Fundamentals of Integrated Circuit Design

1. Definitions

Integrated circuits

An integrated circuit (IC) is formed by components and interconnections that are fabricated on a single silicon piece of semiconductor, typically indicated as "chip" or "die". An integrated circuit can be divided into several sub-circuits, performing different functions, typically indicated as "cells". A whole chip can also be regarded as a cell (top-cell).

Each cell includes different views (cell-views), consisting in different ways to represent the cell. Different cell-views are used at the various steps of the design flow. We will examine this concept later. Now, let us focus on two different cell-views:

1) Schematic cell-view

2) Layout cell-view

The schematic view is the schematic diagram of the electrical circuit, representing the cell as a set of simpler parts, connected by means of their terminals.

The layout view is a geometrical representation of the cell. The object is contained in a 2-dimensional region (cell-outline), which define the substrate area occupied by the cell. The layout is a collection of 2-D shapes (e.g. rectangles), which tell the factory (called silicon foundry) the exact areas, within the cell outline, where the various process steps have to be carried out. Each shape represents an object that will be fabricated on the substrate. The particular type of object (e.g. polysilicon, metal, doping implantation) is represented by a property of the shape, which is called "layer".

Figure 1 shows a possible schematic and layout view for a CMOS nand gate. The available layers in a CMOS process are collected in the so called "layer palette". Note that modern IC fabrication processes include a much greater number of layers. The layer set is completely defined by the process and the designer has no control on it. The layout designer decides where the various layers (roughly corresponding to process steps) have to be applied. For example, referring to Fig.1, the areas that will be covered by polysilicon in the fabricated cell are shown in Fig.2. Note that, to improve readability, the layout editor shows different color shades where two or more layers cross each other.

The actual project of an integrated circuit generally uses a much wider set of views. For example, complex logical cells are often represented by means of hardware description languages (such as VHDL and Verilog). Therefore, the views "vhdl" or "verilog" may be necessary. We will introduce another type of view, the "symbol" view, later in this document.

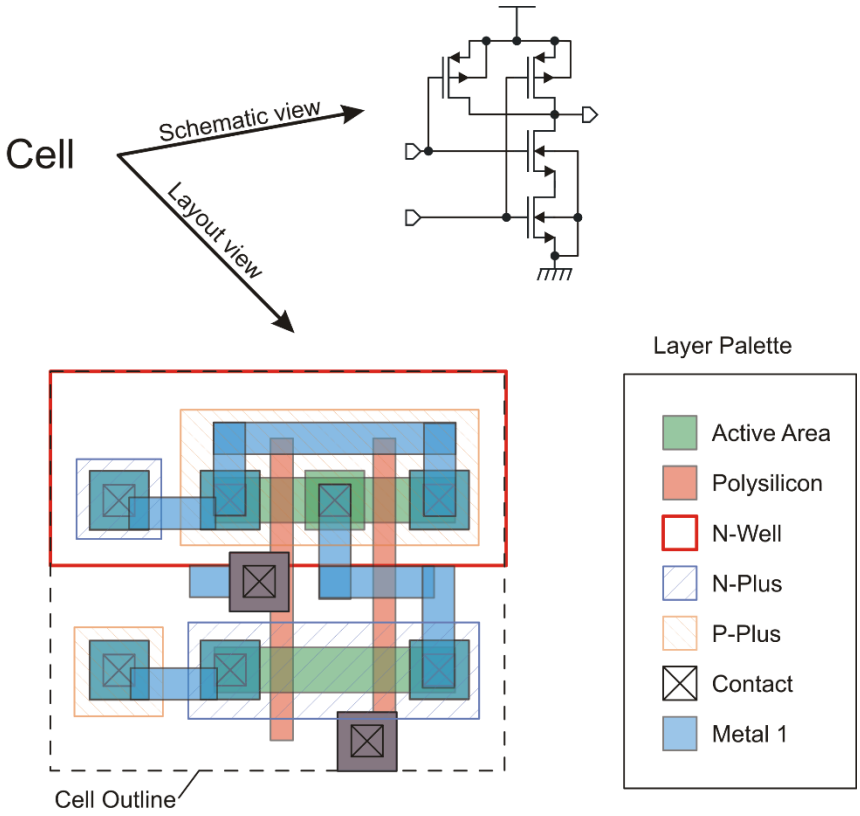

Fig.1 Schematic and layout views of a cell

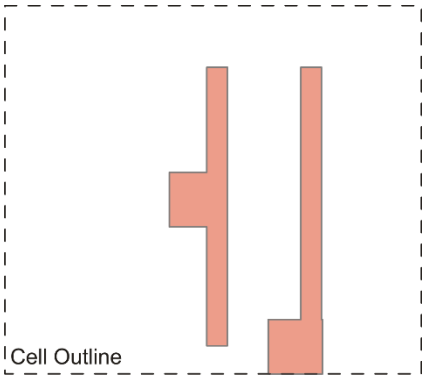

Fig.2. Areas covered by polysilicon in the cell to be fabricated from the layout of Fig.1

Design flows

A Design Flow is the collection of all steps that are necessary to design the IC, from the specifications to the layout. The three main design flows that are currently followed are:

- Analog Design Flow
- Digital Design Flow

- Mixed-Signal Design Flow

The various design flows with their typical products and dependencies are shown in Fig.3. The analog design flow is optimized to produce analog cells. Generally, at the core of the design flow there is the schematic view, which represents the electrical network (simpler parts, connected through nodes), generally indicated with "netlist". The design process is mainly manual, with very low standardization (the designer "style" still counts). On the other hand, the Digital Design Flow is founded on an HDL description, which is often behavioral (that is, not based on connection of actual components). The Digital design flow is used to produce complex digital architectures, such as finite-state machines, microprocessors, digital filters. The design flow from the HDL description to the layout is mostly automated. In many cases, the designer even does not need to inspect the final layout. Note that the Digital Design Flow requires the availability of a library of simple standard cells (logical gates, flip-flops, multiplexers etc.) which should be produced with the analog flow (see the figure).

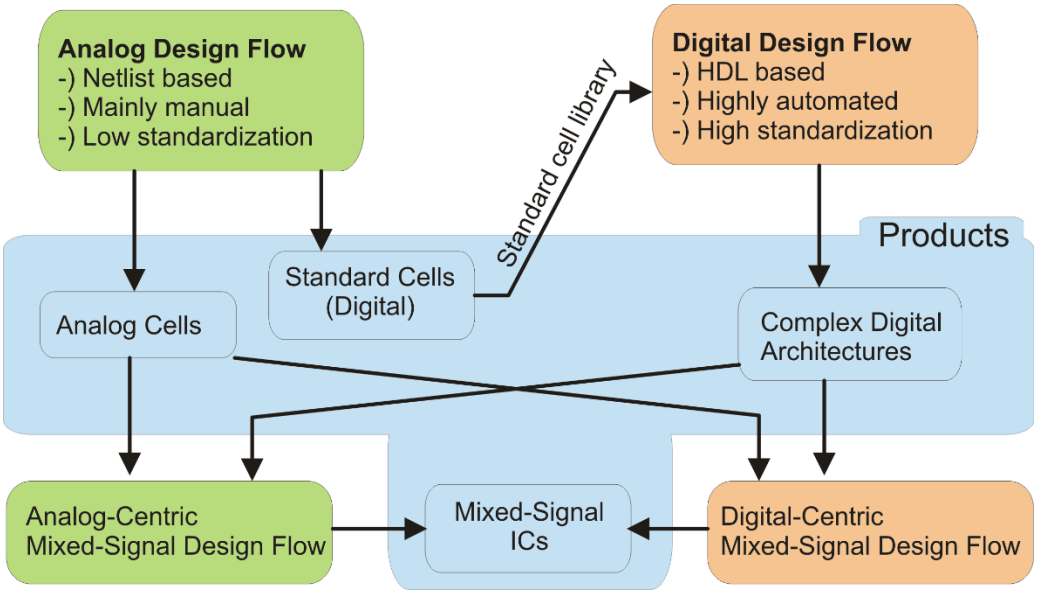

Fig.3 Integrated Circuit Design Flows

Mixed-Signal circuits, i.e. circuits including both digital and analog blocks, can be designed with either an "Analog-Centric" or "Digital Centric" flow. In both cases, the analog and digital cells are designed with their proper flows. In the Analog-Centric flow, the digital cells are combined with the analog ones using the same tools of the analog flow. In the Digital-Centric flow, the analog cells are combined with the digital ones using the Digital-flow approach. Generally, the analog-centric flow is preferable when the analog subsection is dominant and/or when the analog and digital cells cannot be simulated separately.

2. Analog Design Flow

A simplified diagram showing the various steps of the Analog Design Flow are shown in Fig.4. The figure is formed by three columns. The central column, enclosed into green boxes, includes the main design steps. The specific CAD tool, when applicable, are indicated in italic, between round parentheses.

