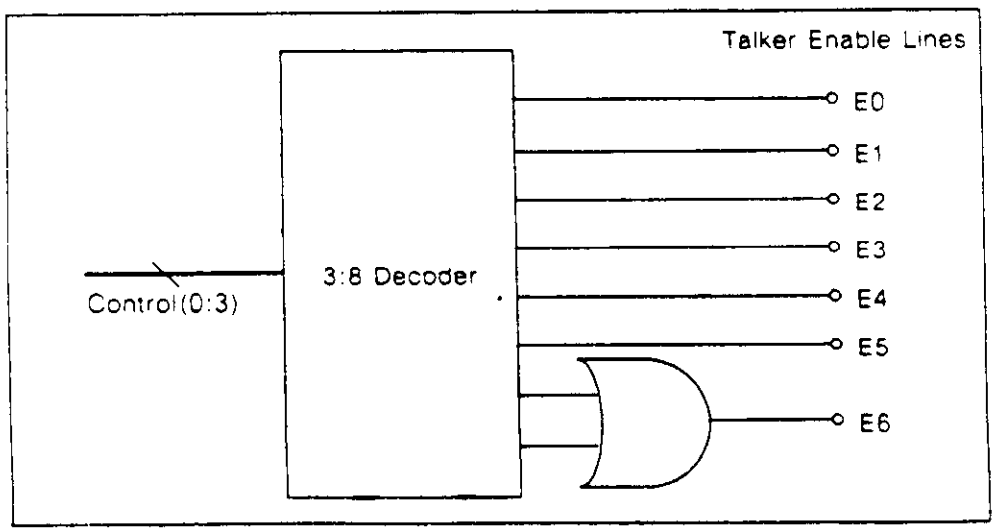


Figure 8.14: Enabling Bus Talkers using a Decoder

Partitioning

An important issue in VLSI design is partitioning. How are you going to design: in a single chip or in more than one chip? Of course, this may be dealt with selecting the greatest available chip (if you're working with gate arrays) which, by the way, may not be of interest to you because of several reasons. So, if you decide to partition your design There's a number of definitions you should consider.

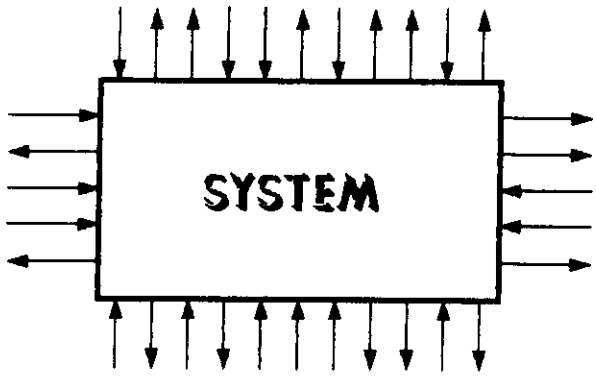
Cell or equivalent gate: is usually referred to a 2-input NAND gate.

Thus:

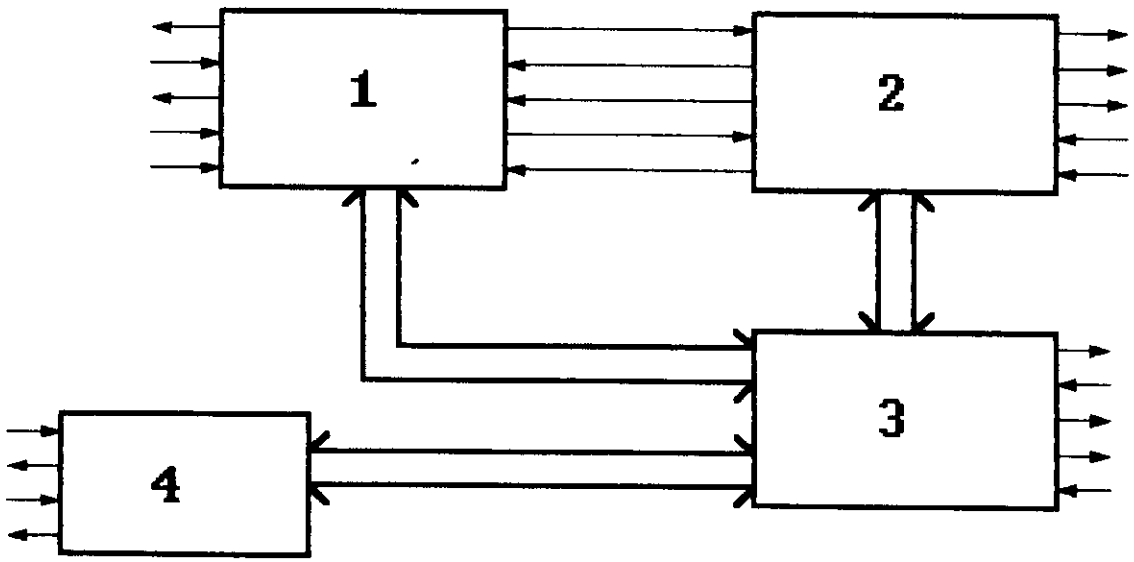
2-to 3-input gate	1 cell
4-input (NAND or NOR	2 cells
2-to 3-input AOI	2 cells
4-input AOI	3 cells
XOR	2 cells
D flip-flop	4-6 cells
4-to-1 MUX	3 cells
JK flip-flop	6 cells
3-to-8 binary decoder-	10 cells
16-to-1 MUX	≥40 cells

This is just an illustration and may vary somewhat depending on the vendor you choose. Anyhow, some table, program, slide-rule or other means will be provided by the vendor to help you estimate the number of equivalent gates you need; i.e.: the size of gate array you need.

Supposing the system you're designing is the one shown schematically in the figure below you must partition it if necessary. Perhaps a good start is trying to divide it into blocks which accomplish a definite function.



An example might be dividing your system into the four blocks in the next figure. With fewer inputs each, the question still remains: How many chips for the design?



Try to layout on a larger chip initially keeping in mind that you must estimate your total gate and cells count and that you might as well add a generous 20 to 25% of the total cell count of your design for test purposes. About this we'll talk on the last lecture. Laying out on a larger chip initially might prove useful to avoid topological and space constraints since you might later on change your design to a smaller gate array.

If you (or the program) have specified a given package type and size this will limit the size of the die that can be accomodated both to a maximum and to a minimum. In the latter due to limitations imposed by the vendor on the maximum length of your bond wires from pads to pins.

Pin count and pin-outs. Determination of pin count is the sole responsibility of the designer. Pin-out determination of both, the die and the package is usually left to the designer to decide. Eventually, a change in pins might be suggested to ease somewhat routing interconnections.

Pin-outs-limited circuits

A fairly common problem in partitioning is what to do when the pin count of the circuit exceeds that of the package into which the die will go. One possible solution is to use that will fit in to

The package but that has a larger array than the one initially considered. This should usually bring out a fresh supply of extra pins

Another solution is multiplexing signals, thus using a single pin shared among several outputs. Multiplexing is controlled via signals generated internally.

Making chip-pads bidirectional might help if you can afford three-state circuitry and its control inside the chip requiring extra pins for bidirectional control. Usually not useful unless several pins are made bidirectional.

Now: Back to partitioning itself!

Partitioning in itself is a highly technical, experienced and, why not, a little of an art. How to partition can not be decided by a recipe. Of course, rules help a lot to achieve success. We had a look at some of them.

But, cheer up!, no one has been, and presumably won't ever be, born an expert!

You are right if you guess there's no standard time or cost associated with the tasks involved.

There's something sure: beyond a certain point it simply doesn't pay to keep trying to patch up a layout or design if there happen to be a great many changes to be made.

A final advice: CAD tools, as essential and powerful as they are for the design of the more complex chips are not to be used in place of thinking.

Think first and then get your computer to work just on what you do really need.

Most of us have been, sometime, bitten by the small but powerful computing monster. So, we have run endless simulations whose results don't allow for a clear analysis and interpretation and are, thus, of dubious help. Spending a little more time in thinking will help a lot anyone to get a better result

