



Budapest University of Technology and Economics
Department of Electron Devices

BSc Course in Microelectronics

Laboratory Practice: CMOS circuit design and simulation

- Read through this summary carefully and answer the questions listed on the last page (similar questions are expected in the midterm test)

Introduction to field effect transistors

The metal–oxide–semiconductor field-effect transistor (MOSFET, MOS-FET, or MOS FET) is a transistor used for amplifying or switching electronic signals. Unlike the bipolar junction transistor, the MOSFET is a unipolar device.

The MOSFET is a four-terminal device with source (S), gate (G), drain (D), bulk (B). In practical applications the bulk is shortened to the source, therefore the bulk is not shown on schematic symbols. The MOSFET is by far the most common transistor in both digital and analog circuits, though the bipolar junction transistor was at one time much more common.

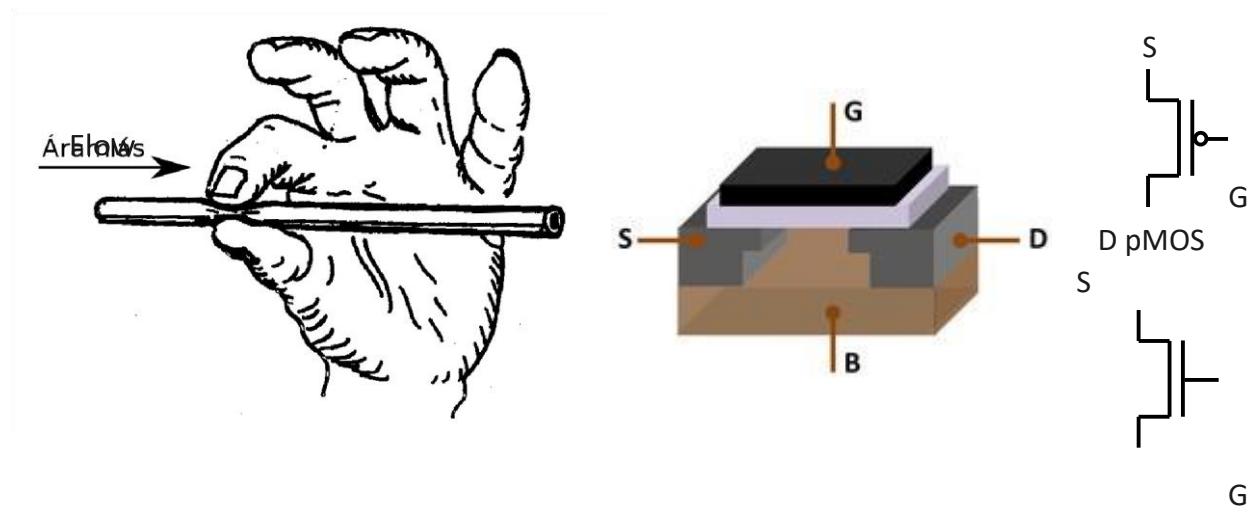


Fig 1. a) understanding the field-effect b) cross section of a MOSFET transistor, c) schematic symbols of pMOS and nMOS transistors

In enhancement mode MOSFETs, a voltage drop across the oxide induces a conducting channel between the source and drain contacts via the field effect. The term "enhancement mode" refers to the increase of conductivity with increase in oxide electric field that adds carriers to the channel, also referred to as the inversion layer. The channel can contain electrons (called an nMOSFET or nMOS), or holes (called a pMOSFET or pMOS), opposite in type to the substrate, so nMOS is made with a p-type substrate, and pMOS with an n-type substrate.

The operation can be imagined by a straw tightened by two fingers. The flow-through rate can be affected by the force of tightening. In field effect transistors the channel can be closed or opened by applying external forces as well, though the external force is the voltage applied to the gate electrode. Therefore, the electric current flowing through the channel (source to drain) is affected by the gate voltage applied (drain current is controlled by gate voltage). Note, that in bipolar junction transistors the current flowing through the device (from emitter to collector) is affected by the base current (collector current is controlled by base current).

MOSFETs in digital circuits

MOSFETs are commonly used in digital circuits. For investigating the digital operation, the following rules may apply:

- In switching mode (in digital circuits) only two states of the transistor are utilized: the channel conduct, when we say 'the transistor is opened', or the channel does not conduct, when we say 'the transistor is closed'.
- Binary values are corresponding voltage values. E.g. binary 1 means 5 V, binary 0 means 0 V.
- From this aspect the operation of the nMOS and pMOS are the opposite.
- The nMOS transistor is normally closed, and opens when a positive voltage (e.g. 5 V) is applied to the gate electrode. When the gate voltage is 0 V, the nMOS transistor is closed.
- However, the pMOS transistor closes when a positive voltage (e.g. 5 V) is applied to the gate electrode, but the channel is opened otherwise (e.g. the gate voltage is 0 V).
- Schematically, when the transistor is opened, it can be substituted by a short. When the transistor is closed, it can be substituted by an open.

Digital circuits are commonly built of using both pMOS and nMOS transistors. This type of digital circuits is called cMOS (means complementary MOS). A cMOS circuit consists of a pMOS circuit block connected to the power supply (V_{dd}) and an nMOS circuit block connected to the ground (V_{ss}). Two important characteristics of CMOS devices are high noise immunity and low static power consumption. Since one transistor of the pair is always off, the series combination draws significant power only momentarily during switching between on and off states. Consequently, CMOS devices do not produce as much waste heat as other forms of logic, for example transistor-transistor logic (TTL) or NMOS logic, which normally have some standing current even when not changing state. CMOS also allows a high density of logic functions on a chip. It was primarily for this reason that CMOS became the most used technology to be implemented in VLSI chips.

CMOS inverter

The simplest digital circuit is the inverter. An inverter has an input and an output. The output is always the opposite value of the input. The figure above describes the way of operation. When the input is '1' (gate voltages are e.g. 5 V) the pMOS closes and the nMOS opens, therefore the output is shorted to the ground. The output voltage equals the ground potential, the digital value is '0'. If the input is '0' (gate voltages are 0 V) the pMOS opens and the nMOS closes. The output is shorted to the power supply, therefore the output voltage refers to '1' (e.g. 5 V).

IN	OUT
A	NOT A
1	0
0	1

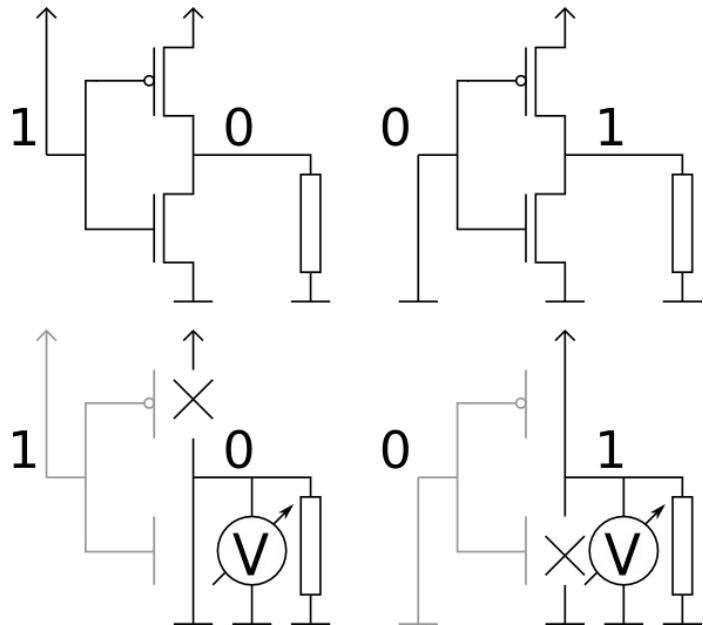
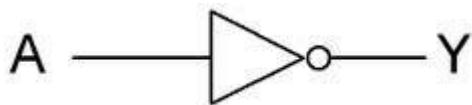


Fig 2. The inverter operation

Questions

1. What MOSFET stands for? What is the difference between nMOS and pMOS? How the terminals are called?
2. Describe the main differences between a bipolar junction transistor and a field effect transistor.
3. What is cMOS? What are the main advantages of cMOS circuits?
4. *Collect information from the internet:* What is the minimal feature size (MFS) for a modern CPU today?
5. Describe the operation of a cMOS inverter, if the input is 1 (/ 0).

Analog Circuit Design I. – Circuit simulation of a CMOS inverter

In this lab, we will use LTspice XVII, a widely used industrial circuit design tool. Since it is freeware, it's free to download and use, and there's no upper limit of nodes, components, or even sub-circuits.