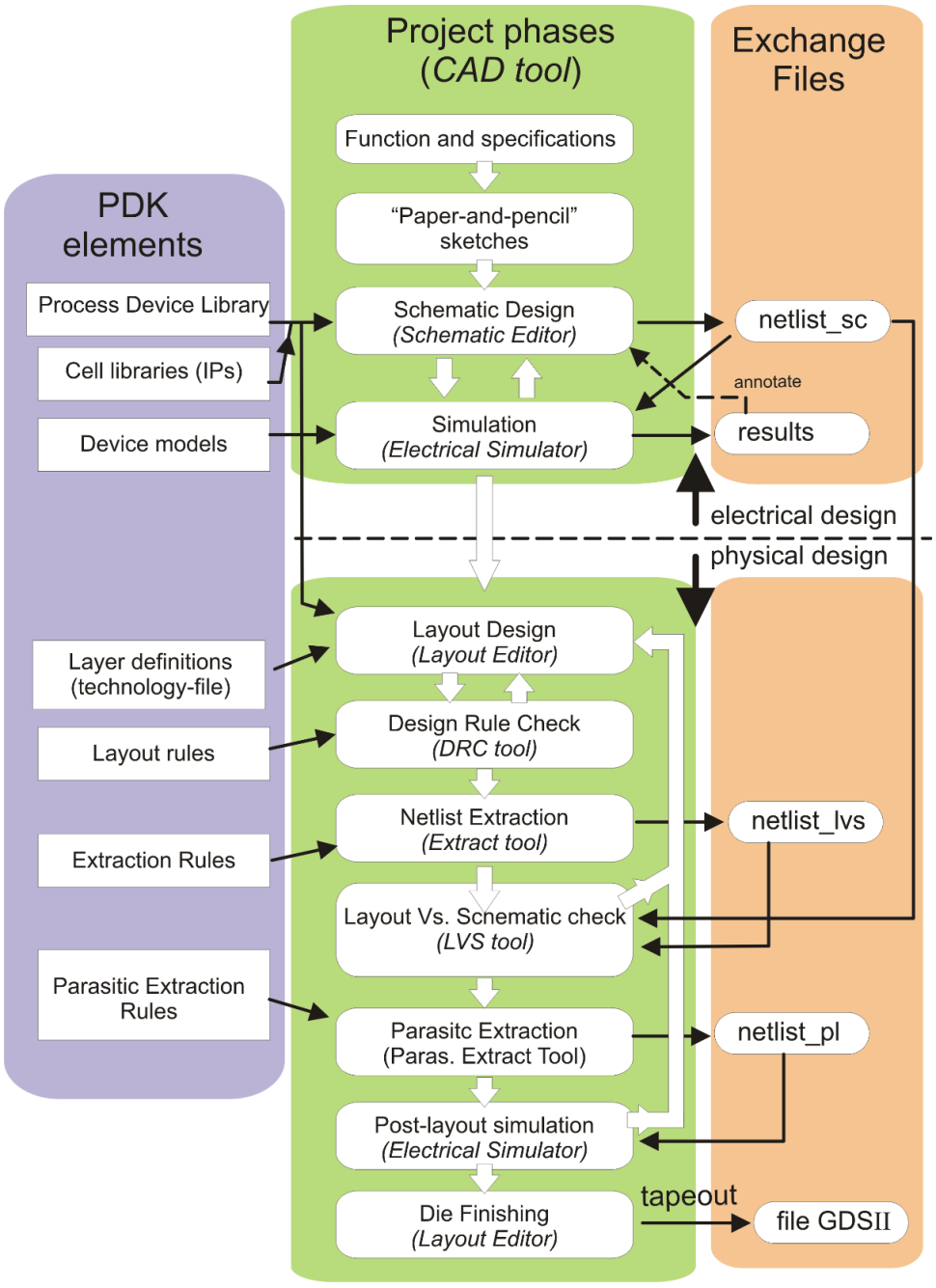

Fig.4. Analog Design Flow. The required PDK (Process Design Kit) files, the CAD tools and common exchange files are indicated

The CAD tools are generally available within a single CAD platform. The platform allows the various tools to be executed from the same user-interface and simplifies data exchange between the tools. A given CAD platform may include tools of different vendors.

The CAD platform should be customized with a series of files that contain information about the fabrication process. This collection of files forms the so-called Process Design Kit (PDK). A PDK includes files that are required at different stages of the project. The main PDK elements are collected inside the violet box in Fig.4. Most of the data included in the PDK are detailed in human readable form in the DRM (Design Rule Manual).

Finally, the orange boxes collect the main files that are generated and exchanged between the tools.

The process flow of Fig.4 is applicable to the design of a complete chip or of a single cell, with minor differences. The project is divided into the two main phases:

- Electrical (or Schematic) design. In this phase the schematic view of the circuit is developed.
- Physical design. This phase includes all steps necessary to design a layout that implements the schematic view designed in previous step.

At the beginning of the design flow, there is the description of the function that the circuit must implement and a series of specification that define the desired performances of the circuit. For example, we may have to design an operational amplifier (= function) with at least 1 MHz Gain-Bandwidth Product and a static gain greater than 80 db (= specifications).

The first phase of the design is the most creative and generally consists of drawing sketches by "pencil and paper" and making approximate, first order calculations. The designer uses his/her experience to find the best topology that meets the requirements. If a suitable topology is not available in textbooks and in scientific articles, the designer should try to develop an original topology, using transistor-level design skills. In this phase, it is very important to be able to perform simplified circuit analysis to obtain a rough estimate of the circuit performances and perform transistor sizing in order to obtain a first version of the circuit to start with. In this preliminary phase it is very useful to divide complex circuits into simpler ones (design partitioning).

The circuit is then drawn using a *schematic editor*. The designer will use the PDK device library, which includes all the devices that can be actually implemented with the chosen technological process. If available, the designer can use also macro-cells (e.g. operational amplifiers, comparators, etc.) that the foundry or other designers of the group have previously created. These cells are grouped into independent libraries and are generally referred to as "Intellectual Properties (IP).

When the circuit or part of it has been drawn, then a series of simulations are performed by means of an *electric simulator*, in order to estimate the circuit performances. Currently, several different electrical simulators are available, but all of them represent further developments of the SPICE (Simulation Program with Integrated Circuit Emphasis) program. In order to allow a circuit to be simulated, it is necessary to extract the netlist ("netlist_sc" in Fig.4), which is a textual file (ASCII characters) that lists all components and connections present on the circuit. The netlist should adhere to the typical SPICE format. For the simulations to be executed, it is necessary to provide the simulator with the so-called "device models". These files, which are part of the PDK, contain a set of parameters that are required to represent the behavior of all available devices. Device models generally include a very large number of parameters for each device, which are estimated by the foundry by means of a long series of delicate

measurements. The simulations produce data that can be plotted as graphs (e.g. voltage vs time plots), typically referred to as "waveforms" or simple collections of node voltages and branch currents, as in the case of the rest point. The waveforms are saved in binary files that are displayed with proper programs, which generally are part of the simulator package. Single data sets, such as rest-point data, can even be displayed directly in the schematic view, with an operation called "annotate", facilitating interpretation. Analysis of the simulation data gives information to the designer on the changes that should be applied in order to approach the target performances. Generally, a very large number of cycles between the schematic editor and the simulator are necessary to get the desired results. As the design gets refined, simulations become more and more sophisticated and start regarding the analysis of "manufacturability", that is the impact of process variations on the performances of the circuit. A good design should guarantee that most of the dies that will be fabricated satisfy the initial specifications, regardless of all possible variations and different operating conditions (supply voltage, temperature etc.).

When the final version of the schematic design is reached, then the physical design phase can be initiated. The physical design is carried out using the *layout editor* which is a graphical tool that allows insertion of shapes (rectangles, paths, polygons etc.) into a design area, where each point of the plane (i.e. substrate surface) is represented by means of orthogonal coordinates, typically expressed in microns. For the layout to be consistent with the chosen process, it is strictly necessary to use only the layer-set provided by the foundry within the PDK. The layer definitions are included in a file called "technology file", or "tech-file". Depending on the available tools and type of PDK, the layout creation is widely assisted by automated procedures. Generally, it is not necessary for the layout designer to draw the devices, since the latter are generally available as cells (p-cells, as we will see later). In the analog design flow, the designer should place and size the devices and draw the interconnections. The exception is given by the design of elementary digital gates (inverter, nand gates etc.) where, in order to optimize area occupation and speed, the designer defines also the shape of the individual devices.

In order to guarantee that the layout can be actually fabricated, it is necessary to respect a series of design rules, defined in the DRM and included into the PDK. A particular tool, the DRC (Design Rule Checker), is used to check whether the layout meets all the design rules. During the creation of the layout, it is necessary to launch the DRC several times and to apply corrections according to indications contained in the DRC report. The fact that the layout does not contain violations of the design rules is not sufficient to assert its correctness. Indeed, it is necessary to check also whether the layout implements the same electrical network as the schematic. The operation of checking the correspondence between schematic and layout is called LVS (Layout *Vs* Schematic). To perform this operation, the netlist actually implemented by the layout is extracted (netlist_lvs in Fig.4). Then, the LVS tool performs comparison between the layout netlist and the schematic netlist, producing a report that details all discrepancies between the two networks. Clearly, if the LVS detects errors, it is necessary to correct the layout and repeat all verifications (DRC and LVS). Netlist extraction from the layout is controlled by proper "extraction rules" included into the PDK.

For critical designs, where the parasitic components introduced by the interconnections can significantly affect the circuit behavior, it is necessary to perform a different kind of extraction, called "parasitic extraction". This operation extracts a netlist (netlist_pI in Fig.4) from the layout, including also interconnect parasitic components (capacitances and resistances). This netlist is then simulated (post-layout simulation), obtaining an estimation of the circuit performances which closely match the physical design. Due to the presence of parasitic components, the performances estimated at this stage may significantly differ from those obtained from the simulation of the schematic design. Note that the

parasitic components of most devices (Mosfets, BJTs etc) are already well represented by the device model. Thus, simulations performed in the electrical (schematic) design phase is already close to the actual behavior of the circuit, even regarding dynamic parameters (such as gain-bandwidth product). The effect of interconnect parasitics can be critical only in particular cases, such as switched capacitor circuits, complex digital architectures with particularly long interconnections or radio-frequency (RF) circuits. Clearly, parasitic extractions needs an extended set of extraction rules ("Parasitic extraction rules" in Fig.4) with respect to the simpler case of the extractions made for the LVS purpose. If the performances estimated with the post-layout simulations are not satisfactory, it is necessary to find the critical interconnects and modify the layout to reduce their impact on the circuit behavior.

When a layout that meets all specifications is obtained, it can be released for production. If we are dealing with a single cell (an "IP"), then no other steps are necessary. The cell will be stored for further use and a brief report that summarizes the cell performances will be created. In the case that the circuit that we are creating is a complete chip, then it is necessary to perform some further steps that are generally referred to as "die finishing". These steps, which are well documented in the DRM and automated by CAD routines, vary according to the foundry and the particular technological process. An example of die-finishing is the placement of "planarization dummies", which are generally metal patches placed in empty areas to obtain a nearly constant density across the chip for all metal levels, in order to facilitate planarization procedures. When the final layout is ready, it is exported in a universal format. The current format for layouts is the GDSII format (GDS means Graphic Database System while "II "stands for the roman number "2"). The GDSII is then sent to the foundry for fabrication. This final step is called "tapeout", since in earlier times, the GDSII file was saved on a magnetic tape that was physically sent to the foundry. Nowadays, file transfer is performed over an Internet connection.