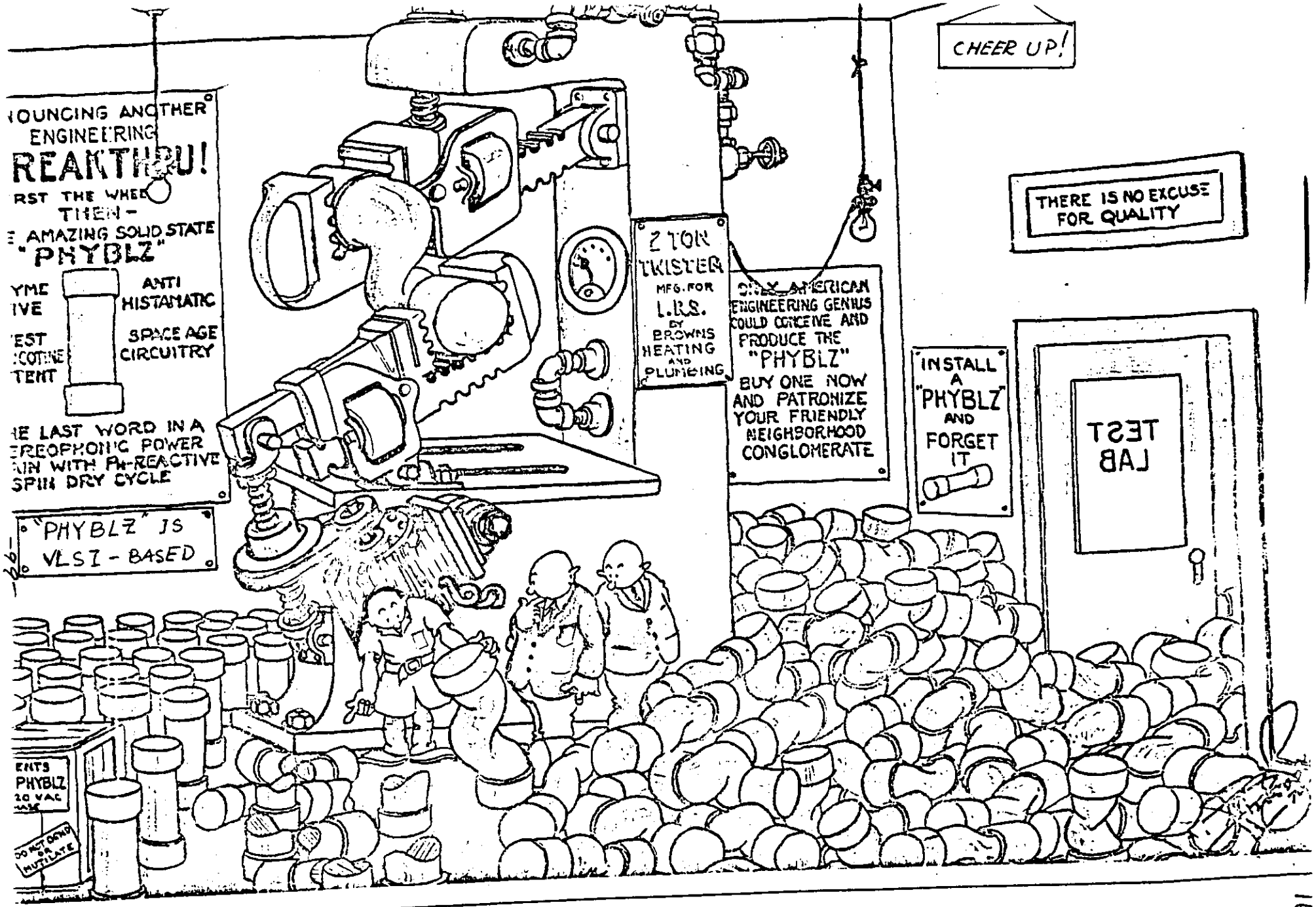
Is QUALITY for you?

In this world of ours a lot is talked about Quality. Total Quality is something many people talk about and it's up to us to introduce Quality in VLSI Design.

If we accept the fact that your vendor, which you've chosen for a number of reasons, offers you a high-quality product, then it's up to us, as we just said, to make a high-quality design.

Not every design we think about will function properly" so we must resort to lots of simulation in order to check this out. Being too optimistic might induce you to drop some of the necessary simulations. Being realistic will help to achieve the desired success. Test vectors and Design for Testability, as well as, worst-case simulations will help testing our VLSI and checking on it before it is produced.

But before we start dealing with Design for Testability basics let's have a look at what happened to the producers of "PHYBLZ" which, as they claim, happens to be VLSI-based. Please have a look at the next figure! Next we'll get a refreshing splash into Design for Testability.



"Too Bad We Had To Wreck The Rest Just To Find These 5 Bad Ones"

Design for Testability (DFT)

"Every circuit produced must be tested; otherwise, it represents a potential failure". This statement is a well-known fact regarding ASIC design. It's easily understood that testing is an important step regarding an integrated circuit comprising tens of individual components; more so when we deal with tens of Thousends components. The same thought applies to circuits designed on the basis of standard integrated circuits.

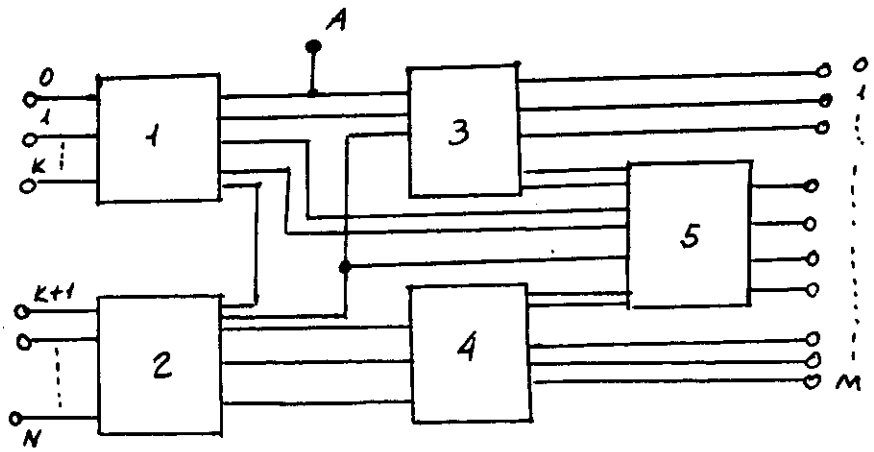
Unfortunately designers forget - from time to time - Their designs are prone to malfunctioning and Think in terms of "I did it and did it allright; Therefore testing is not of great importance". This concept leads to a delightful failure and, perhaps, to sound finantial losses and going out of business.

Testability is a must in today's VLSI design: There's no way, no other way, to cope with the issue.

Tests must be Thought of in early design stages, implemented in The design and used Thereafter in development and production stages of your personal design. Remember: The test engineer is no wizard at all!!

Basic concepts

Let's introduce some basic concepts which will make life a little easier for us.
A hypothetic circuit, comprising N inputs and M outputs is shown in the figure.



Primary input : Input connecting the circuit with the "outer world" [0, 1, ... N]

Secondary input : Not accesible from outside i.e: node A as related to block 3.

Primary output : Output connecting the circuit with the "outer world" [0, 1, ... M]

Secondary output : Not accesible from outside. i.e: node A as related to block 1

Observability ; The ability to "observe" the state of any internal node through primary outputs

Controllability ; The ability of determining the state of any internal node through primary inputs

Test Vector: A set of input states at a certain time t

An important issue, the generation of test vectors, is the major concern of both the test engineer and the circuit designer.

There's a golden rule regarding Design for Testability:

> "Any applied measure should be directed to increase both, observability and controllability of the Device Under Test (DUT)"

A simple set of rules for accomplishing this goal is the following:

1. Use additional inputs to select working modes (Test/normal)
2. Use additional outputs to enhance observability
3. Use multi-function outputs whenever possible
4. Use multi-function inputs whenever possible
5. Divide the circuit in simpler blocks easier to test (Degating)

The implementation of these rules demands extra electronics. Perhaps up to 25% or 30% of the total gate count. But: It pays back!!

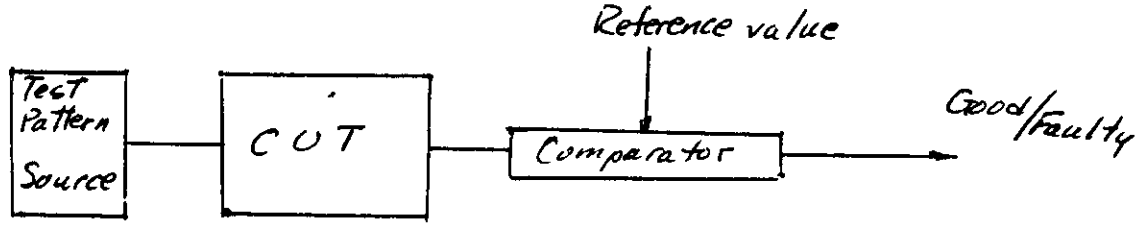
Design for Testability Techniques

An attempt should always be made to maximize the testability of a circuit and to minimize the cost of its testing.

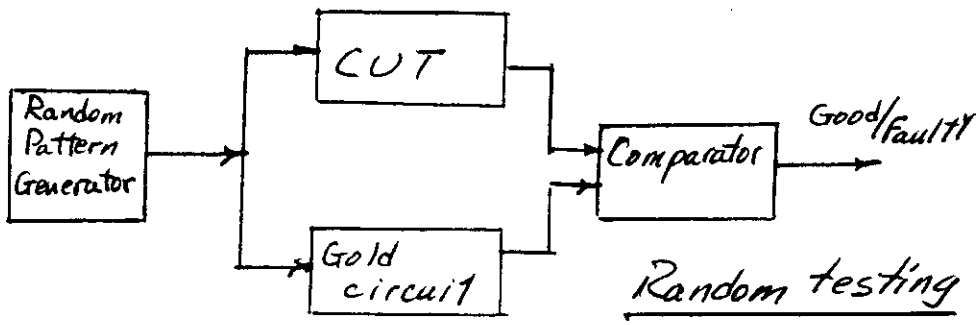
DFT requires extra circuitry, so increased hardware cost and some performance degradation are to be expected. Testing cost and the cost of design and production should be estimated.

Let's review some of the techniques used in DFT and Testing itself.

External-testing: The circuit under test (CUT) is supplied with a series of test patterns and the responses are compared against reference values. These patterns and reference values are produced either by software-based or hardware-based methods.