Main parts:

- a) Schematic design editor
- b) SPICE simulation engine
- c) waveform display

Run simulations:

- a) time-domain (transient)
- b) small-signal (AC)
- c) large-signal (DC)
- d) large-signal transfer (DC Transfer)
- e) operating point calculation
- f) noise

The software is also equipped with the features needed to design switching power supplies, which is the main application of this tool today. However, it is not suitable to design printed wiring boards (in a discrete case), and neither to create the physical layout of integrated circuits (layout), nor for logical simulations.

You can download the latest version of the software from the following link:

<https://www.analog.com/en/design-center/design-tools-and-calculators/ltpice-simulator.html>

Around the middle of the page, click on *Download for Windows 7, 8 and 10*. A 42 MB installer will start to download. Once it's done, install it on your computer in the usual way. If you have not changed the installation location, you will find the program in the `c:\Program Files\LTSpiceXVII\` folder along with other files (parts symbols, components, etc.) required for its operation. In order to design integrated circuits provided by Austria Microsystems (which are N and P channel MOS transistors), we need to add additional symbols and models to the contents of the `lib` folder. To do this, you need to download the AMS components from the EDU system.

After unpacking (or looking into .zip with Total Commander for example), you will find two folders. Inside the `sym` folder you can find an `AMScells` subfolder containing two `.asy` files, which are no less than hierarchical symbols (NMOS, PMOS) edited with a graphical editor. As you look into them with a text viewer (F3 in Total Commander), you can see that the symbol drawings are described in a unique format. The subfolder contains an `AMSLv49.sub` file which is no less than the description of the transistor model in SPICE language. Here, we are talking about the BSIM3 level 49 transistor description, which is a quite complex model with many parameters.

Copy the „`ltpice`” folder to the root of C: drive, and copy the „`sym`” folder `C:\Users\USERNAME\Documents\LTspiceXVII\lib`. IMPORTANT! DO NOT copy it to `C:\Program Files\LTSpiceXVII\` because it will not be visible for the program. In order to use the newly added components, you need to create an `.include` directive in the schematic, which will be discussed later.

Start the circuit design tool by clicking its icon on your desktop or by using the Start menu shortcut (path: "C:\Program Files\LTSpiceXVII\XVIIx64.exe").

Use the File - New Schematic command to create a new wiring diagram. Components, wiring and simulation commands can be placed here.

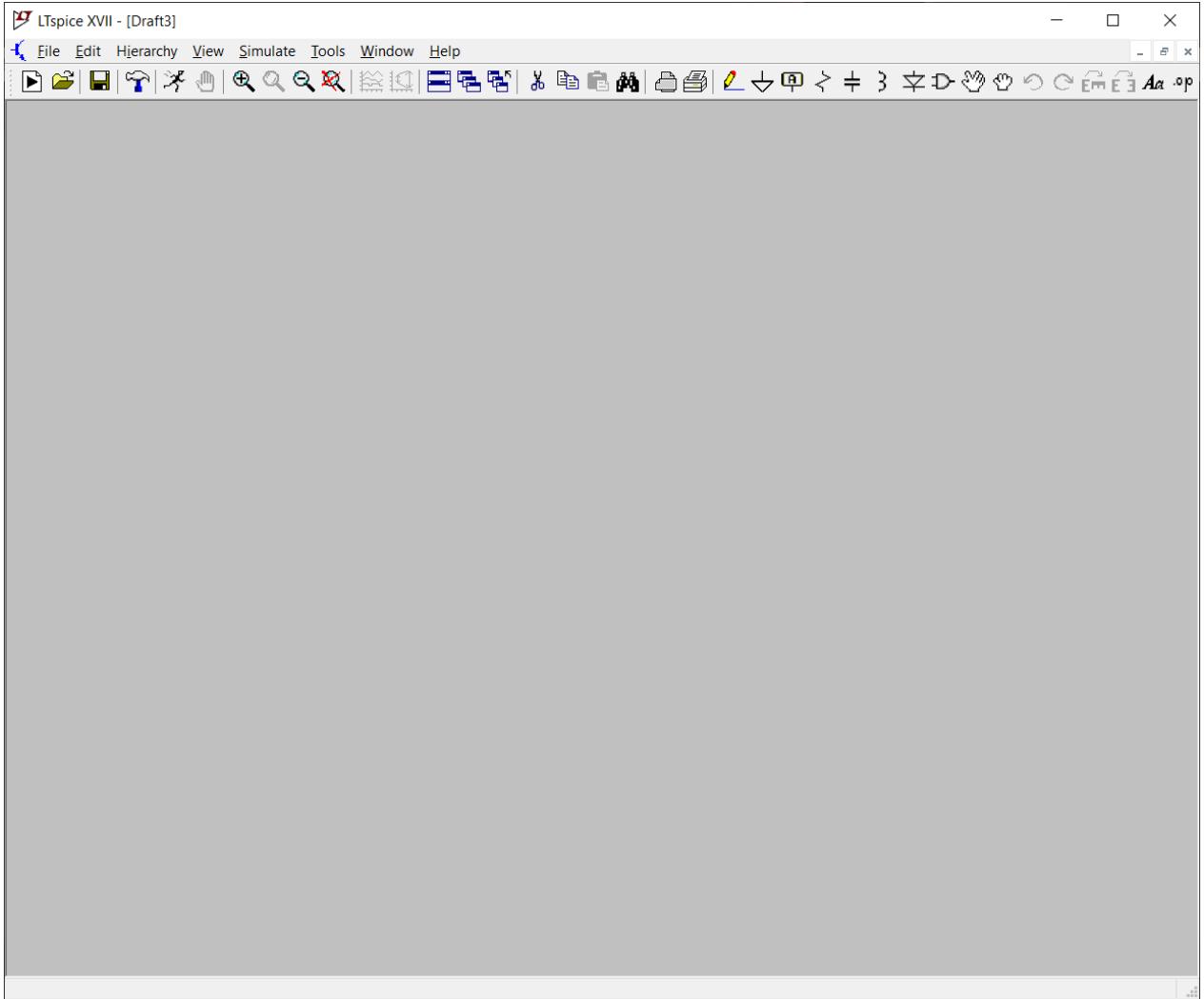


Figure 1. Schematic design window of LTspice

Click on the Edit drop-down menu to see a list of commands needed to create the schematic, with shortcuts enclosed in apostrophes. Some follow logical pattern (e.g. resistance - 'R'), but there are some interesting ones (e.g. wire - F3, undo (Ctrl + Z) - F9).

Let's draw an inverter!

Select Edit – Component 'F2', where we can select a component from the default folder (`C:\Users\USERNAME\Documents\LTspiceXVII\lib\sym`). The folders are in square brackets, and they contain more components. Please choose `AMScellsDigit` folder, and `p4` component, and place one. (If you cannot find `AMScellsDigit` here, please try to copy the content of the .zip file into the right folder). Now go back to Edit – Component 'F2', select `AMScellsDigit` folder, and `n4` component, and place one (Fig. 2).