3. Common features of a CAD environment

Structure of a project: libraries

In most CAD platforms, projects are organized as a set of libraries. Libraries are simply a collection of cells, which, in turn, are described by a series of views, as shown in Fig.5.

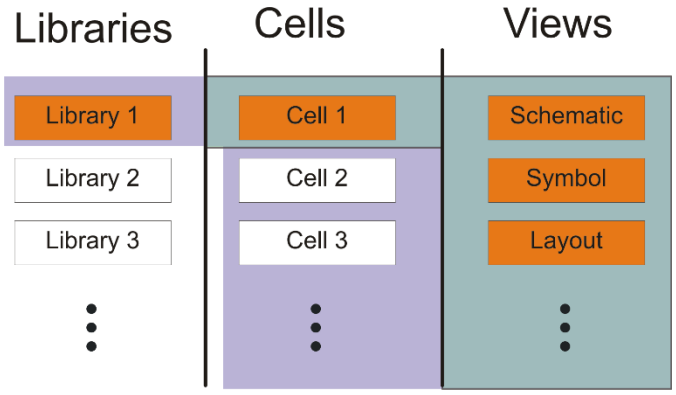

Fig.5. Project organization as a collection of libraries cells and views.

Libraries may have several purposes. The current design generally consists of one or more libraries, which includes the cells that are being developed. All devices that are available in the chosen process are given as a series of cells included in the so-called "process library". Cells that were designed by other

employees or by different companies are organized also in specific libraries. A common library is the standard-cell library, which includes all elementary digital cells to be used to synthesize digital sub-circuits (generally with a digital design flow). Finally, auxiliary libraries can also be present. These libraries includes ideal components (such as resistors capacitors, voltage sources etc.) to be used to simulate the circuits that we are developing (e.g. provide the supply voltage and input signals, represent a certain load condition etc.). These ideal components should clearly not be present in the cell being designed, since they would alter the result of the LVS.

Structure of a project: instances and hierarchy

A project generally consist of a top cell, which is the target cell that should be produced in the end. The top-cell can be either a functional block (e.g. an IP) or an integrated circuit. The top-cell is composed of simpler cells, which are connected each other by means of proper nodes called terminals. In the example of Fig.6, cell *C* includes as sub-parts cells *A*, *B* and *K*. A cell may appear into another an arbitrary number of times. For example, cell *A* appears two times into cell *C*. Every time we use a cell inside another one, we create an "instance". All instances are independent objects, even if they refer to the same original cell (as the two instances of cell *A* into *C*). This corresponds to the physical structure of the final object: cell *A* is a circuit with electron devices inside. The two instances correspond to two copies of the same cell placed into two distinct areas of the chip. The electron devices of the two instances will be perfectly independent objects, with their own voltages and currents.

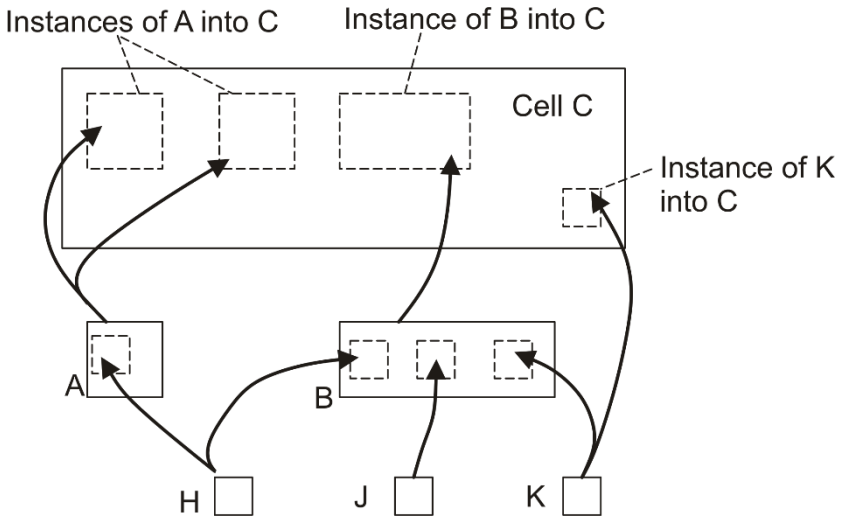

Fig.6. Example of hierarchical organization of a project

The fact that the two (or more) instances are linked to the same cell, is clearly used to optimize the design flow, since one needs to draw the cell only one time and then he/her can place the cell thousands of times with no need of redraw it. Note that if we change the original cell, all instances are changed: this is a powerful property, that allows us to update a project in a very fast way, but the designer should be well aware of it, since it may cause unwanted effects. For example, if we want to change the properties of only one instances, we cannot change the original cell but we have to create a new one and to replace the instances with it.

If we expect to make many instances of the same cell that differ by some property (e.g. a resistor value, or a MOSFET width), it is convenient to create a so-called P-Cell (Parametric Cell). A P-Cell is

characterized by a series of parameters that can be set when the cell is instantiated. In this way, it is possible to personalize the instances and it is not required to create a different cell for each different value of the parameter(s).

The fact of describing a cell as a combination of instances of simpler cells implicitly create a hierarchy, simply defined by the following law: *if cell Y includes at least an instance of cell X, then the hierarchical level of Y is higher than X one.* The transitive relation applies to hierarchy: if, in terms of hierarchy, *Y* is higher than *X* and *X* is higher than *Z*, then *Y* is higher than *Z*. Considering Fig.6, *C* is clearly at an higher hierarchical level than *A*, *B* and *K*, but, for the transitive relation, also than *J* and *H*. On the other hand, *J* and *K* are at the same level as *A* even if *J* is instantiated into *B*, which is at the same level as *A*.

A cell can contains instances of simpler cells from different libraries, which can either be custom libraries, created by the user, or PDK libraries, including all available devices or IPs created by the foundry or third parties.

Schematic editor

There are two main cell views that are involved in the schematic design:

- Schematic view
- Symbol view

The schematic view is a way to represent a cell as the connection of simpler ones. Connection occurs through terminals. A set of terminals connected together forms a node (or "net" in the CAD jargon). Then a schematic view includes instances of simpler cells and nodes. The schematic view is conveniently represented in a graphical way, where lines (wires) are used to connect the terminals and instances are represented with simple shapes (e.g. rectangles). Terminals are generally represented as small dots or short straight lines. The object that is used to represent the instance of a cell is the symbol view. Terminals are selected nodes of a cell that we intend to use in order to connect the instances of cell to other instances. For general purpose cells performing universally known functions (like logical gates, amplifiers, electron devices), a picture recalling the standard symbol is used for the symbol view (e.g. a triangle for an amplifier, classical symbols for MOSFETS, resistors, capacitors, etc.). For custom cells created for the user, a generic rectangle is preferred.

Figure 7 shows the hierarchical schematic design of a 3-input nand-gate (Nand3). The schematic view includes two instances of a 2-input nand gate (Nand2) and one instance of an inverter. The symbol view of the Nand3 includes the three input terminals and the output one. The Nand2 cell also owns a schematic and a symbol view. Let us focus on the schematic view to define all passible node types:

- **Terminals** (in1, in2, out). The purpose of terminals is to allow the cell to be connected to other ones when it is instantiated into a higher-level cell. Terminals should be defined first in the schematic view by connecting the selected nodes to proper objects (generally called "pins" or "ports". Different types of pins are possible, with "input", "output" and "bidirectional" being the most common. The symbol type is used only for checking purposes, for example to issue a warning when two output pins are connected together. In an analog design, "bidirectional" pins are often used since "input" and "output" categories are often not applicable for analog nodes.
- **Global nodes** (gnd, Vdd). Global nodes are used to connect nodes of a large number of instances together, with no need of defining a terminal for that purpose. For example, in the Nand3 cell of

Fig.7, the gnd nodes of all cells are connected together. This obvious condition recurs in practically all circuits. Use of dedicated terminals to indicate an obvious condition would reduce readability of the schematic representation. The same situation applies to the supply voltage connection (Vdd). For this reason, it is possible to make a node global. This can be done using a conventional name for that node or, preferably, connecting the node to a dedicated object that represent the global node (e.g. the gnd symbol). A set of global nodes (e.g. gnd, Vdd, Vcc, AVdd, DVdd etc) are pre-defined in the schematic editor, but custom global nodes can be created by the user, for example to bring a clock signal to a large number of cells with no need of adding a dedicated terminal.

- **Internal nodes** (N1). All nodes that are neither terminals nor global nodes are internal nodes. Internal nodes cannot be used for connecting instances together.

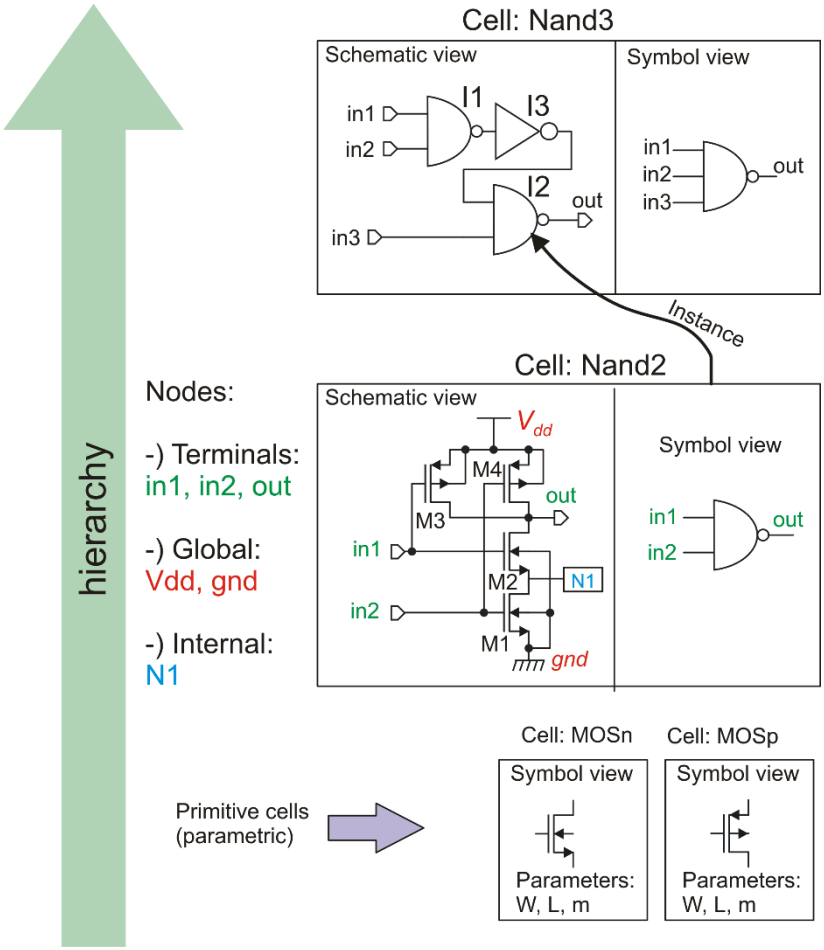

Fig.7. Elements and structure of a schematic editor.