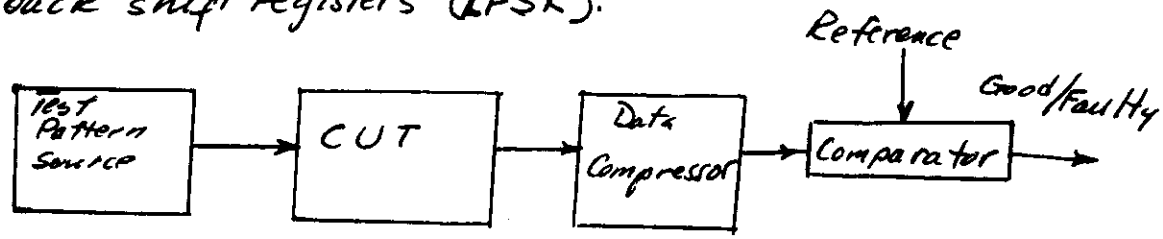
General testing Scheme



Random testing

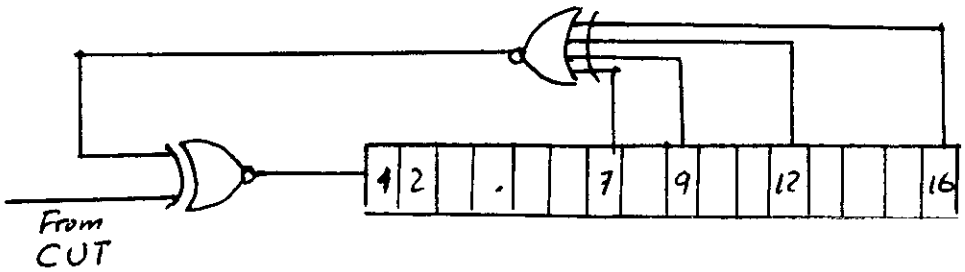
If the reference responses are generated by any software means, this type of testing is called deterministic testing.

Compact testing: Compressed response data are used for comparison. Signature analysis is a widely used method of compact testing. Data compressors can be implemented with relatively simple circuitry, such as counters and linear-feedback shift registers (LFSR).



Compact testing

Shown below is a 16-bit LFSR whose contents are called the signature.



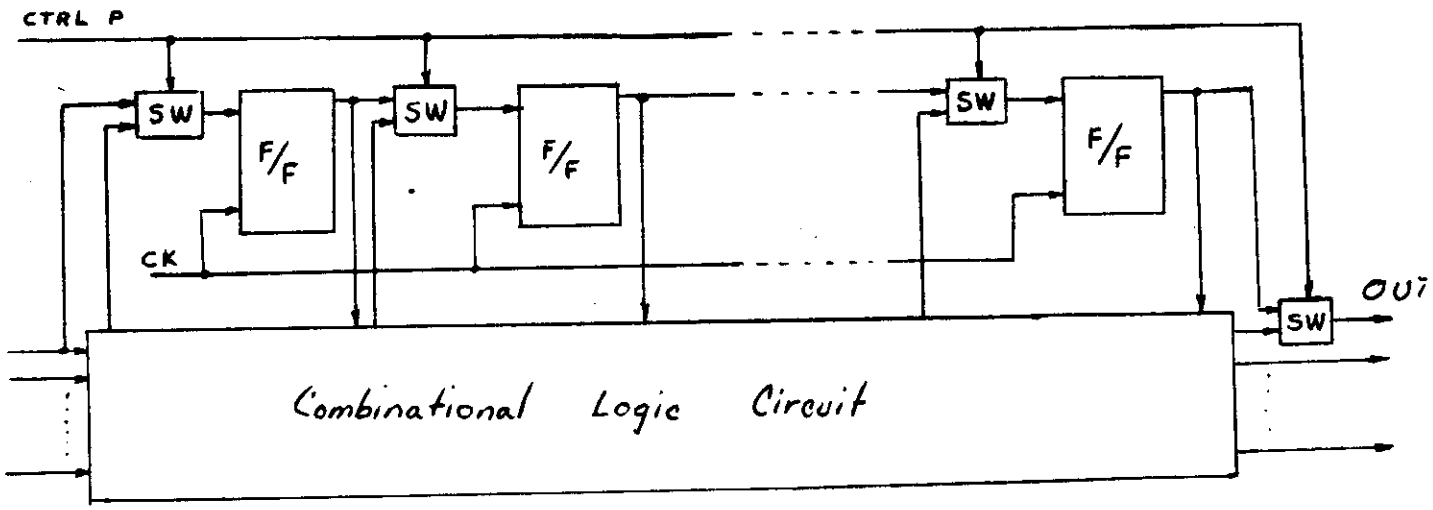
A 16-bit linear feedback shift register

In compact testing the CUT is usually fed pseudo-random input vectors. The output responses of the circuit are compressed and compared against a reference value.

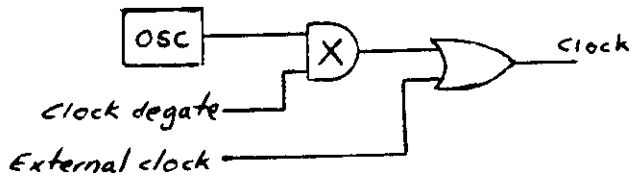
Combinational versus sequential Logic

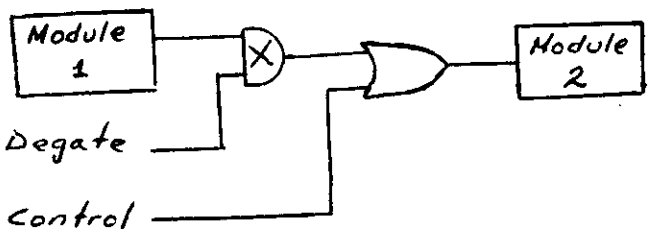
A combinational circuit is divided into smaller blocks and the internal states of the nodes of interest are loaded into a chain of Flip-Flops after the application of a certain test input pattern. The contents of the FFs, configured as a Shift-Register are read out.

Steps: 1 CLR F/F
2 Load Input Vector
3 Load Responses in FF
4 Read Out FF chain through OUT.



Dividing The circuit





Logical partitioning can be accomplished by adding extra logic. Degating is accomplished by signals Clock degate and Dogate, isolating different blocks from each other. Extra TEST inputs allow for a set of internal degating signals

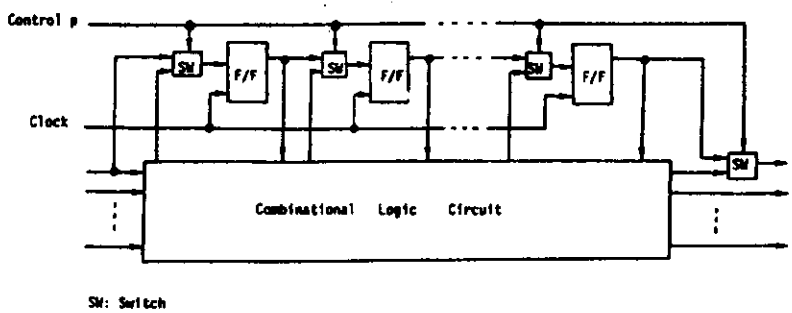


Figure 5.2
Shift-register modification (from Williams and Angell 1973; © 1973 IEEE)

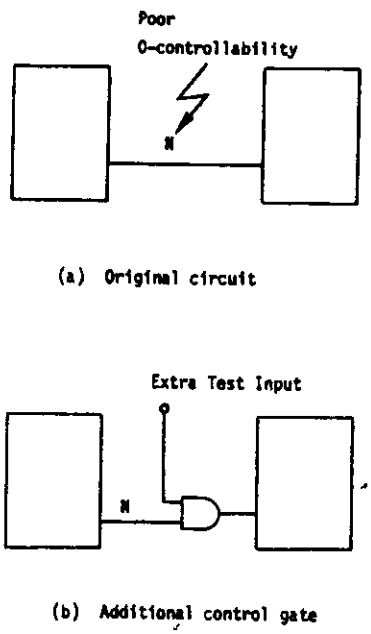


Figure 5.5
Test point for enhancing controllability

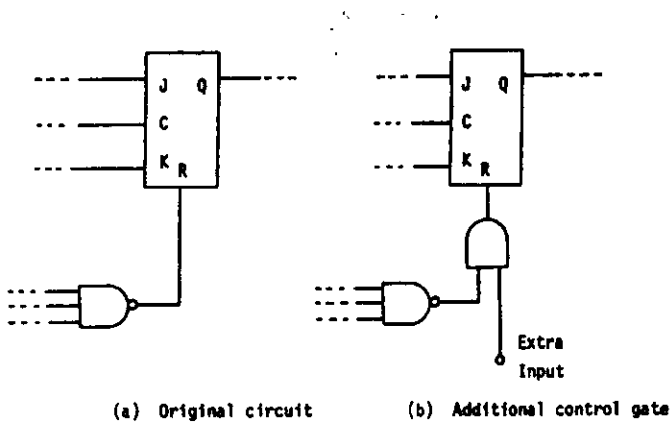


Figure 5.6
Test point for direct resetting

Functional Testing

Let's consider a hypothetical example of a VLSI we designed. This design, comprising a modest 2000-gate count has 10 inputs and 14 outputs. The question now arises: What do we want to test for? If we are circuit designers it's of little or no use at all to test for stuck-at faults in the circuit, due to Technological problems, i.e. open lines, nodes stuck-at-0 or stuck-at-1, etc. We are deeply concerned with a main issue: Is the design working properly? Is it cappable of accomplishing all the functions it was designed for? So, we must look for functions, by means of the so-called functional testing. It's important to achieve, also, exhaustivity, meaning that all the possible functions have been tested and thus our design is working properly beyond all reasonable doubt.
The generation of test vectors can be carried out by one of these ways:

- By hand, that is to say manually, which will proof to be a cumbersome and almost impossible task in the great majority of cases.
- By means of an Automatic Test Vector Generator (ATVG)

A generous dose of simulation must be added to your design before fabrication.
No time is lost in or with simulation, it's lost afterwards when you find your design doesn't function properly, after there's no solution since it's already encapsulated and paid for.
There's an old saying stating that "Time is money". The more so in these cases, VLSI is not an exception to the rule!
Good CAD tools can carry out good simulation for you, with reasonably reliable results. I mean you can trust that your design will behave itself in "real life" as it did in simulation.
You must, though, keep in mind some basic rules.

Testing

If we admit we are already testing our design it simply means we are close, very close already, to verify the proper functioning or - in the worst case - to be deeply deceived noting that the circuit doesn't perform as we expected.