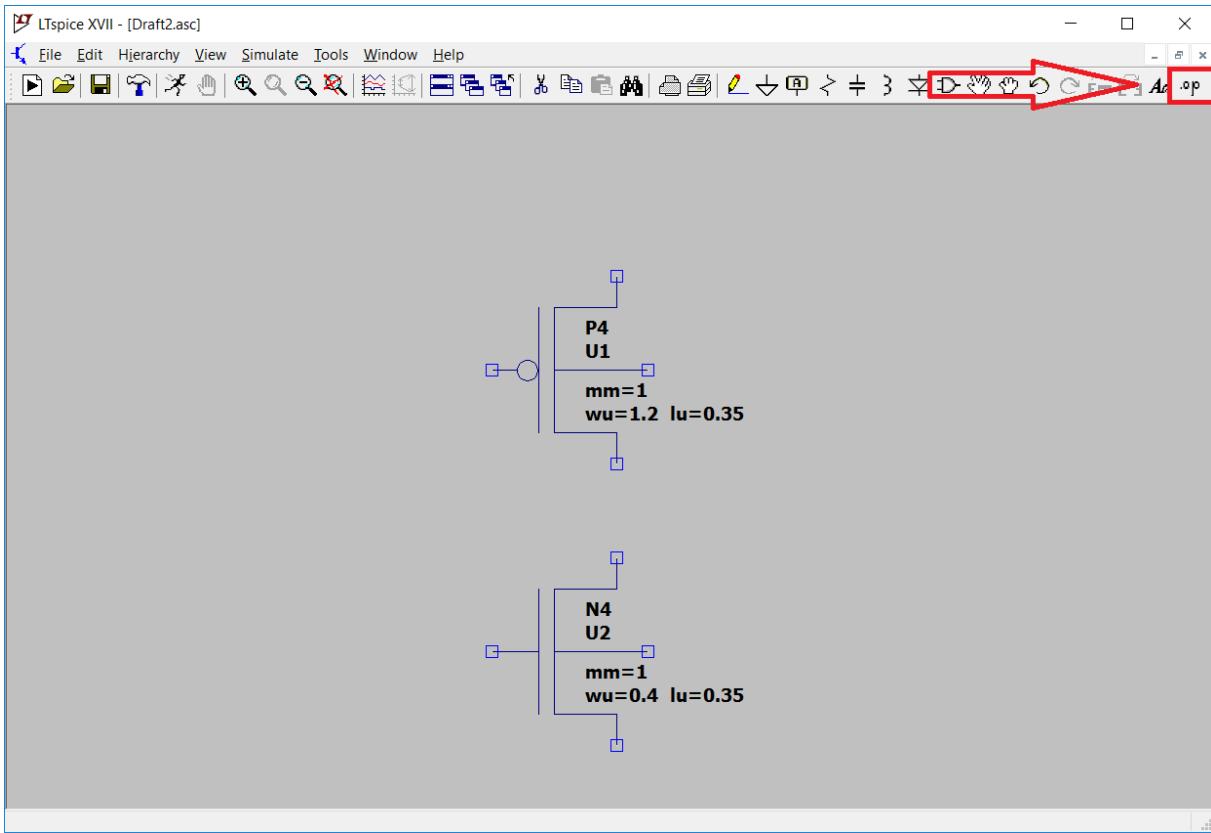


Figure 2. Placement of MOS transistors

Now we have to click on the .op button (indicated in Fig.2.) to create a SPICE Directive. Here we can define the path of the model file. Insert this line below:

```
.include c:\ltspice\sub\AMSLev49Digit.sub
```

And place it somewhere on the schematic (like in Fig.3.).

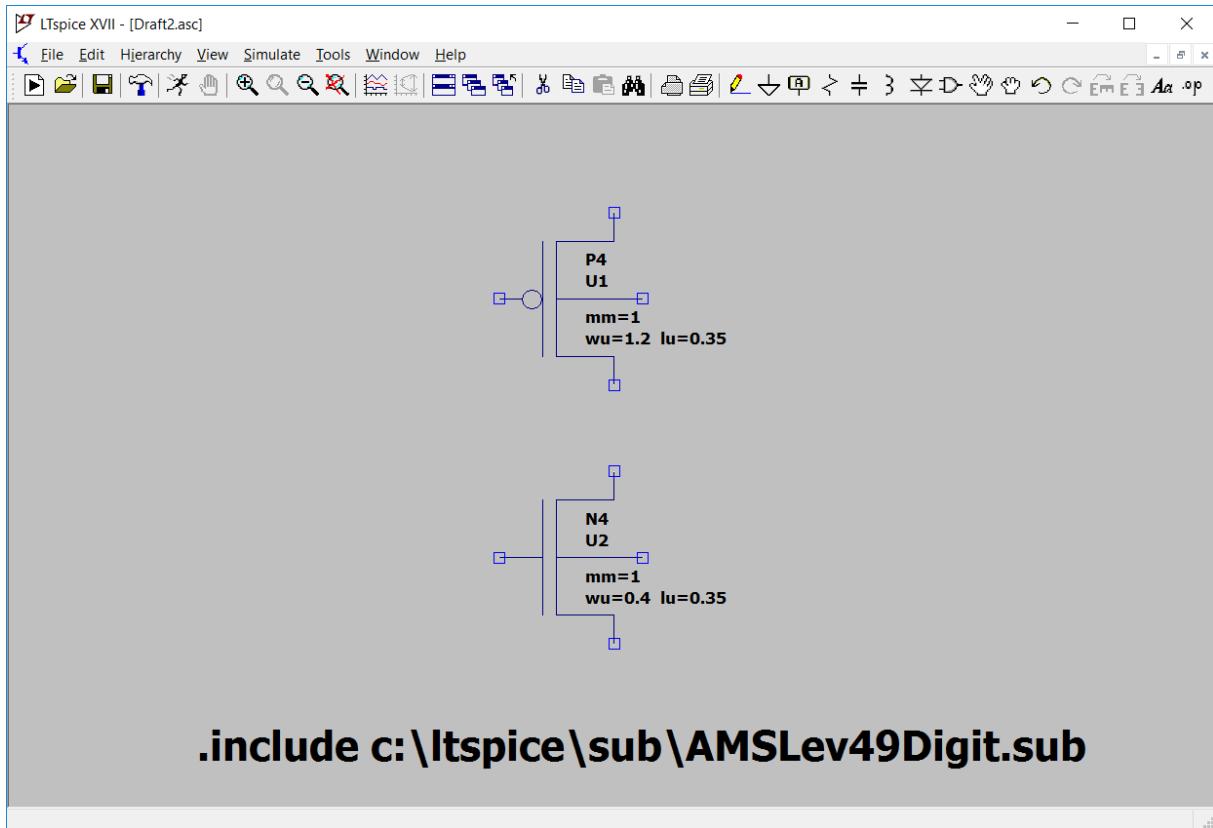


Figure 3. Add SPICE Directive

Now we have to insert two voltage sources, one for the power supply, and one for the input.

From Edit – Component ‘F2’ choose ‘voltage’ (you might have to go back from a subfolder clicking on[..], and place two of it.

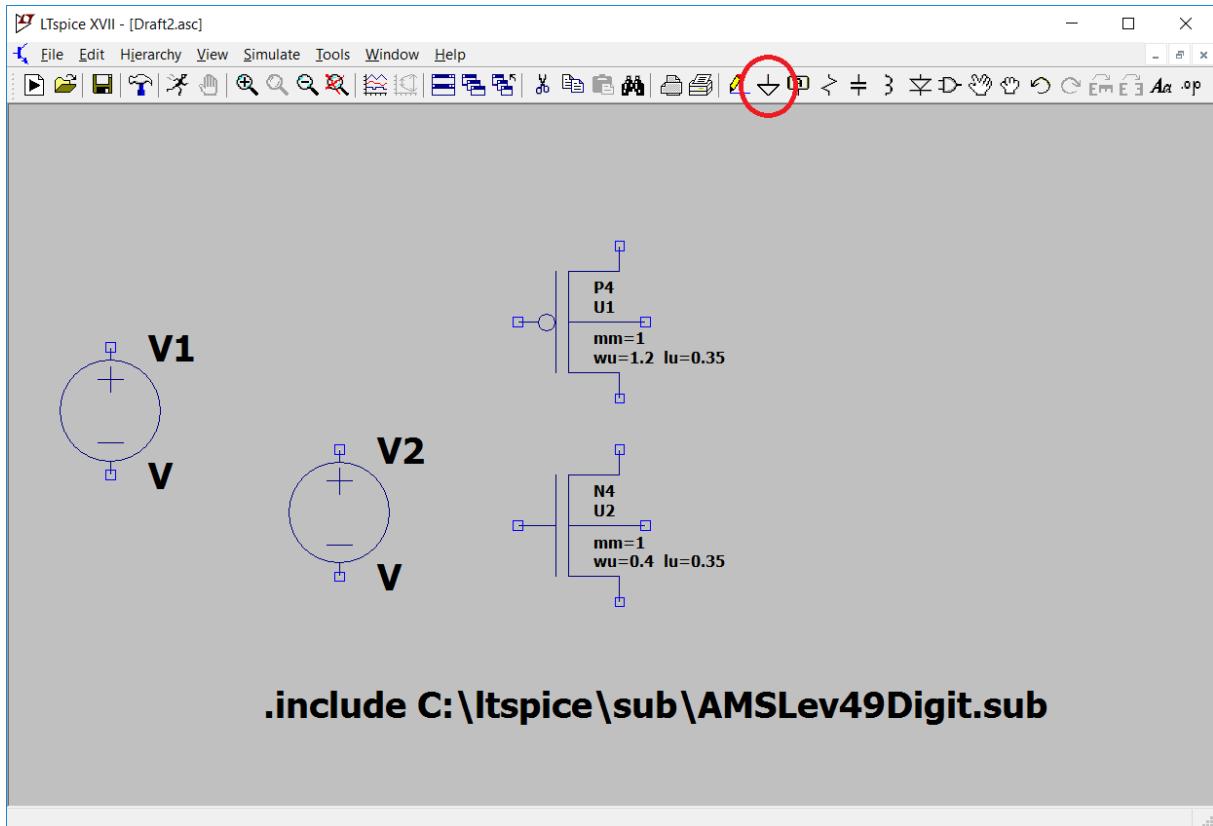
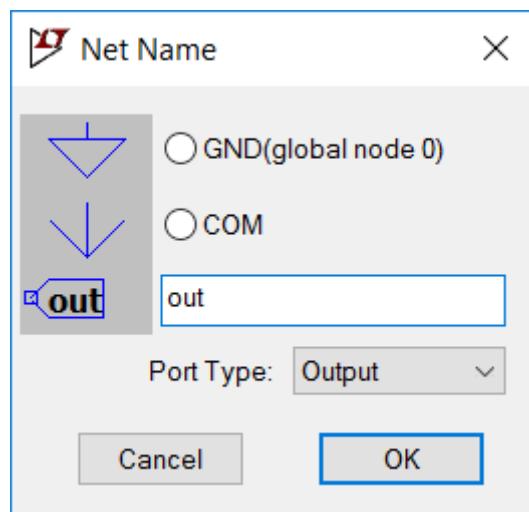


Figure 4. Insert a Ground

Now place three of Ground components (see Fig. 4.). Now we can add an output port to the circuit.