When we create a terminal, we have to choose a name for it. Distinct terminals should have different names. When we connect a node to a symbol pin, in order to give that node the status of "terminal", we give the terminal name to that node. It is possible to give a name also to an internal node by connecting a "label" to the node. This operation can be used to connect several nodes together (forming a single node) without using wires. In fact, in complex circuits, most connections are made by label, since wires

would be too tangled to provide a fast view of the circuit architecture. If we give an internal node the same name as a pin, we connect it to the pin.

Note that there are cells that cannot be decomposed into simpler ones. These are, for example elementary devices such as MOSFETS, resistors BJTs and so on. For these cells, a schematic view is not applicable and only the symbol view is present. These cells are called "primitive cells". Primitive cells are generally P-Cells, since one or more parameters are necessary to define the properties of the cell. For example, a resistor cell will have at least a "resistance" parameter. MOSFETs will require at least a "W" (channel width) and "L" ("channel length") parameter. In most schematic editors, parameter values can be associated to a given instance by means of a graphical interface (e.g. a pop-up menu).

Even if internal nodes are not accessible for connection purposes, they actually exist, since when we create an instance of a cell, we generate an object that contains all the elements of the cell. Internal nodes are still accessible when we perform simulations or verifications (e.g. LVS). For example, if we want to plot simulation data for node N1 of instance I1 in the Nand3 cell, we have to refer to that node with the hierarchical expression I1.N1, where I1 is the name of the instance. With the same syntax, we can access an instance of a cell included into another instance. For example, device M1 (instance of a MOSFET cell) inside instance I1 in Nand3 will be I1.M1. Fig. 8 shows a few example of this way to indicate nodes and instances that are present at lower hierarchical levels. This syntax can be nested to access the lowest level of hierarchy from the current level.

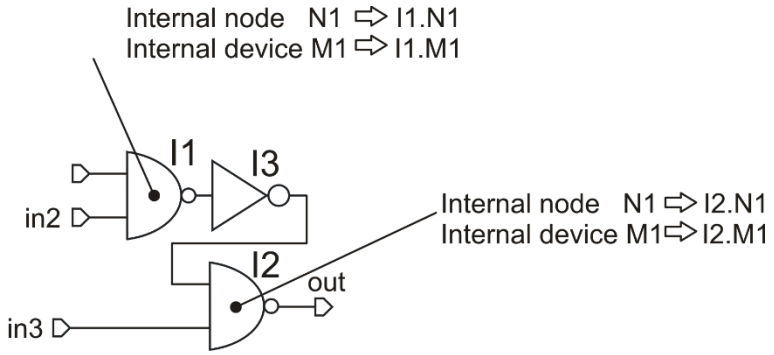

Fig.8 Typical convention used to access internal nodes and devices.

The netlist extracted from the layout generally maintains the hierarchical structure of the

Layout editor

The hierarchical organization of the schematic design is reflected into the layout design. Complex cells are formed by instances of simpler ones. However, instances appear in a different way with respect to the schematic design, where the symbol view conceals the complexity of the schematic view, improving readability. In the layout design, when an instance is created, the whole layout of the cell being placed is visible. This is necessary because, in the layout design, geometrical coordinates are of primary importance. Thus, it is necessary to have visual control on what we are placing into a layout in order to avoid that the cell being instantiated conflict with other elements already present in the layout. Furhermore, while placing (e.g. instantiating) a cell into a layout, we can immediately orientate it (rotate, flip) to facilitate interconnections with the surrounding cells, or to position it in such a way to optimize silicon area occupation. However, the instance operation is different from a simple "copy and paste"

action, since instances are selected, moved, rotated and flipped as a whole, with no risk to break them into parts. As a counterpart, we cannot select or modify any part of an instance, even if we can see all individual shapes that form it. To modify an instance we have to modify the original cell, but as already stated, the change is applied to all instances. To modify a single instance, we have to duplicate the original cell, modify it, and use it to replace the instance of interest. As for the schematic view, we can also define the original cell as a P-Cell. In the layout view, it is possible to use a specialized language to change the shape of the cell according to one or more parameters.

The instance mechanism that we have detailed above is not desirable when we have to conceal the layout of a cell. This occurs when a company designs an IP and sells it to clients, allowing them to use it in their projects, but not to analyze the layout. In this case, it is possible to use a special view called "**abstract**" view. Using this view, it is possible to show only the outline of the cell and the points (pins) where the users have to make the connections. The final layout is not made by the user, but by the foundry, where the company that designed the cell has deposited the full layout. The foundry simply replaces the abstract view with the corresponding layout view. Clearly, use of abstract views in a layout prevents DRC and LVS from being applied to the whole chip.

As shown in Fig.4, the PDK provides the layout editor with a proper set of layers. Layers can be divided into three types, according to the following list:

- **Technological or "tooling" layers**. These layers correspond to objects that can be actually fabricated on the chip with the chosen process.
- **Derived layers.** These layers are obtained with logical operations performed on other layers. For example, we can define a layer "GATE" as the intersection (= logical and) of the POLY and ACTIVE tooling layers. In that case, a GATE layer will be generated (when required) in any region of the layout where POLY and ACTIVE shapes overlap. Derived layers are mainly used to facilitate writing of the DRC and Extraction rules.
- **Service Layers.** These layers are used to add information on the layout and have no direct correspondence to objects to be fabricated on the chip. Examples are the layers to be used to mark selected interconnecting lines (pieces of metal) as pins or layers used to inform the foundry that there are areas where we have intentionally violated the DRC rules to design experimental objects.

Only the tooling layers affect the creation of the set of photomasks by which the chip is fabricated. Nevertheless, there is not a 1:1 correspondence between a layer and a photomask. This is due to several reasons listed below:

- Dimensions of the shapes drawn in the layout are those of the final object that will be fabricated on the chip. Due to several non-ideal effects occurring during the fabrication process, the patterns drawn in the photomasks are altered. Phenomena involved in this process are proximity and interference effects during UV exposure, under-etch and lateral diffusion of dopant. For this reason, masks are properly modified to take into account these effects and make sure that the final object is as close as possible to the designed one. This operation is transparent to the user and is performed by the foundry in the so-called "Mask Data Preparation" (MDP) phase.

- In order to produce an object, often several photolithographic steps are required, each one involving an individual photomask. The designer does not have generally control over the single steps, but only on the final object, for which, he/she will use only a single layer.
- Sometimes, an operation performed on the chip (e.g. doping) is frequently done only in regions that depends on other process steps. For example, in a CMOS process, n-plus doping is applied over areas where p-plus is not applied. Then we can save design time by obtain the n-plus mask by means of a complement operation (logical NOT) applied to the p-plus mask. In these cases, there will not be an n-plus layer, but only a special mask, to be used in the rare case that we do not want neither a p-plus nor an n-plus doping. The choice of the particular layer set depends on the foundry. For the example given above, a foundry may choose to provide the designer of both the n-plus and p-plus layer.

Note that, depending on the type of process step involved, the shape created in a certain layer may need to be inverted before creating the mask. In the case that the mask is a positive copy of the layer (after the adjustments detailed above, during MDP), then the layer association is marked as: "clear field". If an inversion is required, then the layer is marked "dark field". Fig. 9 illustrate this convention.

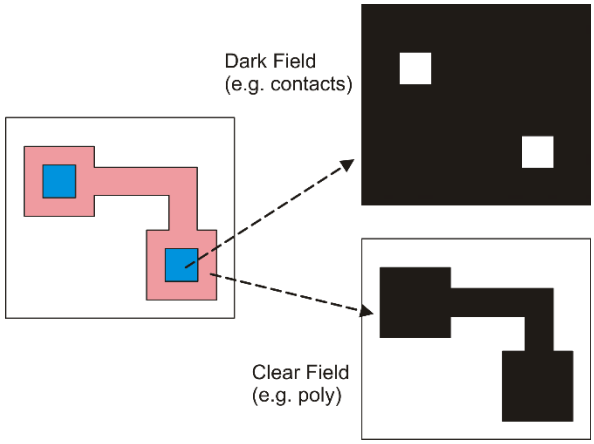

Fig.9. Layer to mask conversion in the case of clear and dark field.

Layout rules

Layout rules (sometimes referred to as "Topological Layout Rules" – TLR) guarantee that the objects that have been drawn by the designer can be actually fabricated. For example, if we draw a long thin line of metal that we want to use to carry a signal across a region of the layout, we need that the line is continuous. If the line width is too small, than there will be the risk that the line will be interrupted in some points. Then, to guarantee that the line is continuous on the chip, we need to design it with a width greater than a minimum value, reported in the DRM. Examples of design rules are illustrated in Fig. 10. Spacing rules guarantee that two object that we draw as distinct entities, actually have no interactions that can alter normal operation. This concept covers the trivial case of two conducting lines that we want to be isolated and that, if the minimum spacing is not respected, risk to come into contact. Spacing applies also to more complex cases. For example, a minimum distance can be applied to two diffusions whose depletion regions risk to came into contact, producing a punch-trough effect. The extension rule is

introduced to guarantee that a shape of layer "B " (pink in the figure) crosses a shape of layer "A" (light blue) dividing the latter into two separate parts. This is, for example, the case of planar MOSFET layout, where layer "A" is active area and "B" polysilicon.