During design, when simulating, you should keep in mind some considerations which will help you to extend your simulation results to real life.

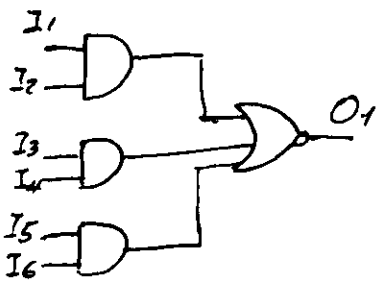
1: Keep in mind that the manufacturer you have chosen as a partner in this venture is your friend, not your foe! He'll try to help you within reasonable limits, but he also needs your cooperation in achieving a testable circuit. Usually he'll demand a node coverage greater than 95% or 97% during simulation (nodes exercised at least once during simulation)

2: Don't you ever forget that the ATE (Automatic Test Equipment) your manufacturer uses runs with a clock frequency which may be considerably lower than the frequency at which you want your circuit to function. For instance: ATE has a clock frequency of 1 MHz, while you expect your circuit to function properly at 25 MHz. It's up to you to design your circuit with enough safety limits, but be aware that the ATE certifies GO-NOGO at it's clock frequency.

3: Never fall into the subtle trap of "My design is good, therefore it will function OK, since I don't make mistakes"

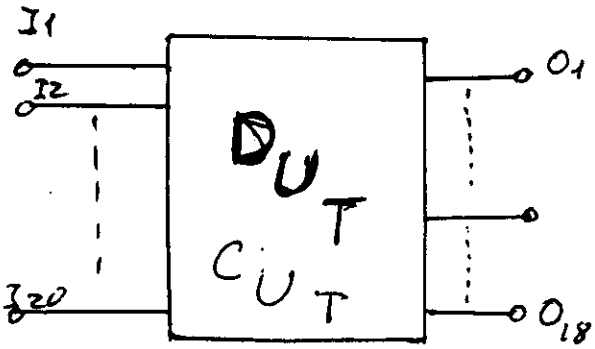
Exhaustivity and Cost

A simple AOI (And-Or-Invert) gate is not difficult to test; It only demands a simple set of test vectors to be used. Thus testing an AOI doesn't represent any special problem at all.



If we suppose that our Automatic Test Equipment (ATE) has a clock frequency of 1 MHz, Then checking output O_1, as you change the states of inputs $I_1 \ldots I_6$ to cover for the 2^6 possible combinations, takes a short time. The cost of testing will be small.

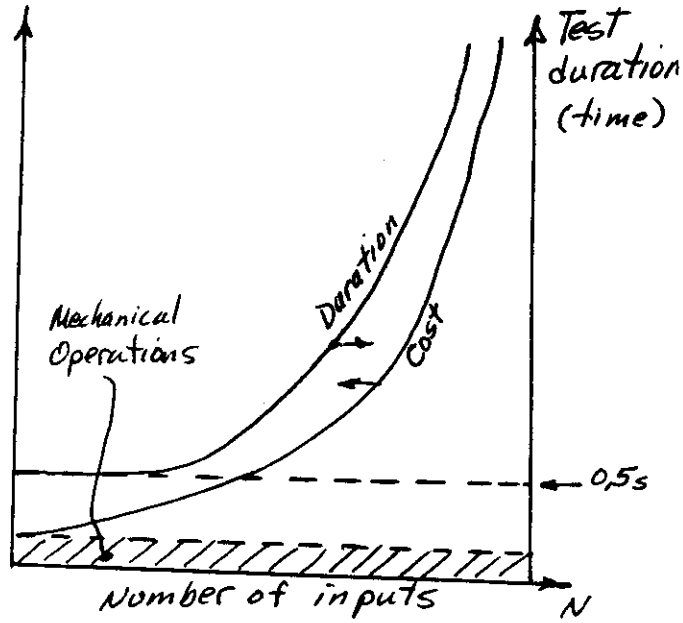
Now, with this in mind let's try to figure out what should happen if my design is the one depicted below, with 20 inputs and 18 outputs.



Your (or my) design is now, or is going to be, a DUT (Device Under Test) or CUT (Circuit Under Test). If you (or I), lets say We, try to carry on an exhaustive test, We'll need 2^{20} combinations of input vector if all inputs are independent among each other. This will take more than 10^6 test vectors, which is not quite acceptable for your manufacturer. If you insist

on doing it so he might be "convinced", charging you more money for testing.

Exhaustivity and cost are deeply related. Below you can see a relation between both of them.



For small values of N (inputs) Test Cost is relatively independent of N. This is because the testing time (electronically speaking) is much less than the time due to mechanical operations in the ATE.

For intermediate values of N, Test Cost starts to influence the total test Cost. Then, as N further increases test duration climbs up making testing unaffordable.

As an illustration, let's try to quantify Test Cost. For this honorable purpose we will make the following assumptions:

1: Testing time due to mechanical operations is constant and equal to 0,5s.
2: Depreciation of ATE is 25% per year, and the total cost of ATE is $500K, Thus we get a depreciation contribution of $4,0187754 \times 10^{-3}$ $/s
3: Payroll (Salaries) $10K per month which gives a contribution of $3,8580243 \times 10^{-3}$ $/s
4: Energy, maintenance etc $10K per month and $3,8580243 \times 10^{-3}$ $/s

Testing cost and time per chip

N	$t_{mechanical}$ (s)	$t_{meas.}$ (s)	t_{total} (s)	t(total)	Cost $	Remarks
5	0,5	very small	≈ 0,5s	0,5s	0,006	–
10	0,5	0,005	≈ 0,5s	0,5s	0,006	–
15	0,5	0,164	0,664	0,664 s	0,008	–
20	0,5	5,243	5,524	5,524 s	0,065	–
25	0,5	167,8	168	2,8 min	1,97	Oh! Oh!
30	0,5	5368,7	5369	1hr 39min	63,00	Very Bad!
35	0,5	171798,8	171799	47,72hr	2016,00	Nuts!
40	0,5	5497560	5497561	1527h	64512	Out of Business

Of course, you can note that $ 63 fo measuring (testing) one chip is unbearable. From there on you lost your battle, you're out!

Fortunately enough a 40-input chip is <u>never</u> a combinational chip exclusively. Anyhow, you won't have 2^{40} possible combinations, but will have other situations to cope with.

Well, now go on!.

A final note on this topic:

How would you think about calling this testing approach The Brute Force Method?

Partitioning

Partitioning a complex circuit in smaller and less complex blocks can be accomplished via multiplexers with their corresponding control signals. Subcircuits can be bypassed as required.

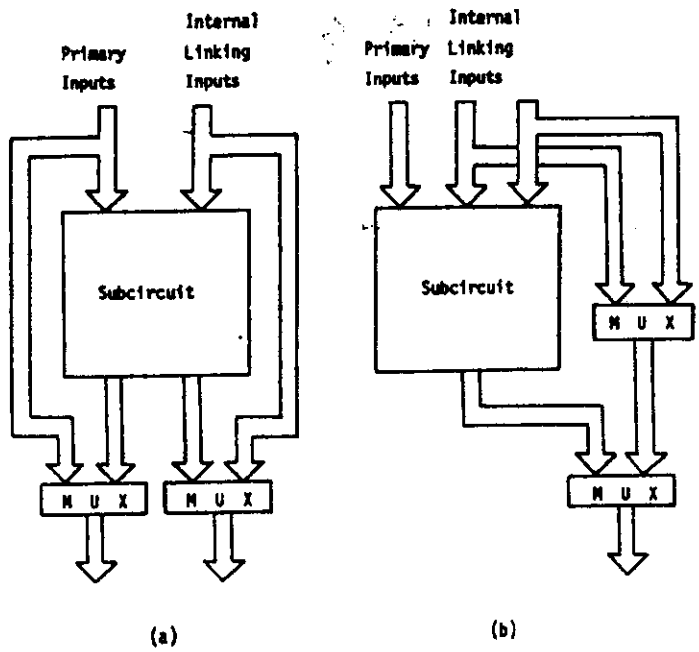


Figure 7.1
Bypassing subcircuits (Sakauchi et al. 1975; © AFIPS 1975)

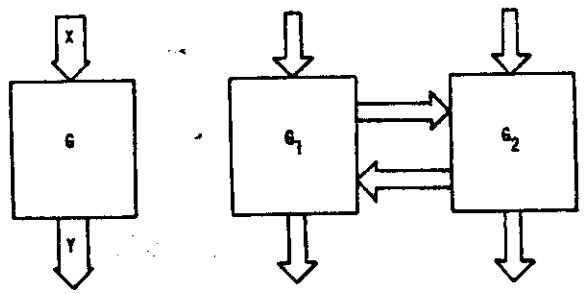


Figure 7.2
Partitioning (Bozorgui-Nesbat and McCluskey 1980; © IEEE 1980)