Select button, and Set the Port Type to Output, and write *out* into the input field.



Now we can wire the circuit, so please use Edit – Draw wire ‘F3’ command to do it.

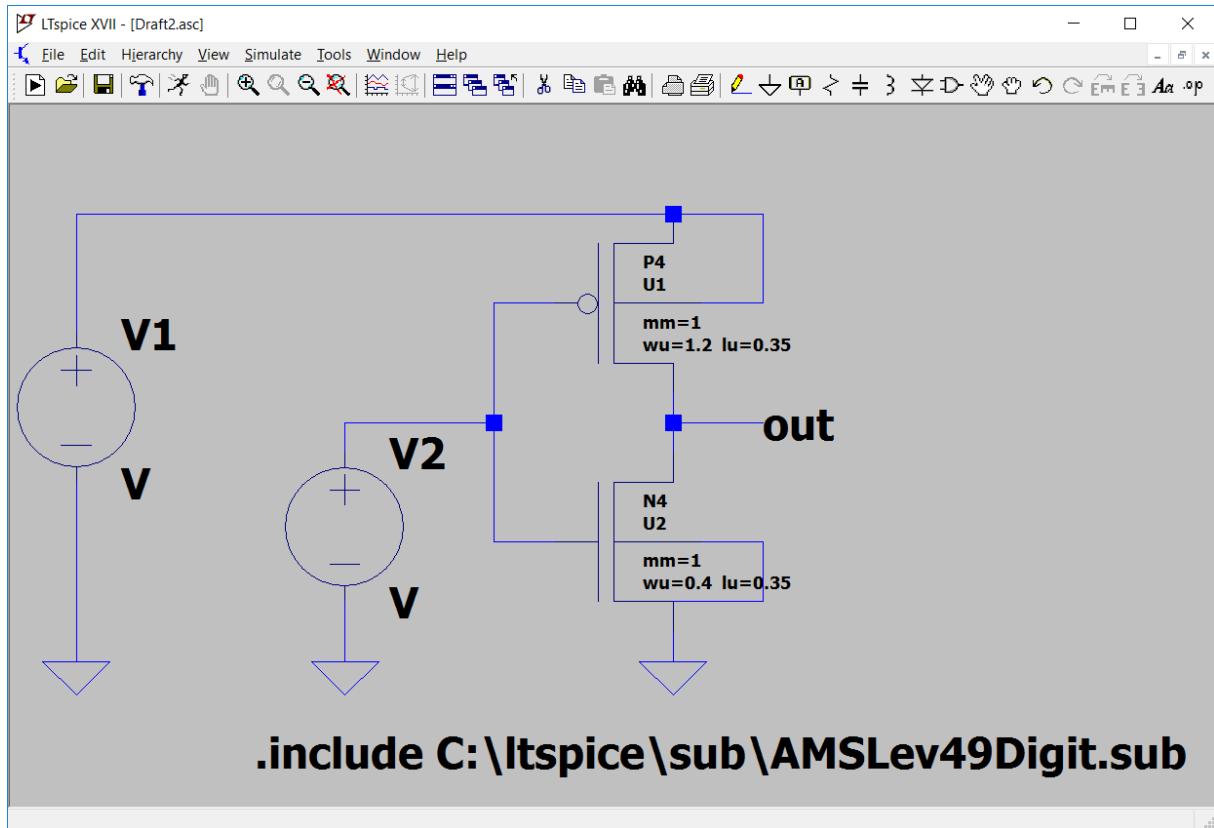


Figure 5. Wiring the schematic

Now we have to set the voltage of the voltage sources. We can do it by clicking the right mouse button on it. The power supply voltage has to be 5 V (see Fig.6.).

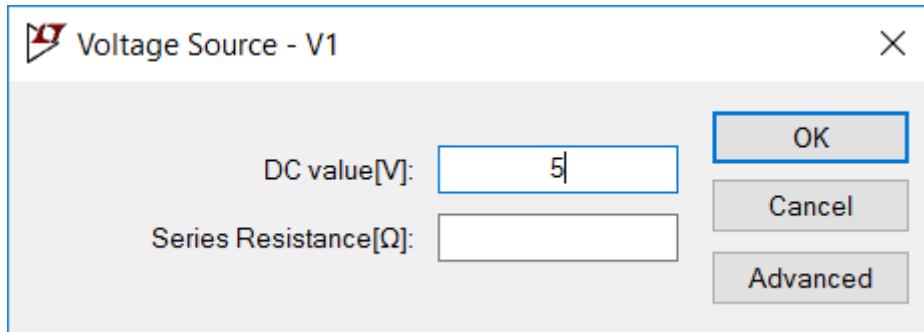


Figure 6. Power supply voltage

Set the other voltage source to 0.

DC Simulation

Now we are ready for the first simulation. Select Simulate – Edit Simulation Cmd, and choose DC sweep tab. Fill the input field, as it can be seen in Fig. 7.

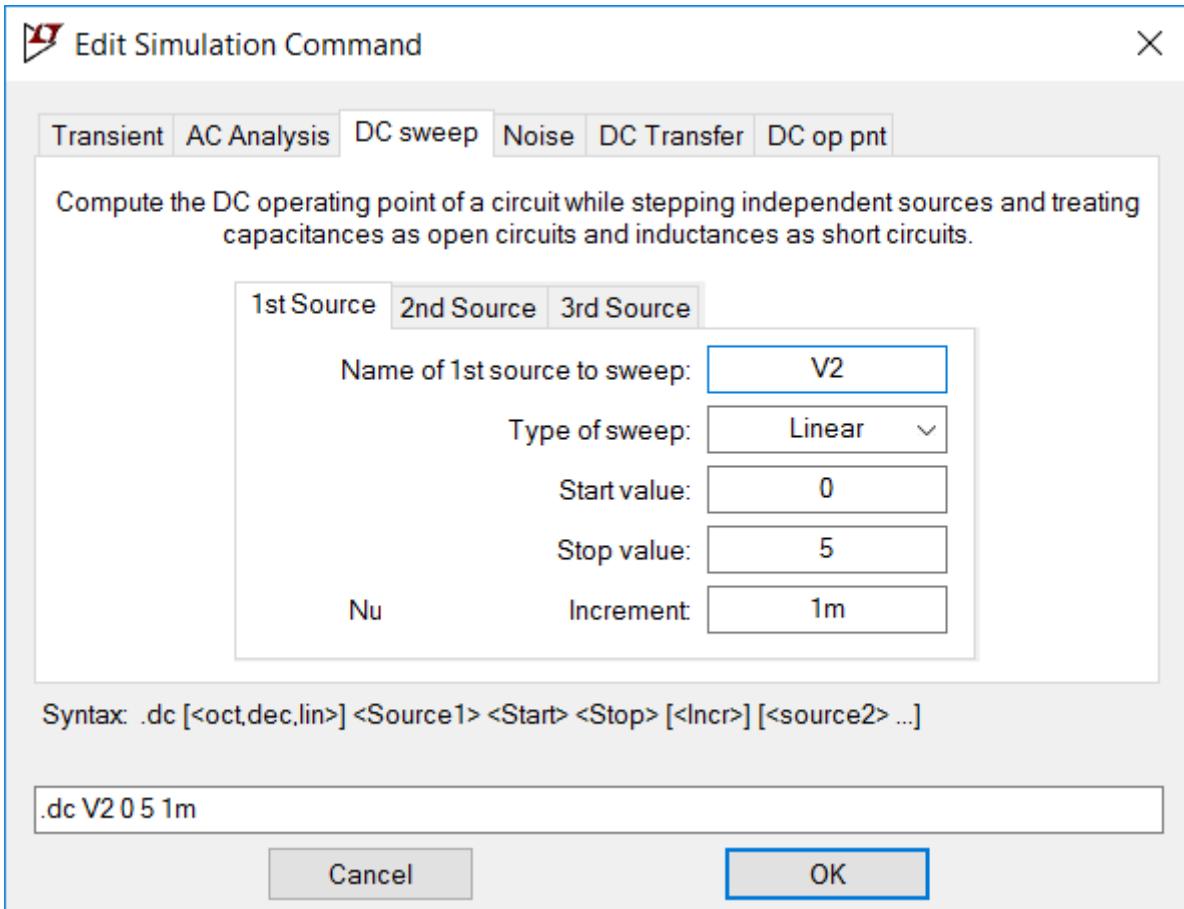


Figure 7. DC sweep simulation settings.