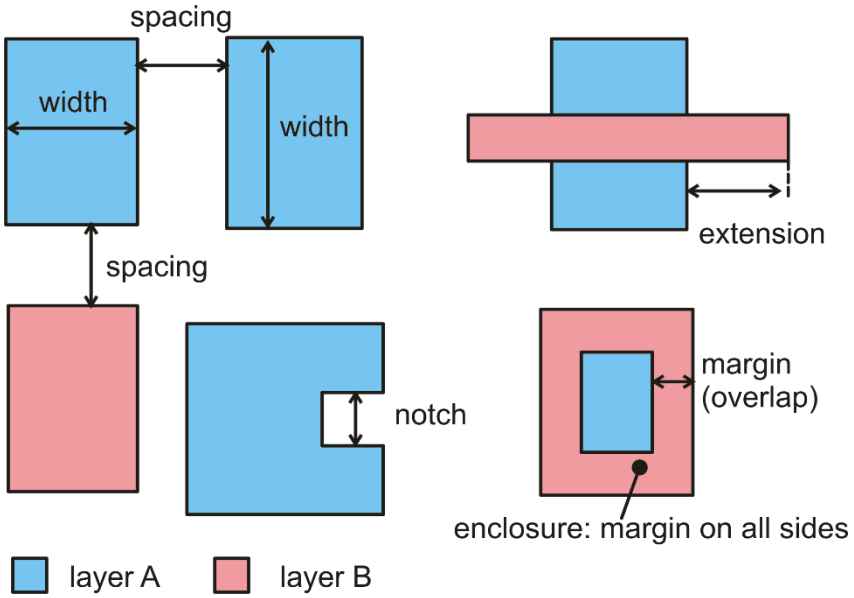

Fig.10. Example of layout rules. Notches are generally treated as spacing.

The margin rule guarantees that object in layer "A", which is drawn over object in layer "B" stays at sufficient distance from the border of layer "B". The enclosure rule commands that a margin is respected from all edges of the shape in layer "B". The margin rule is sometimes indicated as "overlap", with identical geometrical meaning.

In most cases, layout rules are expressed as minimum values. There are rare, albeit important, cases in which the rule is an exact value. This applies, for example, to the width of contacts and vias. Cases om maximum value are also possible.

An important rule, which is derived from the minimum width and minimum spacing the pitch, defined by:

$$pitch = W + S \tag{1}$$

where W and S are the minimum width and spacing, respectively. As Fig. 11 clearly illustrates, the pitch is minimum step by which objects in a given layer can be repeated. If we have to create a required BUS of N parallel lines, we need at least an N*pitch space. The pitch is the parameter that is more frequently used to indicate how the interconnection layers of a given process are efficient. Pitch applies to every element that have to be arranged in regular arrays, such as memory cells and pixels of a image sensors.

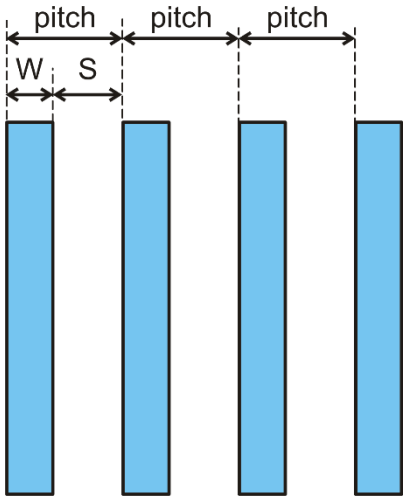

Fig.11. Definition of pitch as the sum of the minimum width (W) and minimum spacing (S).

Layout rules can be expressed in two different ways:

- Micron (or absolute) rules
- Lambda-based (or scalable) rules

Micron rules are expressed with their absolute value, typically given in micron or, more recently, in nanometers.

Lambda-based rules are given as multiples of a *length unit* called "lambda" (λ), which is given in micron and is a property of the process. The minimum dimension in a lambda-based rule set is 2λ. Other rules can be 3λ, 4λ and so on. Lambda rules were popular when scaling down of circuit size was mainly limited by the resolution of photolithography. Then, every time photolithography was improved, all rules of the process could be scaled down uniformly. Therefore, a layout designed for a given version of the process was still usable when the process was improved. The only required operation was changing (i.e. reducing) the value of lambda. The idea of lambda-based rules was introduced by Meade and Convey [1] around the end of the '70s and was aimed to facilitate migration of a design across different technological processes. Lambda rules turned out to be no more convenient when photolithography resolution got better than 1.0 μm and other physical phenomena started to contribute to limit device scaling. The design rules of modern processes do not scale down uniformly, so that they must be re-written when a technology is improved. For this reason, the design rules of practically all available IC fabrication technologies are micron rules.

There are two notable exception. The first one is constituted by the so-called MOSIS scalable rules. MOSIS is a service, based in the United States, which takes processes of international silicon foundries and re-elaborate the design rules (micron rules) to derive a set of lambda-rules. To do this and maintain manufacturability, a few rules should be properly increased, reducing the competitiveness of the process. In order to reduce the value of lambda and allow the layouts developed with a set of rules to be fabricated with an improved version of the process, it is necessary to wait until all rules are scaled down significantly. Even in that case, it is the rule that has been scaled-down by the worst factor to determine the scale-down factor for the lambda value, and then for all other rules. Layouts designed with these sets of rules are still correct, but are also somewhat larger than the process could allow if the design were

done following the actual (micron) rules provided by the foundry. The MOSIS service is mainly targeted to University or research institutions, for which the possibility of re-using older layouts and educational documents (made possible by scalability of the lambda rules) is more important than competitiveness.

Another example of scalability is represented by the so-called "shrunk" processes. These processes are upgraded versions of an older process for which it is possible to apply a uniform scaling of all rules. The scaling factor is generally modest (e.g. 10 %), so that the "shrunk" process cannot be considered as a new technology. However, this operation allow extending the competitiveness of the process, which, not being the leading technology any more, can also be offered at a discounted price. Generally, in order to save time, a new set of rules is not created and the design kit remain the same. The only difference is that all geometries are simply scaled down before manufacturing. The same scale factor is also automatically applied to all drawn geometries (i.e. MOSFET lengths and widths) before every simulations is launched.

4. Example of technology: the CMOS process

General considerations

A first example of complementary MOSFET p-n logic was proposed by F. Wanlass of Fairchild R & D Laboratory in 1963 [2]. After being limited only to ultra-low-power, low performance circuits (such as digital watches), CMOS integrated circuits progressively supplanted all other type of digital circuits. At present, CMOS technologies are by far the most used process for fabrication of all kind of digital circuits and constitute the preferred choice for mixed-signal circuits. For these reason, a standard CMOS process will be briefly recalled in this section.

The purpose of this description is showing which elementary objects (e.g. doped areas, metal shapes, etc.) can be fabricated on the chip with a typical CMOS process and highlighting the relationship of these objects with the layers that are generally available in the PDK. The way the object are fabricated (i.e. the actual technological procedures) is not the subject of this document.

Simplified process flow and layout elements of a standard 1P2M n.Well CMOS process

We will refer to a standard n-well CMOS process with one polysilicon and two metal levels (1P2M). N-well processes are used more frequently than P-well ones. Many CMOS process flows starts from a heavily doped substrate (p doping, order of 10^{19} cm^{-3}) on top of which a lightly doped epitaxial layer (p doping) is grown. Selected areas of the epitaxial layer are either n-doped (forming the n-wells) or p-doped (forming the p-wells). In this way, it is possible to set the correct doping for the n-wells and p-wells independently of the initial substrate doping (twin-tub processes), obtaining the desired threshold voltage for both the n-MOSFETs (placed inside the p-wells) and p-MOSFETs (placed into the n-wells).

An important distinction should be made at this point: since the substrate is p-type, all p-wells are electrically connected by ohmic paths. Thus, p.wells cannot be electrically independent On the contrary, n-wells forms a p-n junction with the substrate and can be isolated from the latter (and each other) by providing a proper reverse bias to the junctions. This fact has an important impact on the degree of freedom that the designer has in connecting the substrates (bodies) of MOSFETs. To make sure that the well-substrate junctions and drain/source-substrate junctions are reverse biased, the p-wells (and then the substrate) are connected to the lower supply voltage node. This node, conventionally indicated with Vss, coincides with ground (gnd) in single supply circuits. The heavily doped substrate placed under the epitaxial layer works as a ground-plane that improve potential uniformity of the p-wells, reducing the so-

called substrate noise and the risk of latch-up. The lightly doped epitaxial layer, instead, is ideal for device fabrication, due to the initial low impurity density.

CMOS processes are classified by the minimum allowed channel length for both the n and p MOSFETs. In a polysilicon-gate process, the minimum channel length coincides with the minimum width of the polysilicon lines.

Figure 12 (a) shows the situation after p-well and n-well definition. In the following step, represented in Fig. 12 (b) creation of the active areas is depicted. At this stage, the silicon surface is covered by a thick oxide layer (FOX – Field-oxide) except for the active areas, where the crystalline silicon is exposed. In modern processes (0.25 μm channel length and below), active areas are separated by shallow trenches (STI: Shallow Trench Isolation) and the FOX is the oxide used to fill the trenches. The designer decide the position and shape of the n-wells by means of the N-Well layer and the active areas by means of the Active layer. An example of layout corresponding to the cross section of Fig.12 (b) is shown in Fig.12 (c).