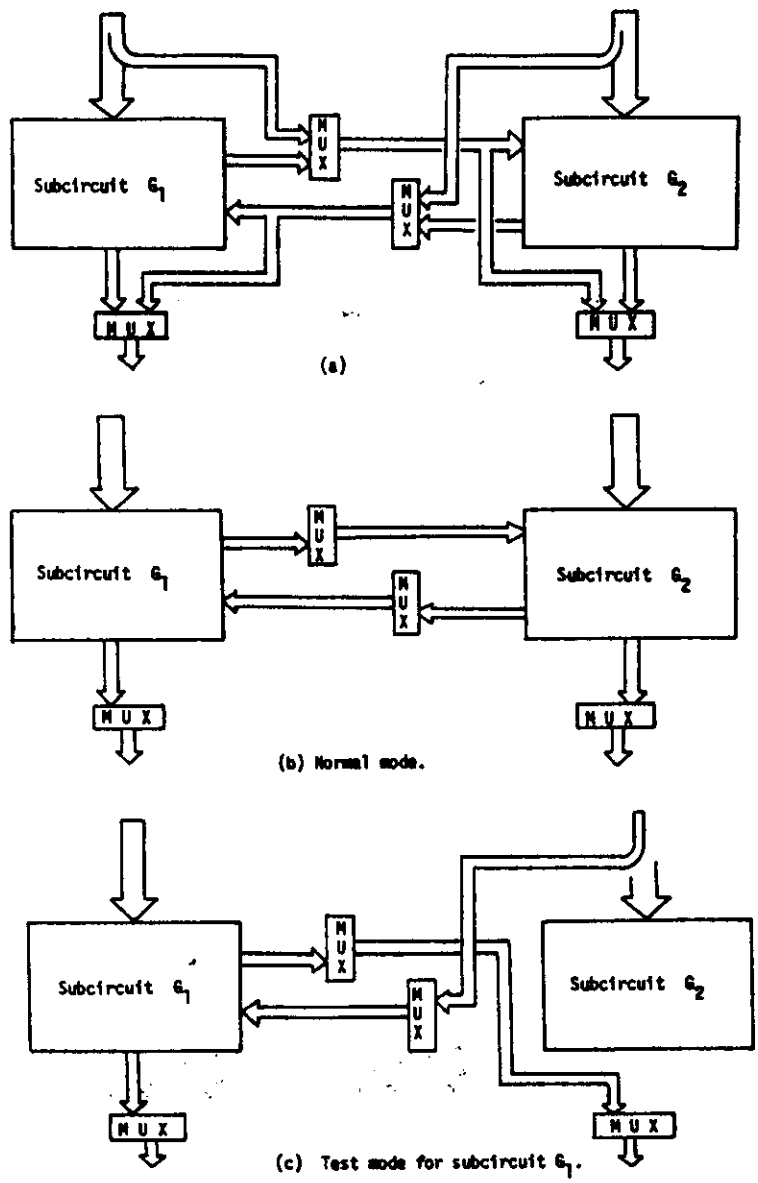


Figure 7.3
Partitioning scheme using multiplexers (Bozorgui-Nesbat and McCluskey 1980; © IEEE 1980)

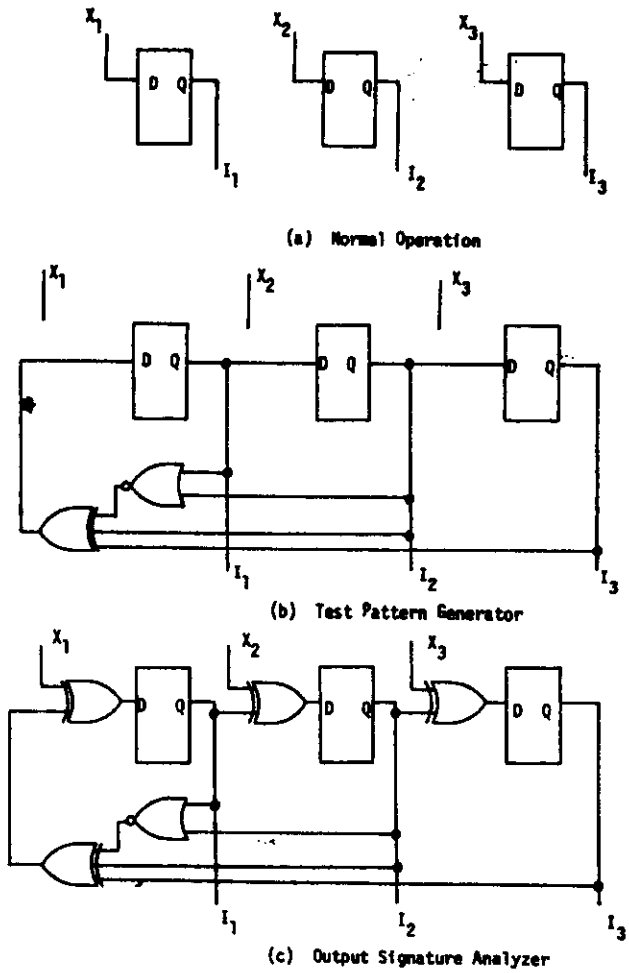


Figure 7.6
Various modes of 3-bit LFSR module (McCluskey and Bozorgui-Nesbat 1981; © IEEE 1981)

Packaging and Sockets.

If one looks around into the world one lives in there's something that calls for attention: a great percentage of manufactured products are packaged, in some way or another. Just imagine having a beer without a package (a bottle or metal can in this case).
Now let's think about VLSIs: you can consider that the last stage in the manufacturing process is, precisely, packaging.
Unencapsulated or let's say "naked" chips are useless for later application if you try to use them that way.
So, packaging deserves some attention, even when you think "Well, anyhow, I am not doing the packaging myself. Since someone is doing it for me I don't need to worry about it".
True that you don't carry out the packaging yourself, but you must be aware that some decisions have to be taken in this, the final of a long series of steps.

Remember that, usually, your design is not only a part of a system but maintains a very strong interaction with it. Including, of course, the printed circuit design, lead positioning, expected temperature rise and so on.
Packages play a most important role, since they must comply with a set of conditions, both present (depending on the characteristics of your chip, and future, depending on HOW are you going to "use" your packaged chip in your design (i.e.: mounting, soldering, etc.) and on COST.

Some words about packages

Packaging may be arbitrarily divided (perhaps not so arbitrarily) into two classes:

1- Packaging for High-End Computing Systems and
2- Packaging for Low-End Work Station, Peripherals and Consumer Electronics.

Packaging for High-End Computing Systems is characterized by the use of proprietary, high-capacity carriers.
Examples of this are IBM's TCM (Thermal Conductor Module) which may contain nearly 100 chips, bonded to a 90-mm-square and 5,5-mm-thick multilayer ceramic substrate (MLC). This MLC is a 33-layer substrate made of 90% Al_2O_3 and 10% glass dielectric layers, where Molybdenum is used for interconnection wiring. One of these modules may contain 130 m of x-y wiring and 350 000 vias for interlayer connections. Finally, 1800 module pins brazed to the bottom of the MLC are used to plug it onto the mother PCB's. One of these mother PCB's is 600 x 700 mm and comprises 20 layers of epoxy-glass with copper wiring and 40 000 plated through holes for plugging-in the TCM module pins
Other packages, like Hitachi's multichip one hold several chips inside the package.

Packaging for Low-End Applications

In contrast to the high-end computer case, in this Low-End area, cost, I/O pin count and space requirements play major roles in packaging.

Two mounting technologies are in common use:

- The traditional, and perhaps quasi-obsolete, pin-through-hole (PTH) technology.
- The once emerging, and now well-established, surface-mount-technology (SMT).

For the VLSI designer packaging may be oversimplified in answering the following questions:

- What kind of mounting technology will be used?
- What kind of package is to be chosen?
- What kind of material (for the package) is best?
- What I/O count is the best?

Mounting Technology:

It must be quite clear that having a packaged VLSI you can not mount on a PCB board is not the goal to achieve. So, a package must be chosen that is either PTH or SMT and can be mounted on the PCB you have access to produce. No need further arguing about this topic.

Packaging

Some important aspects to consider are the following:

Number of pins

Type of package:

- **Surface Mount**
- **Through Hole**

Material:

- **Plastic**
- **Alumina (Aluminum Oxide)**
- **Beryllia (Beryllium Oxide)**

Power Dissipation and Chip Temperature:

$$T_j = T_a + \Theta_{ja} * P$$

where T_j = Junction temperature
T_a = Ambient temperature
Θ_{ja} = Thermal resistance from junction to ambient
P = Power dissipation

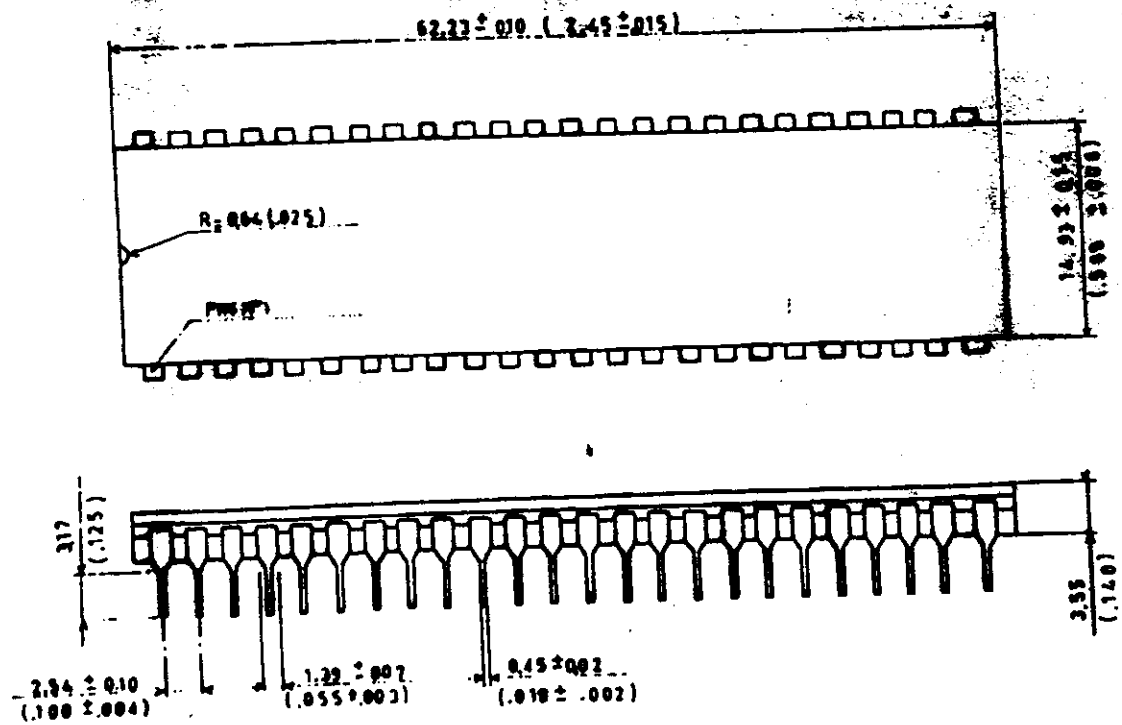
Types of Packages

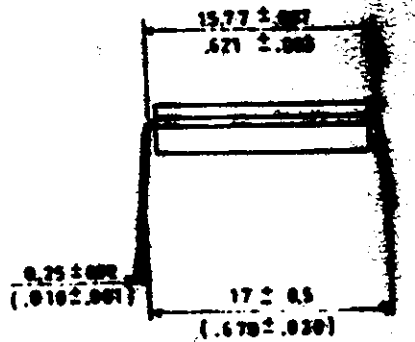
In the following table the main types of single-chip carriers and their characteristics have been included.