If you are done, click on OK, and place the simulation command on the schematic. Select  button to perform the simulation. If everything is done, you can see an empty diagram like in Fig.8.

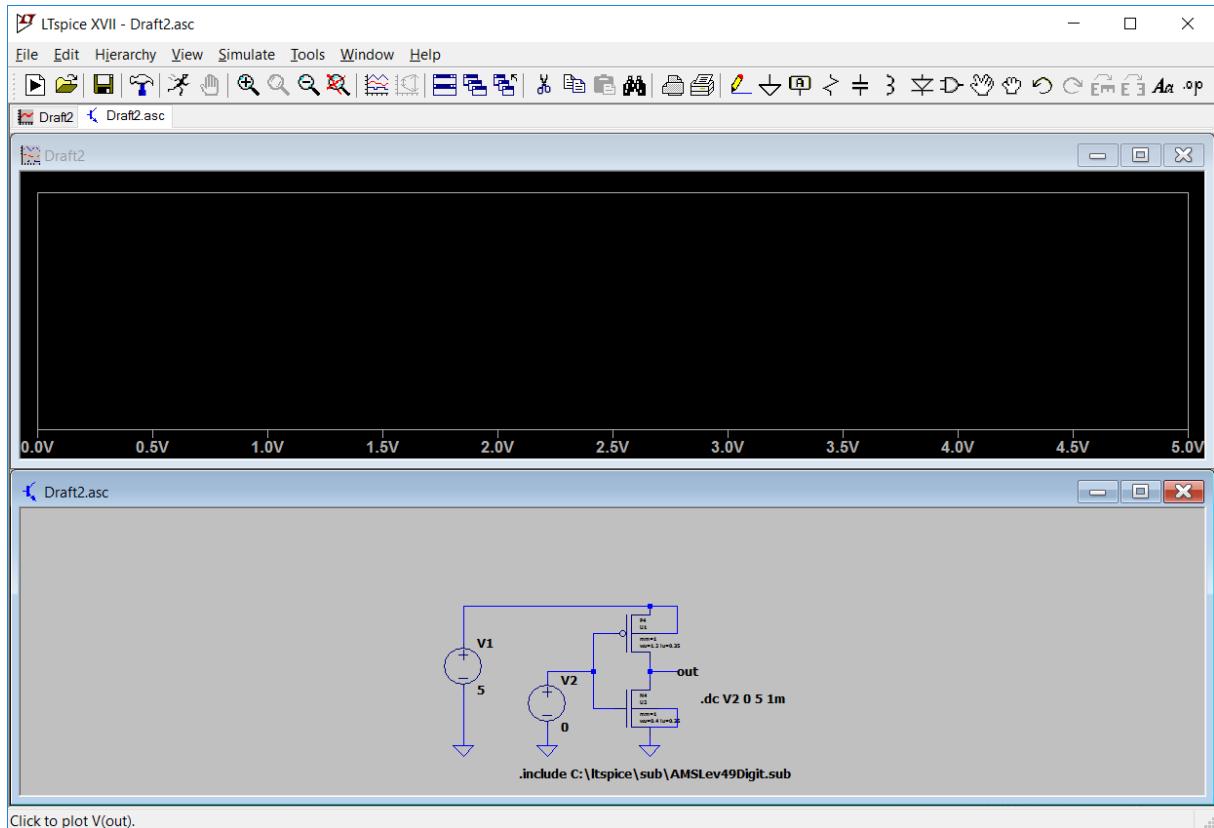


Figure 8. Simulation diagram without waveforms.

If you click on the input wire (the wire from the positive terminal of the V2 voltage source) and the output wire, you can get the DC transfer characteristics of the inverter. Please insert the screenshot of the waveforms. Read the threshold voltage (x value where the curves cross each other).

Transient simulation

First, we have to set the input source for the transient simulation. Right-click on the input voltage generator, then Advanced button, choose Pulse function, and set 200MHz, 50% duty cycle digital signal, which has a 10p second long rising and falling edge.

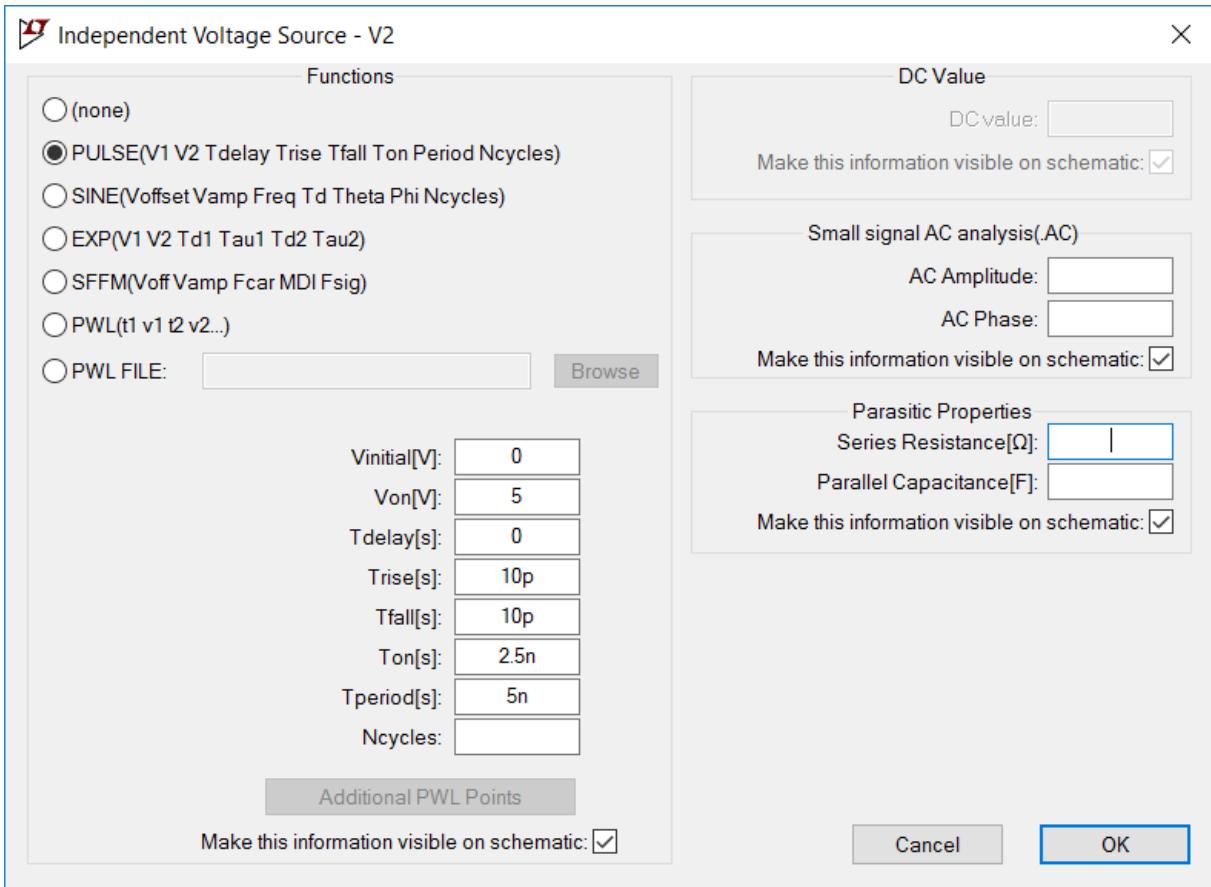


Figure 9. Voltage source settings.