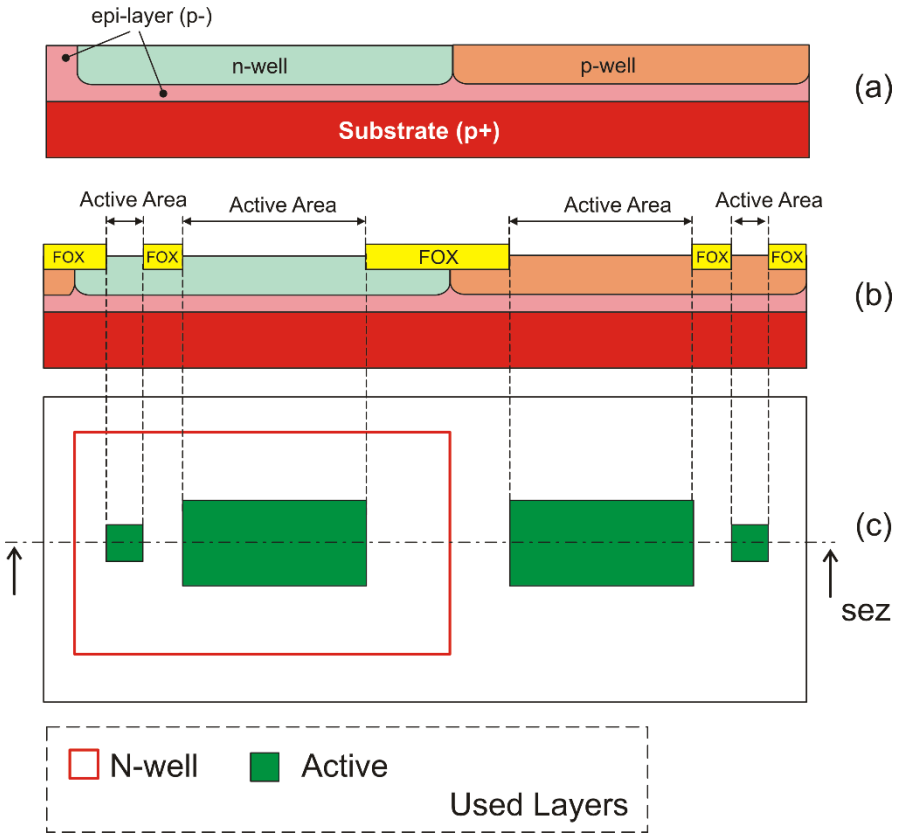

Fig.12.(a): Formation of the n-well and p-well. Note that the p-well electrically short-circuited to the substrate. (b) Formation of the active areas. (c): Top-view (layout) of the circuit at stage (b).

Note that four active areas have been represented in the example of Fig.1. The two bigger areas will be used to design an n-MOSFET and a p-MOSFET, while the two smaller areas will be used to create contacts for the p-well and n-well.

The next step is the growth (or deposition) of the thin gate oxide inside all active areas. After this step, the entire wafer surface is covered by the polysilicon layer, which is then patterned. The gate oxide is

removed together with polysilicon. Therefore, active areas not covered by polysilicon are free from gate oxide. The situation after polysilicon patterning is shown in Fig.13 (a). The following step is doping of the active areas with either n-plus or p-plus implantation. Note that four cases are possible, as shown in table 1:

TABLE 1 – POSSIBLE TYPES OF ACTIVE AREAS

Active area position	Type of doping	Function of the active area
Substrate (p-Well)	n-plus	n-MOSFET drain /source, diode
	p-plus	Substrate contact ("Substrate Tap")
N-Well	p-plus	p-MOSFET drain /source, diode
	n-plus	Well contact ("Well Tap")

Finally, both the active areas and polysilicon are silicided, i.e. their surface is transformed into a metal silicide (e.g. Titanium or Cobalt silicide). The process, which is also referred as "salicide" (Self-Aligned silicide), is used to lower the sheet resistance of active areas and polysilicon. The situation after these steps is shown in Fig.13(b), while the corresponding layout is shown in Fig. 13 (c). Only three new layers, namely Poly, N-plus and P-plus are required. Note that n-plus and p-plus doping occurs only in active areas, since it is stopped by the FOX. Therefore, the N-plus and P-plus layers can be much larger than the active areas, since they have no effect outside of the latter. Active areas must be completely covered by either the n-plus and p-plus doping. The active area portions that are covered by polysilicon are also not affected by n-plus or p-plus doping, since dopant is stopped by the polysilicon layer. This fact is essential to form the MOS transistors (self-aligned drain and source).

Polysilicon that intersects an active area forms a MOSFET gate. Fig. 13(b) and the corresponding layout of Fig.13(c) clearly shows that the bigger active areas are split into three parts by the polysilicon line that crosses them. The two areas that are not covered by polysilicon are the drain and source, which have opposite doping with respect to the underlying silicon (p-well or n-well). The part covered by polysilicon has the same doping than the substrate and forms the channel of the MOSFET. The portions of active area that have received the n-plus or p-plus doping are generally indicated as "diffusions".

Polysilicon is doped by the same n-plus or p-plus implantation used for the drain and source (once again, note that the doping layer covers the entire active areas). Then polysilicon is p-doped when it forms the gate of p-MOSFETs and is n-doped for n-MOSFETS. This is required to obtain complementary threshold voltages. Generally, it is not necessary to provide doping of polysilicon outside active areas, where it serves as an interconnect layer, since silicide is sufficient to provide a low sheet resistance.

At the stage depicted by Fig.13(b) active devices have been already created. The steps required to get to this point are indicated with FEOL (Front-End Of the Line). The following steps, which are required to provide the main interconnections, are called BEOL (Back-End Of the Line).

In order to create the interconnections, the devices and polysilicon layer should be isolated from the metal layer that will be deposited on top of them. Therefore, an insulator is deposited over the whole wafer and then holes are opened through this layer only at the points to be connected. These holes are defined by a layer called "Contacts". The situation after contact opening is shown in Fig.13 (a). Holes are filled with tungsten to provide electrical contact (tungsten plugs). After planarization, the first metal level (metal 1) is deposited and patterned. An insulating layer is then deposited and holes ("vias") are opened through it

where a contact between the metal 1 and metal 2 (second metal level) is required. Again, vias are filled with tungsten plugs, a planarization step is applied and a second metal layer is deposited. We will stop the process here, since we are considering a CMOS process with a dual metal layer. Finally, a passivation layer (dielectric) is deposited to protect the metal-2 layer from humidity and accidental scratches.

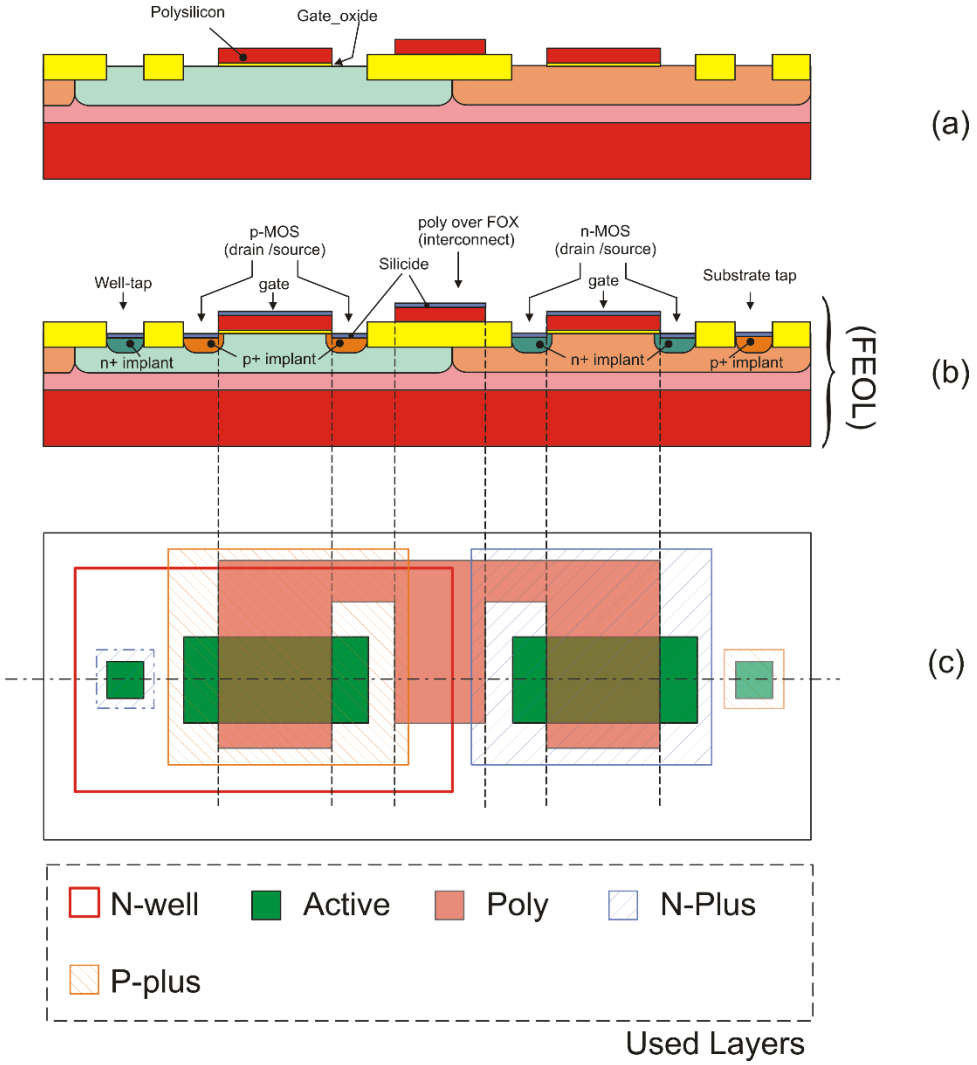

Fig.13. (a) Situation after patterning of the polysilicon layer. (b) Complete FEOL, including active area doping and Self Aligned Silicide (Salicide) growth over polysilicon and diffusions..

The new layers required to define the interconnection layers are the following: Contact, Metal 1, Via and Metal 2. Note that the Contact layer is used to connect a metal 1 shape to both an active area or a polysilicon shape, depending on where the contact is placed. Generally, contacts and vias cannot be drawn with arbitrary dimensions, but they should be drawn as squares of fixed side length, defined in the DRM. This is an example of "exact" design rule. The DRM gives also indication on the maximum current that can be carried by a single contact. To allow a connection between two layers to carry more current than a single contact (or via), the connection will be implemented using a number of contact sufficient to meet the requirement. Note that three contacts are used to connect the drain /source areas of the MOSFETS to metal-1 lines.