Table 1. Characteristics of single-chip carriers

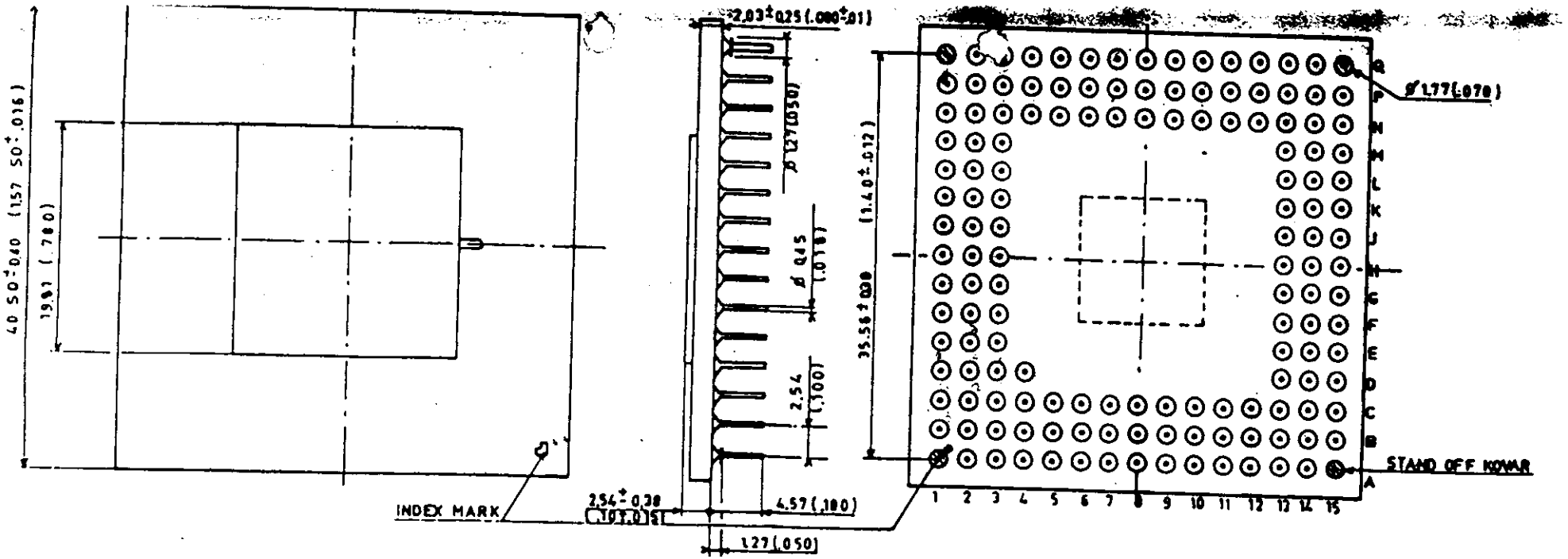
Carrier type	I/O (max)	PCB space for Pins: in cm^2	Materials
Pin-through-hole (PTH)			
DIP	64	7,74	Plastics and Ceramics
PGA	300	2,8	Plastics and Ceramics
Surface-mount technology (SMT)			
SOP	40	3,9	Plastics
FPP	120	2,4	Plastics
LCC	132	1,9	Ceramics
PLCC	68	2,0	Plastics
JLCC	84	2,8	Ceramic

DIP : Dual-In-line Package
PGA : Pin Grid Array
SOP : Small Outline Package
FPP : Flat Plastic Package
LCC : Leadless Chip Carrier
PLCC : Plastic Leaded Chip Carrier
JLCC : J-Leaded Chip Carrier





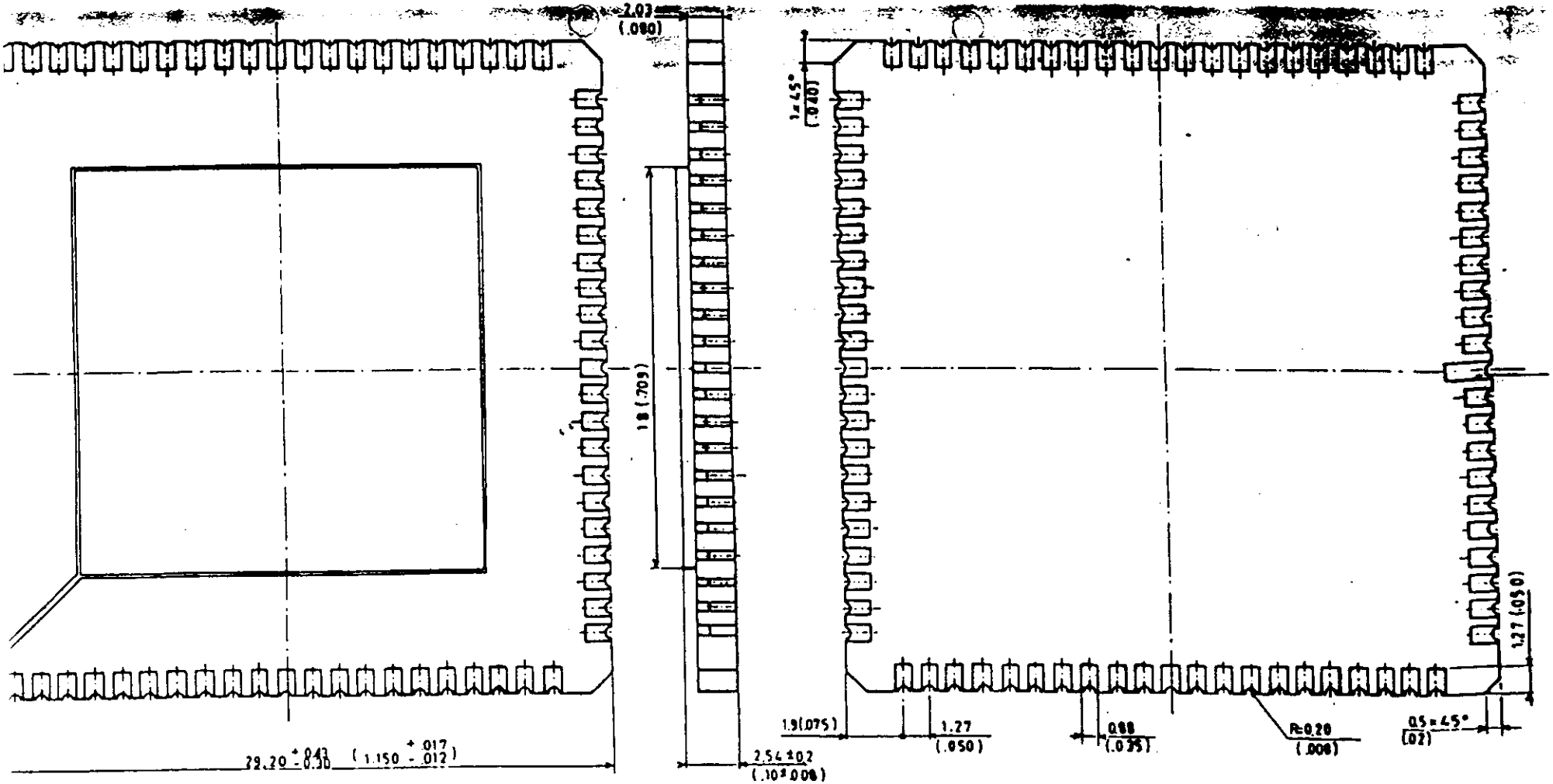
48 LEAD CERDIP



PAD NO	PIN NO	PAD NO	PIN NO	PAD NO	PIN NO	PAD NO	PIN NO	PAD NO	PIN NO	PAD NO	PIN NO
1	D3	25	L1	49	N7	73	M13	97	E15	121	C9
2	C2	26	L2	50	Q6	74	N14	98	E14	122	A10
3	B1	27	M1	51	Q7	75	P15	99	D15	123	A9
4	D2	28	N1	52	P8	76	M14	100	C15	124	B8
5	E3	29	M2	53	Q8	77	L13	101	D14	125	A8
6	C1	30	L3	54	N8	78	N15	102	E13	126	C8
7	E2	31	N2	55	N9	79	L14	103	C14	127	C7
8	D1	32	P1	56	Q9	80	M15	104	B15	128	A7
9	F3	33	M3	57	Q10	81	K13	105	D13	129	A6
10	F2	34	N3	58	P9	82	K14	106	C13	130	B7
11	E1	35	P2	59	P10	83	L15	107	B14	131	B6
12	G2	36	Q1	60	N10	84	J14	108	A15	132	C6
13	G3	37	N4	61	Q11	85	J13	109	C12	133	A5
14	F1	38	P3	62	P11	86	K15	110	B13	134	B5
15	G1	39	Q2	63	Q12	87	J15	111	A14	135	A4
16	H2	40	P4	64	Q13	88	H14	112	B12	136	A3
17	H1	41	N5	65	P12	89	H15	113	C11	137	B4
18	H3	42	Q3	66	N11	90	H13	114	A13	138	C5
19	J3	43	P5	67	P13	91	G13	115	B11	139	B3
20	J1	44	Q4	68	Q14	92	G15	116	A12	140	A2
21	K1	45	N6	69	N12	93	F15	117	C10	141	C4
22	J2	46	P6	70	N13	94	G14	118	B10	142	C3
23	K2	47	Q5	71	P14	95	F14	119	A11	143	B2
24	K3	48	P7	72	Q15	96	E16	120	B9	144	A1