Go to Simulate, Edit Simulation Cmd. Turn off the DC simulation by inserting a % character at the beginning of the simulation command.

```
%dc V2 0 5 1m
```

Choose *Transient* tab, set the Stop time to 10 nanoseconds, and the Maximum time step to 1p. (see Fig. 10)

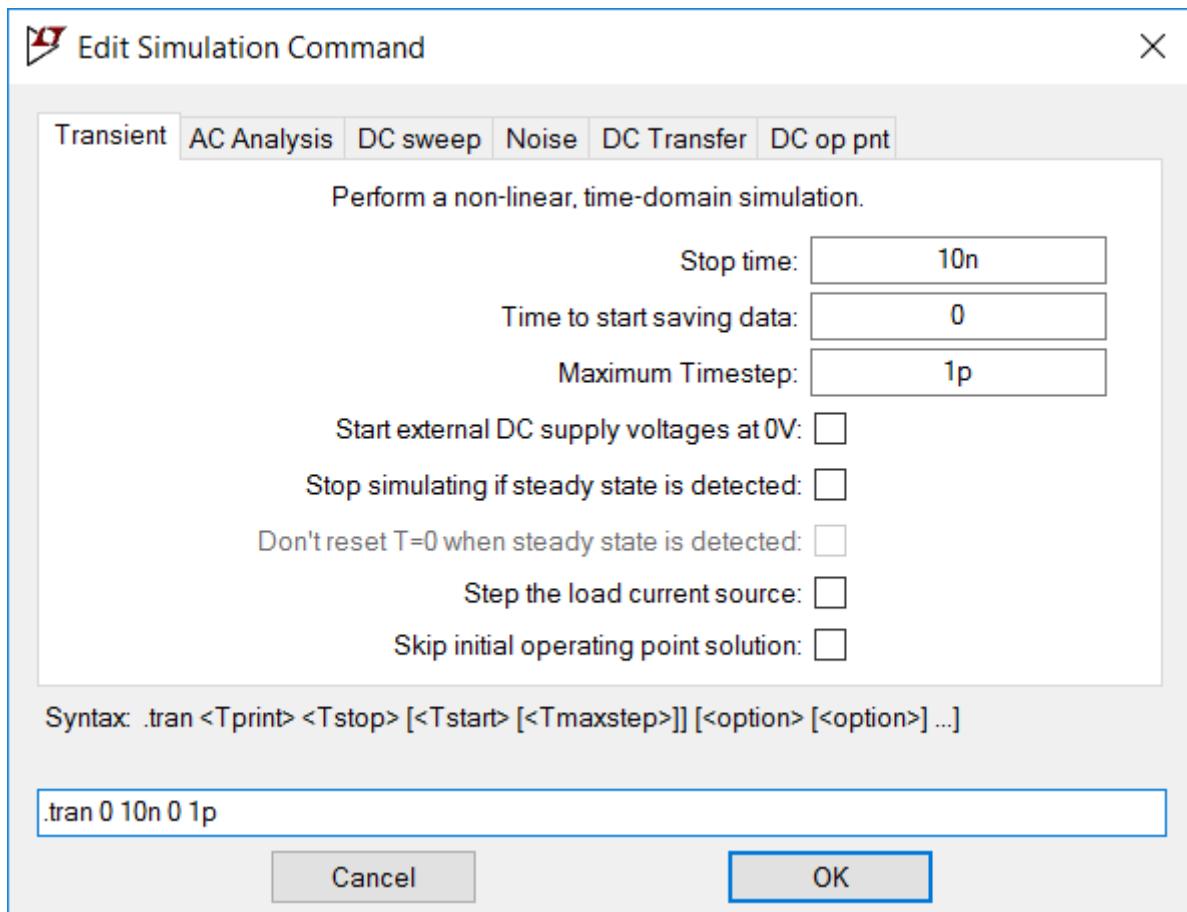


Figure 10. Transient simulation settings.

Now you can perform the simulation. If it is done, please click on input and output wires to see the waveforms.

We are going to measure the time delay for the rising edge and for the falling edge (the time difference between the 50% points). We need two cursors. To reach them, please right-click on **V(out)** and there choose 1st & 2nd (see Fig.11.). After clicking OK, you will have two cursors. You can move them when a '1' or '2' appears, and you can move them.

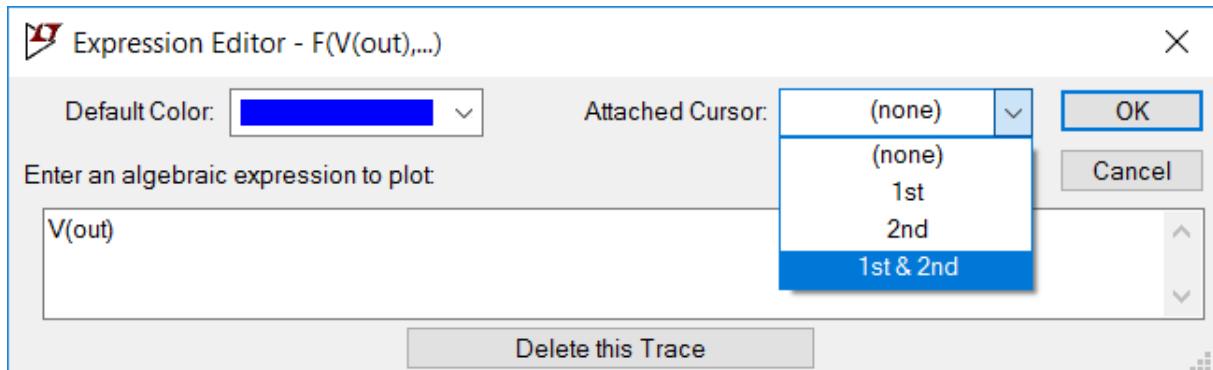


Figure 11. Adding cursors

Move one to the start, and the other to the 50% point of the output signal (2.5 V). In the draft window, you can read the time difference between the cursors (see Fig. 12.)

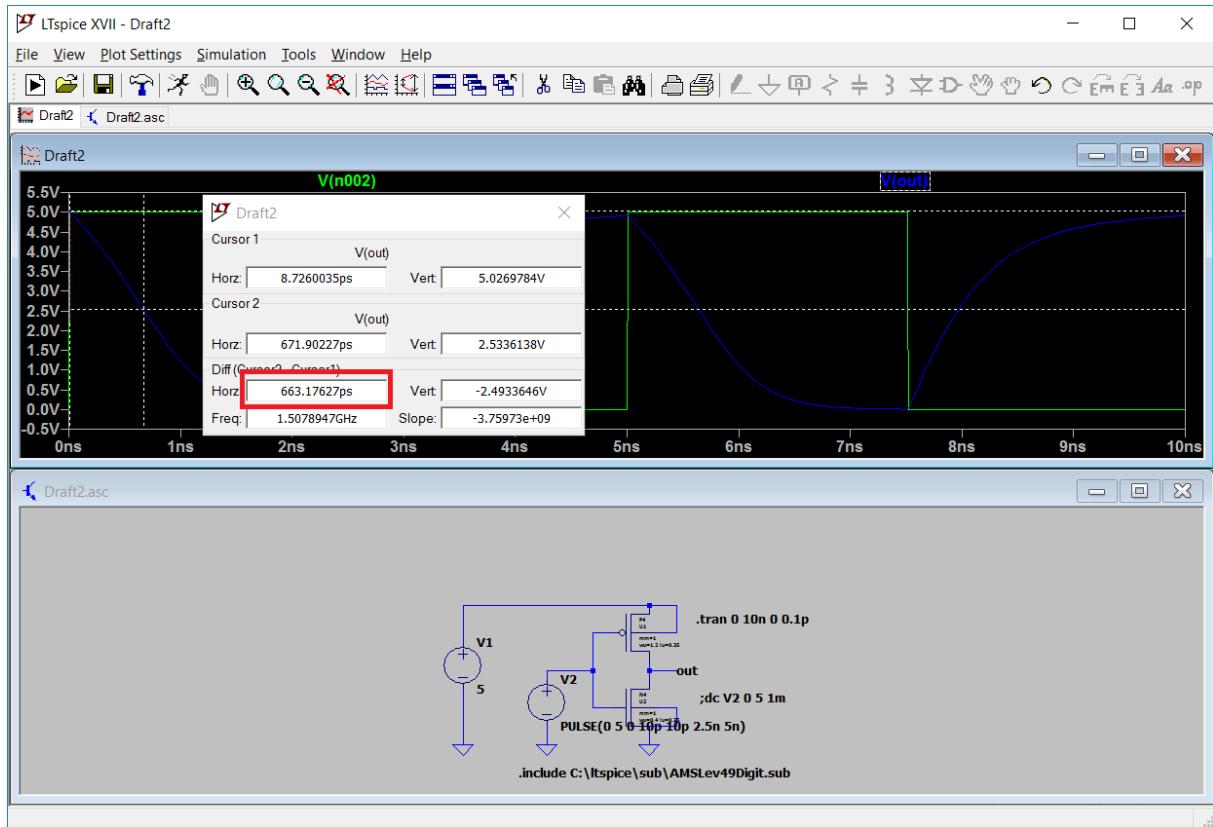


Figure 12. Reading the cursors

Please do the same for the rising edge of the output. Please include these results in the lab report.

Analog Circuit Design II.

Circuit simulation of CMOS logic gates

In this lab, we will use LTspice XVII again with the AMS 350 nm integrated technology library.

Start the circuit design tool by clicking its icon on your desktop or by using the Start menu shortcut (path: "C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe").