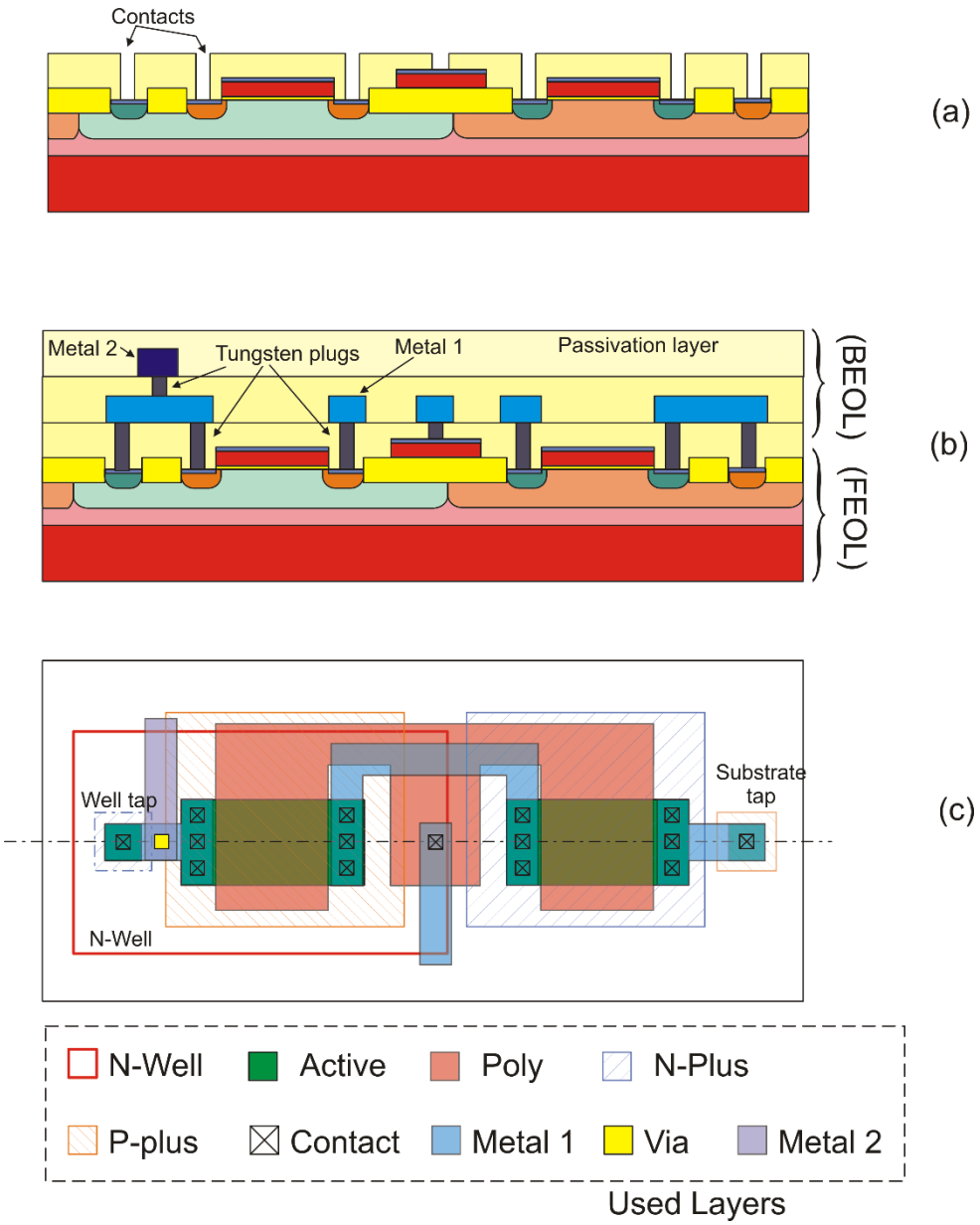

Fig.14. BEOL for a dual metal layer CMOS process. (a) Contact opening through the dielectric layer. (b) Situation after deposition and patterning of the second metal level. (c) A possible layout corresponding to the cross section (b).

Bonding pads

The fabrication process of an integrated circuit is concluded with the packaging phase. This consists in gluing the die into a package and then connecting the terminals of the chip to the package pins. Bonding may be accomplished in several ways that depend on the type of integrated circuit, the number of terminals to be connected and the chip performances. In all the cases, it is necessary to create special structures called “bonding pads” (or simply “pads”) on the die. Pads are metal areas (typically squares) as large as several tens of microns (e.g. 60 μm × 60 μm), which are suitable to be connected to similar structures that are present in the chamber of the package where the die is placed. “Wire bonding” is the

most common technique to connect the bonding pads of the chip to the pads of the package. Figure 15 (a) shows packaging of a chip into a surface mount case, with connections made by means of wire bonding. The structure of a bonding pad for a dual metal process is shown in Fig. 15 (b). Opening of the passivation over the metal 2 later is defined by means of the Passivation opening layer (often simply called "Pass", or "Passivation"). Generally more than a metal layer is stacked (metal 2 and metal 1 in the figure) to improve robustness and allow connection with several different metal layers.

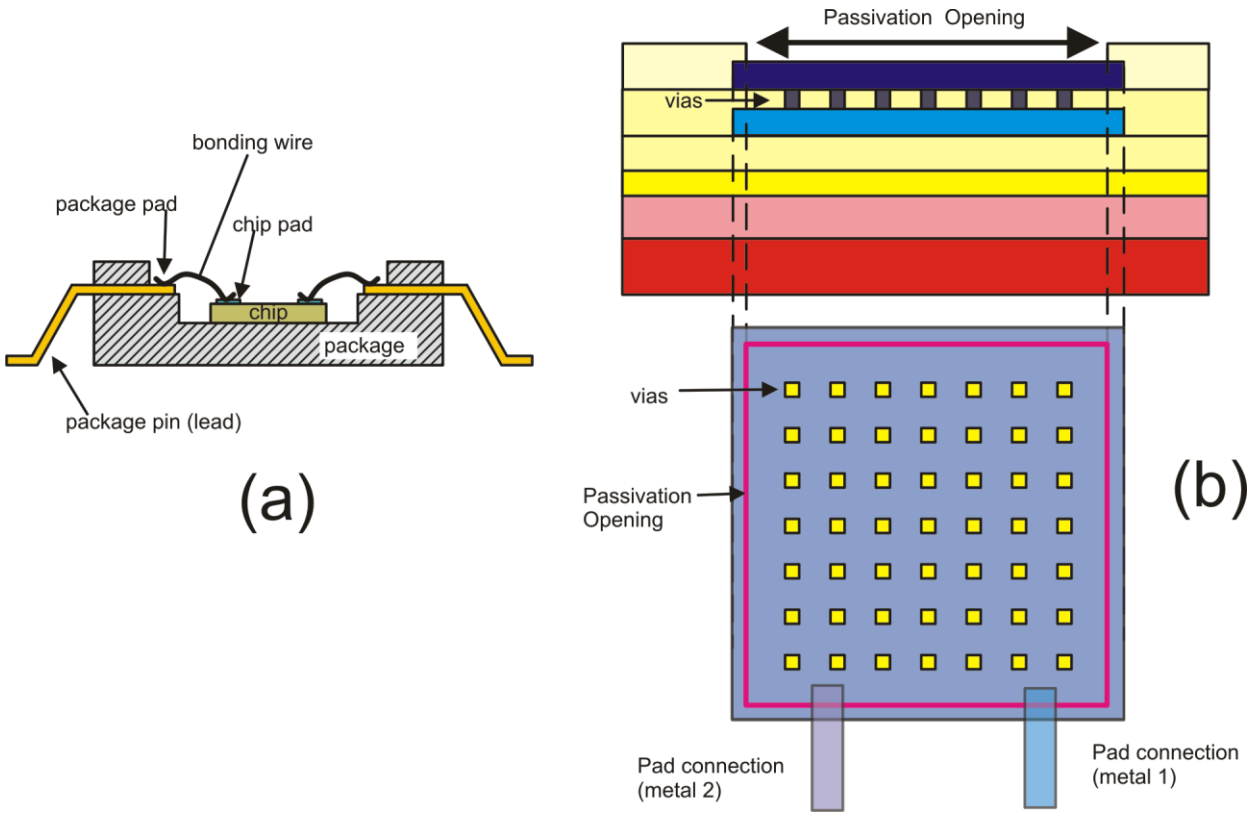

Fig.15. (a) simplified view of chip packaging and wire bonding (b) structure of a bonding pad in a dual metal process.

Possible variants present in commercial CMOS processes

Modern processes may present additional characteristics that make them more performant or more versatile. A non-exhaustive list of additional features is the following:

- More metal levels (even 6-8 metal levels). A large number of interconnection layers allows more dense integration of digital circuits and may facilitate carrying large currents by simply placing lines of different layers in parallel. Note that in fast microprocessors the power lines may carry several amperes,
- Thick metal layer. Normal metal layers are generally nearly 0.5 μm thick. This value allow very small line widths to be achieved. The upper metal level is generally thicker (up to 3 μm), in order to carry more current with reduced sheet resistances.