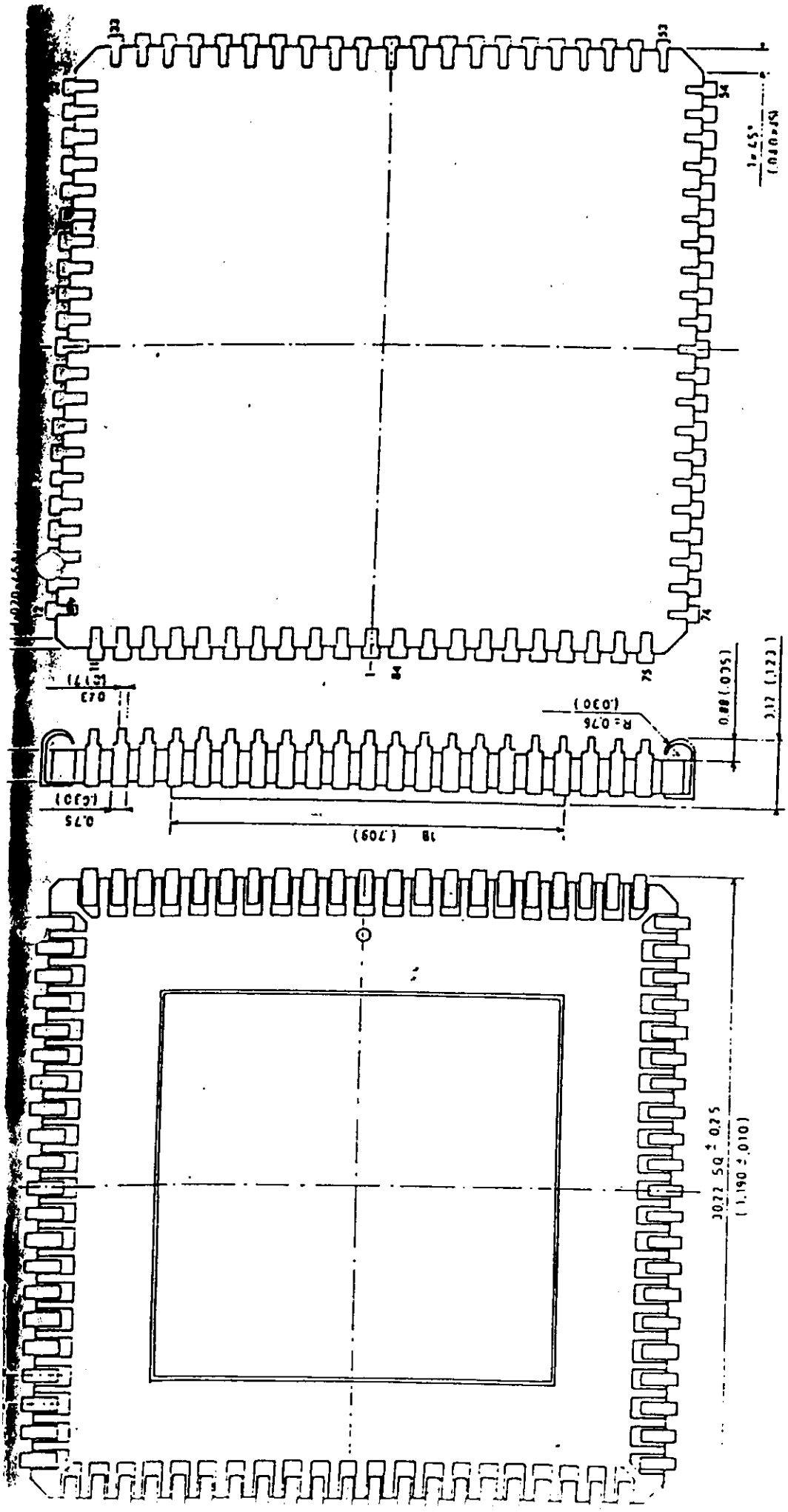
144 PIN GRID ARRAY PACKAGE

European Silicon Structures



84 LEAD CHIP CARRIER

LEADLESS TYPE



84 LEAD CHIP CARRIER
LEADED TYPE

Some facts on dimensions of packages

Perhaps you may think in a package of a certain type, let's say, a 28-pin DIP for your design, which needs only 20 I/O pins. Let's suppose your chip is going to be 5,8 mm square. So you might think that the pad size in the package (the cavity where the chip must fit in) could be 5,84 x 5,84 mm. Then you'll find that your manufacturer won't accept this size, claiming that there's not enough room for your chip. How come?

You have forgotten certain important facts to be considered:

- Your chip has to be diced -or cut- from a wafer and will grow in size with the rest of the scribe lines in the wafer. So you must take this into account, perhaps 50 µm per side.
- The bonding pads in your chip should be 100x100 µm minimum. Separated from each other a minimum of 200 µm (plastic) or 150 µm (ceramic) from outer to outer of pads.
- Wire (used in bonding) should not exceed 3 mm in total length as measured in a horizontal (top) plane.
- Wire angle should be 45 degrees minimum (from edge of chip to wire)
- Wire cannot cross (overlap) more than 500 µm of die surface

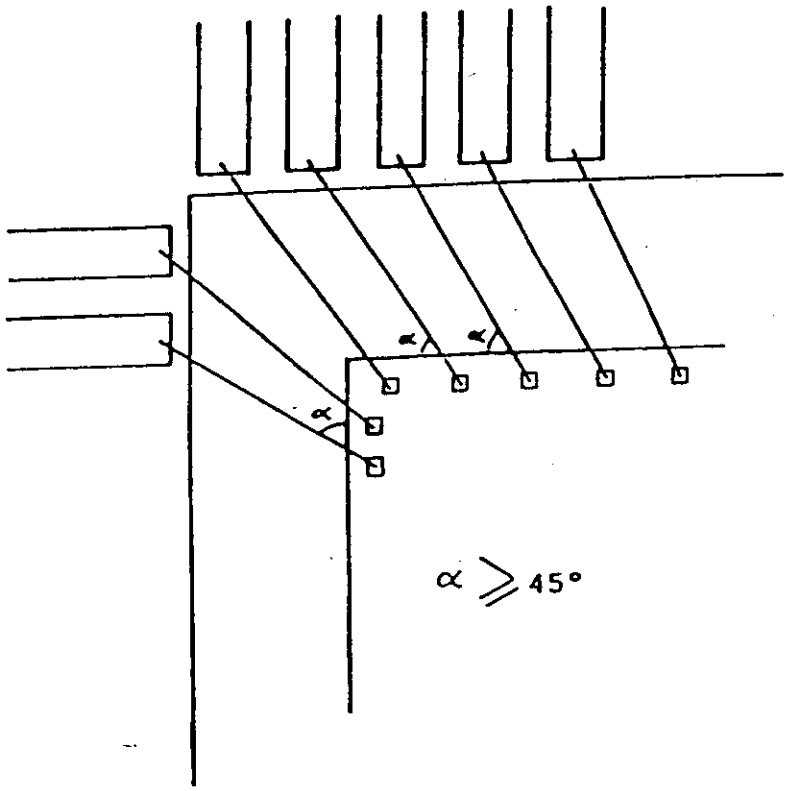
As you can see, these rules are also related to the package and to your chip.

Plastic dual in line packages (suite)

Pin Count	Package width (inch)	Pad size (mm)	Min.die (mm)	Max die (mm)
22	. 300	3.55x4.32 5.59x5.84	1.02x1.78	4.57x4.83
24	. 300	4.06x8.38	1.52x5.84	3.05x7.37
24	. 600	3.05x3.81 3.81x3.81 4.57x5.59 5.59x6.10 5.84x5.54 6.35x6.86	1.02x1.27	5.33x5.84
28	. 600	3.81x3.81 5.08x5.08 5.84x5.84 6.60x6.60	1.27x1.27	5.59x5.59
40	. 600	5.08x5.08 5.33x5.84 6.60x6.76 7.62x8.64	2.54x2.54	6.60x7.62
48	. 600	6.60x6.60 7.11x7.11 7.87x7.87	4.06x4.06	6.86x6.86