Use the File - New Schematic command to create a new wiring diagram. Components, wiring and simulation commands can be placed here.

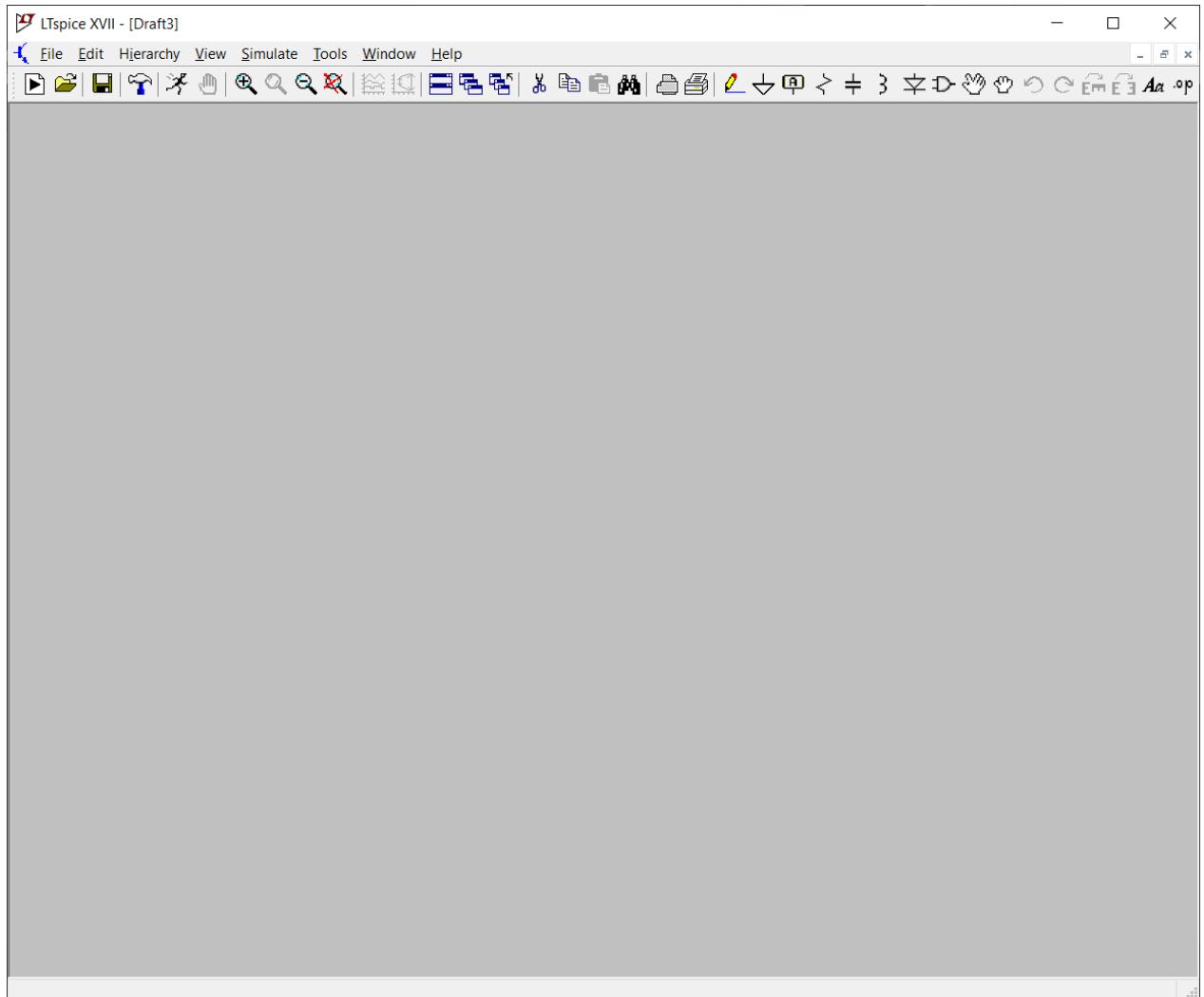


Figure 1. Schematic design window of LTspice

Click on the Edit drop-down menu to see a list of commands needed to create the schematic, with shortcuts enclosed in apostrophes. Some follow logical pattern (e.g. resistance - 'R'), but there are some interesting ones (e.g. wire - F3, undo (Ctrl + Z) - F9).

Simulation of a NAND gate

Select Edit – Component 'F2', where we can select a component from the default folder (`C:\Users\USERNAME\Documents\LTspiceXVII\lib\sym`). The folders are in square brackets, and they contain more components. Please choose `AMScellsDigit` folder, and `n4` component, and place two. (If you cannot find `AMScellsDigit` here, please try to copy the content of the .zip file into the right folder). Now go back to Edit – Component 'F2', select `AMScellsDigit` folder, and `p4` component, and place two, (Fig. 2).

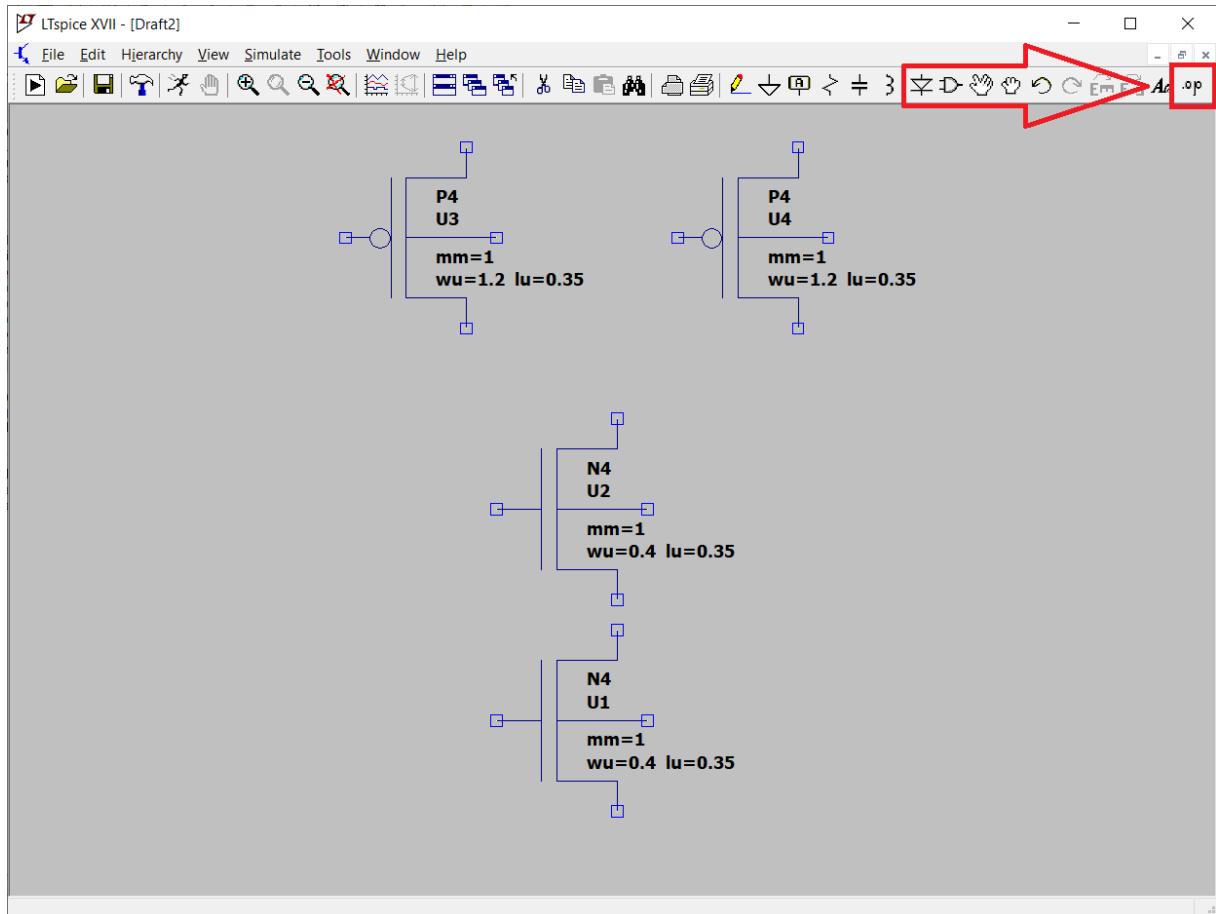


Figure 2. Placement of MOS transistors

Now we have to click on the .op button (indicated in Fig.2.) to create a SPICE Directive. Here we can define the path of the model file. Insert this line below:

```
.include c:\ltspice\sub\AMSLv49Digit.sub
```

And place it somewhere on the schematic (like in Fig.3.).