- Native MOSFETs (ZVT devices). These devices are n-MOSFETs fabricated on the original lightly doped substrate. In this way, their threshold is much lower than the one of regular n-MOSFETs, often close to zero. Native MOSFESs are not used in digital circuits but may be very useful for analog blocks. Generally Native MOSFETs do not require additional masks or process steps.
- Low threshold MOSFETs (LVT devices). These devices are fabricated in special n-wells or p-wells, where the doping is adjusted to obtain a lower threshold voltage. LVT devices can be either n or p MOSFETs and are used for analog applications or ultra-low voltage logic circuits.
- MOSFET families with different voltage ratings. Deep submicron process feature MOSFETs that are damaged by supply voltages higher than 2 V. The weaker point is the oxide gate, which imposes strict limits to the gate-source and gate-drain voltage. Unfortunately, communication between different chips on a printed circuit board still occurs at higher voltages (typically 3.3 V). To allow the chip to communicate with such voltage levels, the process generally includes an additional CMOS family (n and p MOSFET pair) that can stand higher voltage thanks to a thicker gate oxide. These devices are larger and slower than the so-called "CORE" devices of the process and, for this reason, are used only to build I/O buffers or analog blocks. Clearly, additional masks are required to decide in which active areas the thicker gate oxide has to be deposited.
- Passive devices. These devices (resistors, capacitors and, less frequently, inductors) are practically indispensable for precision analog circuits. Generally, additional process steps and/or masks are required. For example, polysilicon resistors require a mask that prevents silicide to be grown onto selected polysilicon stripes that have to work as resistors.
- Triple well option. In the twin-well (twin-tub) process described above, all p-wells are connected to the substrate, which, in turn, is connected to the lowest supply voltage. As a result, it is not possible to connect the body of selected n-MOSFETs to their sources to avoid the body effect. This is possible for p-MOSFETs, since their bodies are the n-wells, which are insulated from the substrate. The fact that all p-wells (i.e. the bodies of n-MOSFETs) are connected together may introduce unwanted coupling between different sub-circuits (substrate noise). Triple-well processes allow creation of p-wells that are insulated from the substrate. The principle is illustrated in Fig.16. The key element is a buried well, which is obtained by means of high-energy ion implantation, or simply as a buried layer created before epitaxial layer growth. The p-well to be insulated is closed into a box, whose side walls are obtained with a ring made with the conventional n-well. The bottom of the box is closed by the buried n-well. Triple well process, beside allowing independent biasing of both p-wells and n-wells, are also suitable for space applications since they are less prone to the latch-up phenomenon, which is much more severe in environments exposed to high energy particles.

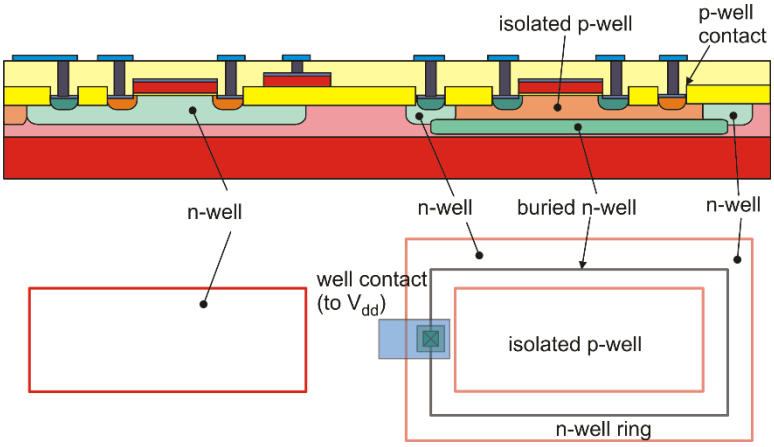

Fig.16. Cross-sectional view and layout of an insulated p-well obtained with a triple-well process.

5. Brief list of technologies alternative to CMOS

CMOS technology is used to fabricate the largest part of integrated circuits produced every year in the world. Currently there is a very large number of CMOS processes available in the market, covering most cost/performance combinations. Speed improvements has recently allowed CMOS technologies to compete with Bipolar and BiCMOS processes in RF applications. The refinement of switched techniques for offset and flicker noise reduction is making CMOS analog circuits match the precision performances of Bipolar amplifiers.

Nevertheless, there are fields where different technologies may still represent a better choice. Table 2 includes significant alternatives to the CMOS technology, including both long established technologies and emerging materials.

TABLE 2. ALTERNATIVE TECHNOLOGIES TO PURE CMOS PROCESSES

Technology	Available Devices	Notes
Bipolar	Vertical NPN, Lateral PNP	Used for precision and/or fast amplifier. Si-Ge versions for RF applications
Complemetary Bipolar	Vertical NPN, Vertical PNP	
BiFet	BJTs and JFETs	Used for precision / low bias current amplifiers
BiCMOS	CMOS + BJTs	Especially relevant for Mixed Signal System on a Chip including RF links or high speed digital communication interfaces.
BCD	Bipolar, CMOS, DMOS	Invented by STMicroelectronics, BCD technologies are nowadays the best choice for smart power applications, due to the high voltage / high power capability of Doble Diffused MOSFETs (DMOS)
SOI Silicon on Insulator.	Depends on the process from which it is derived (CMOS, BiCMOS or BCD)	High voltage applications. Space applications, due to resistance against latch-up
GaN, SiC	BJTs, MESFETs, HEMT	Use of wide-gap semiconductor is particularly promising for high power devices and RF applications.

6. Resistances and capacitances in Integrated Circuits

Every shape designed using a conductive layer (metal, polysilicon, doped active area etc) exhibit resistances and capacitances. As we have seen, simulations that take into account all interconnect parasitic components are called "post-layout" simulation and require a preliminary phase of parasitic extraction.

Resistances

The example of Fig.17 shows a rectangular interconnect line. The resistance between the two indicated terminals is simply given by:

$$R = R_S \frac{L}{W} \tag{2}$$

where R_S is the sheet resistance which is measured in Ohm, or in Ohm/squares (Ohm/sq). The latter unit is still equivalent to Ohm, since "squares" stands for "number of squares", meaning the number of squares of side W that can be lined-up along L. Notice that number-of-squares=L/W.

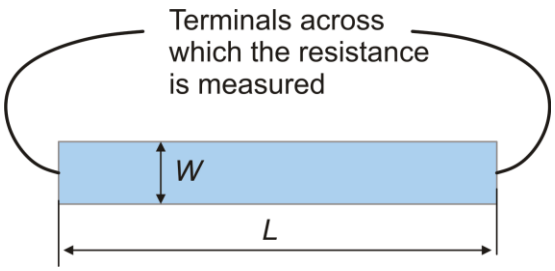

Fig. 17. Length and width of an interconnection line.

Capacitances

The typical capacitances that affects interconnection lines are shown in Fig. 18 (a). Capacitances can be of vertical (C_V in Fig.18) or lateral (C_O in Fig.18) type. Vertical capacitances are created every time two interconnecting lines of two different layers intersect. The intersection capacitance is given by:

$$C_V = k_A A + k_P P \quad \text{with} \quad A = W_A \cdot W_B \quad \text{and} \quad P = 2W_A + 2W_B \tag{3}$$

where A e P are the area and the perimeter of the intersection, respectively, while k_A and k_p are the capacitance per unit area and unit perimeter, respectively. The capacitance proportional to the intersection perimeter is due to the electric field line that, as shown in Fig.18 (a), extend from the edges out of the intersection area. Perimeter capacitances (also called "fringe" or "edge" capacitances) give a significant contribution when one or both sides are very small. If both sides are large (typically much larger than the minimum width) then the area contribution dominates.

The lateral capacitance occurs between two lines that are on the same layer. A rough approximation that does not take into account boundary (fringe) effects is the parallel plate formula:

$$C_L = \varepsilon_d \frac{L \cdot t_m}{d} \tag{4}$$

where ε_d is the permittivity of the dielectric between the lines, tm the thickness of the lines and L the length of the segments that runs parallel to each other. For lines spaced 0.25 μm, the lateral capacitance can be of the order of 0.1 fF/μm. This seems a very small value. However, for lines that runs parallel several hundred micros, the lateral capacitance can be as large as tens of fF. Since C_L introduces coupling between two otherwise independent lines (i.e. a cross-talk), the circuit performances can be strongly degraded. The same consideration apply to polysilicon lines.

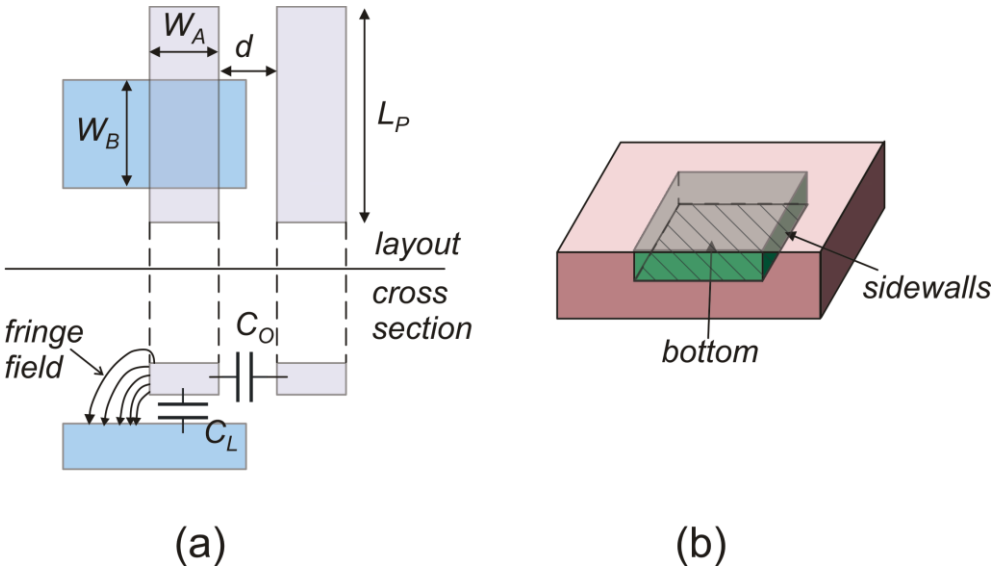

Fig. 18. Layout and cross-section of metal lines coupled by parasitic capacitances (a); simplified view of a diffusion embedded into a substrate (b).

If the conducting layer is a doped active area (diffusion), then it will be insulated from the substrate by a reverse-biased junction. Then the boundary of the diffused later will consist in a bottom surface and four sidewalls. Since the junction depth is fixed, the total sidewall area is clearly proportional to the perimeter. Therefore, the capacitance will be the sum of an area and a perimeter contribution, just as in (3). Since junction capacitances decrease as the reverse bias is increased, a worst-case estimation will be based on the zero-bias capacitances.

7. Bibliografia

[1] Carver Mead and Lynn Conway, “Introduction to Vlsi Systems” Addison-Wesley, Reading (MA), 1980).

[2] Wanlass, F. M. and Sah, C.T. "Nanowatt Logic Using Field-Effect Metal-Oxide Semiconductor Triodes," ISSCC Digest of Technical Papers, February 20, 1963, pp. 32-33.