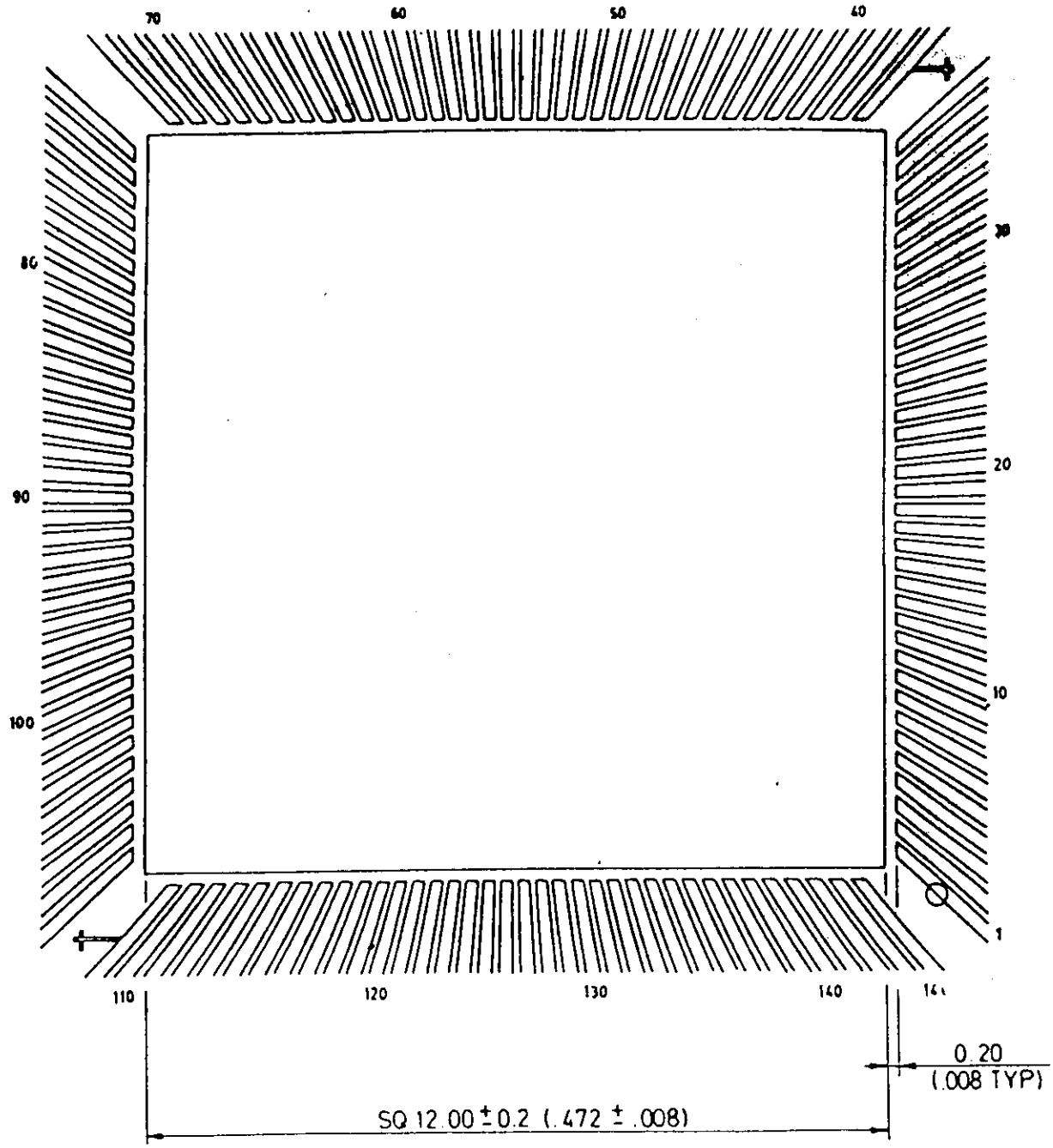
I.2.2. Plastic leaded chip carriers

Pin Count	Package width (inch)	Pad size (mm)	Min.die (mm)	Max die (mm)
28	. 490	5.08x 5.08 6.60x 6.60	1.27x1.27	5.59x 5.59
44	. 690	5.84x 5.84 7.62x 7.62	2.03x2.03	6.60x 6.60
68	. 990	7.62x 7.62 10.41x10.41	3.81x3.81	9.40x 9.40
84	. 1190	9.14x 9.14 10.79x10.79	5.33x5.33	10.03x10.03



MINIMUM WIRE ANGLE

Fig.2



BONDING DIAGRAM
144 PLASTIC PIN GRID ARRAY

European Silicon

Materials used in packages and sockets

Different materials are used in packages, on account of their physical and chemical characteristics, namely Alumina (Al_2O_3), Beryllia (BeO), Silicon carbide (SiC), Silicon nitride (Si_3N_4), Aluminium nitride (AlN) Quartz glass (SiO_2), Epoxy glass, Polyimide.
Of these the first two are used for ceramic packages and different plastics, usually epoxy-based are employed for plastic ones.
Sockets, about which we'll talk a little in a few moments, are usually plastic and glass or alumina filled for increasing the quality of their thermal and mechanical properties.
Alumina and Beryllia offer good insulating characteristics and good thermal conduction properties, the latter being better than Alumina and thus more expensive.

Thermal conduction - and, therefore, thermal resistance depend on the geometry (dimensions) of the package and the material it is made of.
The heat developed inside the chip bulk must be carried away, via the package and the surroundings, in order to guarantee safe operating margins and long life (and reliability) for the chip.

Thermal resistance helps in illustrating the package characteristics regarding heat dissipation.
Remember that:

$$T_j = T_a + \Theta_{ja} * P$$

where T_j, T_a = Temp. (junction, air)
Θ_{ja} = Thermal resistance
P = Dissipated Power.

Plastic leaded chip carriers

Pin Count	Thermal Resist OjA (oC/W)	Comments
28	68	Frame material : Copper
44	58	Die size : 120 x 120
68	42	" "
84	33	" "

Smalll outline packages

Pin Count	Thermal Resist OjA (oC/W)	Comments
16	120	Lead frame material : Copper
20	90	Die size : 100 x 100
24	80	" "
28	70	" "

Note : These values are given for the die size shown in comments. For larger die, values are typically lower.

As can be seen in the previous table, thermal resistance values can be as high as 120 °C/W for a plastic 16-pin SOP or as low as 33°C/W for an 84-pin PLCC.
Ceramic packages have values of thermal resistances ranging from 70 °C/W for a 14-pin ceramic DIP to 35 °C/W for a 144-pin ceramic PGA.
The values shown are for a specific die size and should not be taken as absolute values.
Smaller values may be attained employing external cooling means (forced ventilation, radiating fins).

I/O Count : Which one is the best

Savings in I/O pins may result in later trouble.

Remember, anyhow, that in VLSI there's no harm done in:

1. Using more than one pin for VDD.
   Several pins will make routing easier inside the chip and can be paralleled when you deal with your PCB
2. The same advice from the previous paragraph is also valid for GROUNDS. Use more than one GND pin whenever possible.
3. Use independent ANALOG AND DIGITAL GNDs whenever you are designing a mixed VLSI, that is, a digital plus analogue VLSI.

Answering the question: "Which I/O count is the best for my design?" is not straightforward.
We must now resort to a fact: You've already decided the total number of I/Os on your chip, considering even multifunction pins and so on.
So the pin count for your package shouldn't be an extremely stringent issue. You must fit ALL of your I/O pins in the package you choose and there's no harm at all leaving unconnected pins in it.
Do remember that you may be packaging prototypes and maybe, afterwards, you'll decide to have a production lot from your design.
A nice (and economic) practical rule is:

"Choose ceramic carriers for prototyping and plastic carriers for production"

The cost ceramic carriers add to development is paid by a fast delivery of the final product to be tested in your application.
Have in mind that sometime you'll need to switch packages from ceramic to plastic. Make sure your die fits in the new plastic package.
No doubt you'll have the support of the manufacturer you choose. Consult him and get his rules and advice.

Sockets

Now, after some lullabies and some fuss regarding packages, pin count and so on and after having talked about mounting technologies (PTH and SMT) for PCB there is the need to talk a little bit about sockets.

Why sockets?

Perhaps because Reliability exists and all components, including VLSI circuits, are prone to fail or to misfunction sometime in their lives and then you (or someone else, with less tools, means and knowledge than you) are faced with the job of removing your VLSI and inserting in place a new one.

You may argue: "What are soldering irons made for?" "What are the special desoldering tools made for?" This reasoning, no doubt, points out to the fact that you yourself have probably removed a bad component from a PCB boasting that you desoldered a 14-pin DIP without no harm to the PCB. Generally speaking I figure that desoldering difficulty increases as the power of the number of pins. So trying to make this feeling a little heuristic it could be stated as

$$DD = K_1 \cdot K_2^{NP} + K_M^{NP}$$

where DD: Desoldering Difficulty, K_M: Murphy's Law Coefficient
K_1, K_2: Constants
NP: Number of Pins.

If you figure out that you should implement, besides all your functions in your VLSI, an easy maintenance, then you might be convinced of the need for sockets.

Of course, the socket must be thought about as another component to be mounted on your PCB, but mounted only once, and never desoldered again. Sure it adds cost to your system design -thinking about your application-, so this cost must be weighed in the overall system development cost.

If your PCB, let's say, will sell for $19,99 it isn't worthwhile considering a socket that will add some 3 or 4 extra dollars to your production cost.

Anyhow, returning to desoldering: What about if you think on a PC Mother Board with its huge VLSIs in chip carriers mounted in sockets? Guess what would happen if they were soldered directly on your Mother Board!

Well, now after this talk about the need for sockets let's quit the ballyhoo and call your attention to some simple, even naive, but nevertheless important hints regarding sockets.

1: Sockets are usually plastic, some may be glass-filled or have some compound to add-on mechanical characteristics. The material may be flame-retardant, which adds an additional safety characteristic.

2: Pins must be of such materials as to guarantee good electrical contact on all pins and good mechanical contact with a reasonable pressure. Phosphor-bronze with a nickel and later gold coating serve these purposes.

3- Be sure your carrier - The one chosen by you - will fit the available socket. You might be surprised to see how difficult you could find the task of getting the right socket for some high-pin-count carriers.

4- "Choose the socket, don't let the socket choose you" could be the final advice.