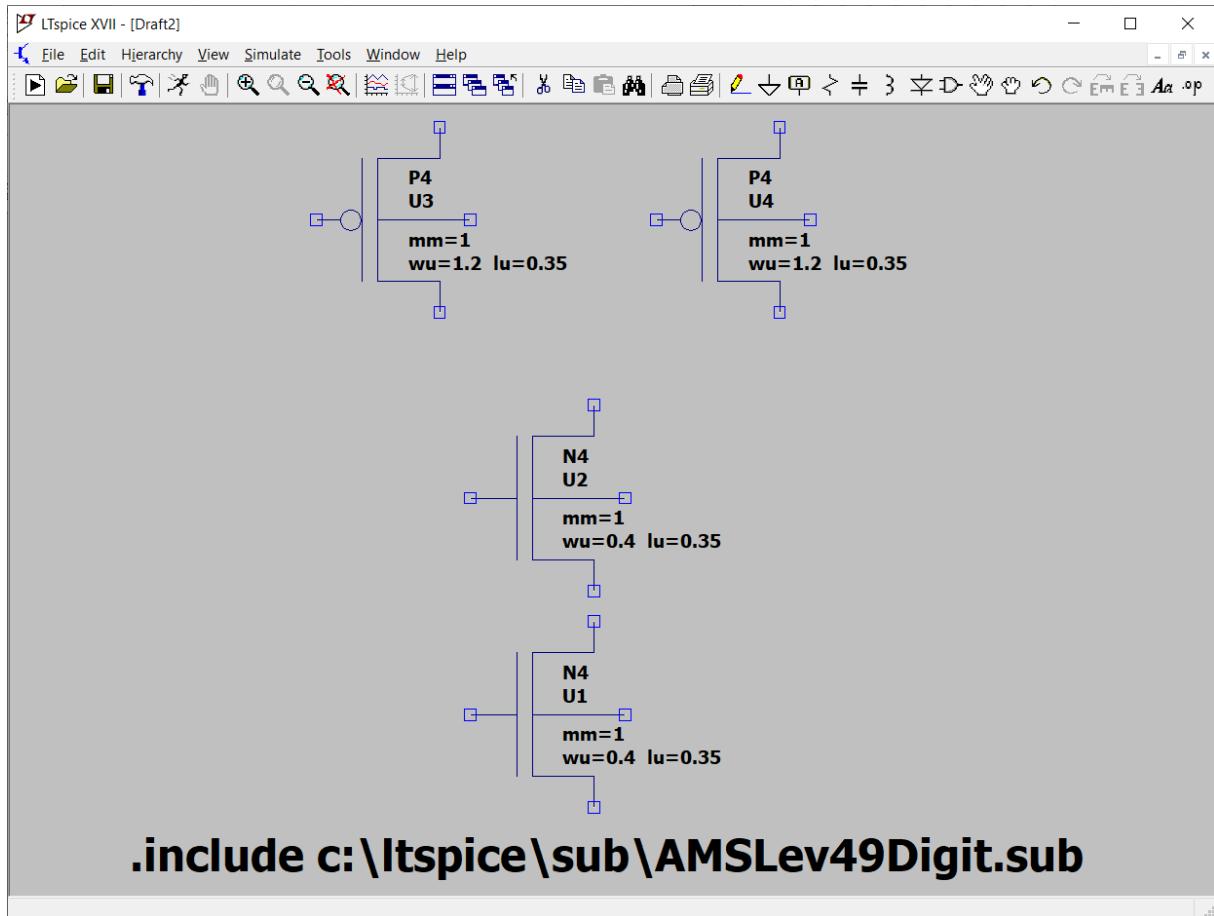


Figure 3. Adding SPICE Directive

Now we have to insert three voltage sources, one for the power supply, and two for the inputs.

From Edit – Component 'F2' choose 'voltage' (you might have to go back from a subfolder clicking on[..]), and place three of it.

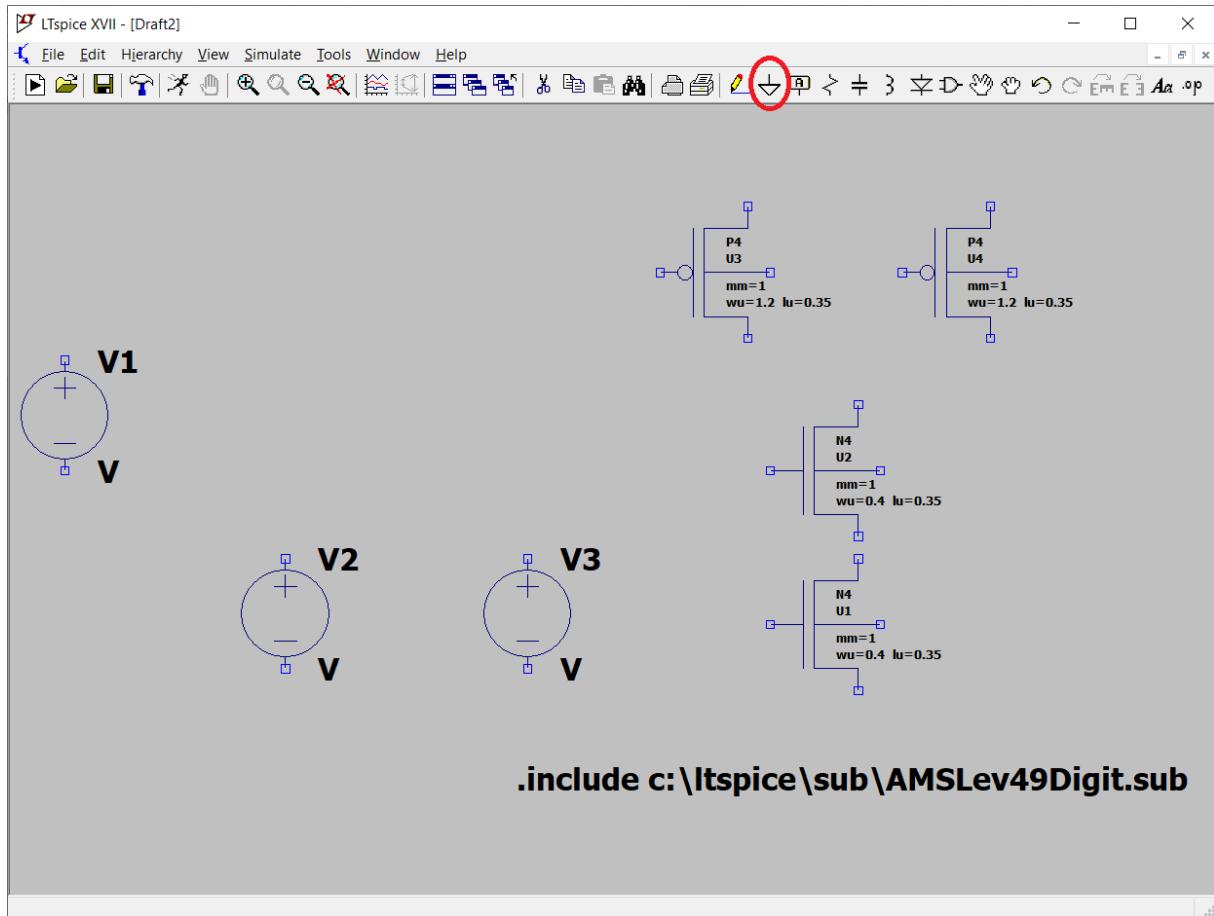


Figure 4. Insert the Grounds

Now place four Ground components (see Fig. 4.). After you finished it, add an output port to the circuit.

Select button, and Set the Port Type to Output, and write *out* into the input field.

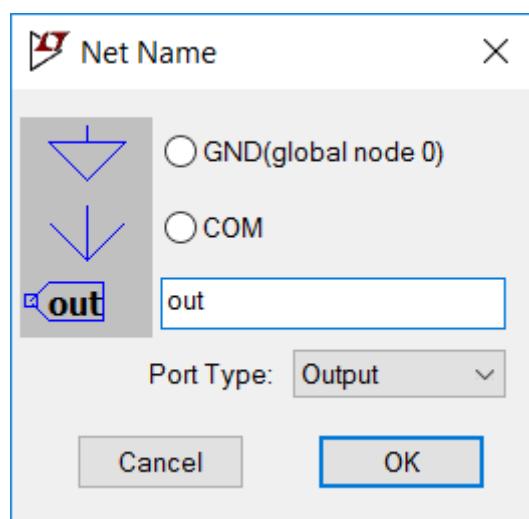


Figure 5. Naming the net

Instead of complicated wiring, we can use net labels. If the name of two wire labels is the same, the wires are (virtually) connected together.

But first, wire some part of the circuitry using Edit – Draw wire ‘F3’ command, as it can be seen in Fig. 5.

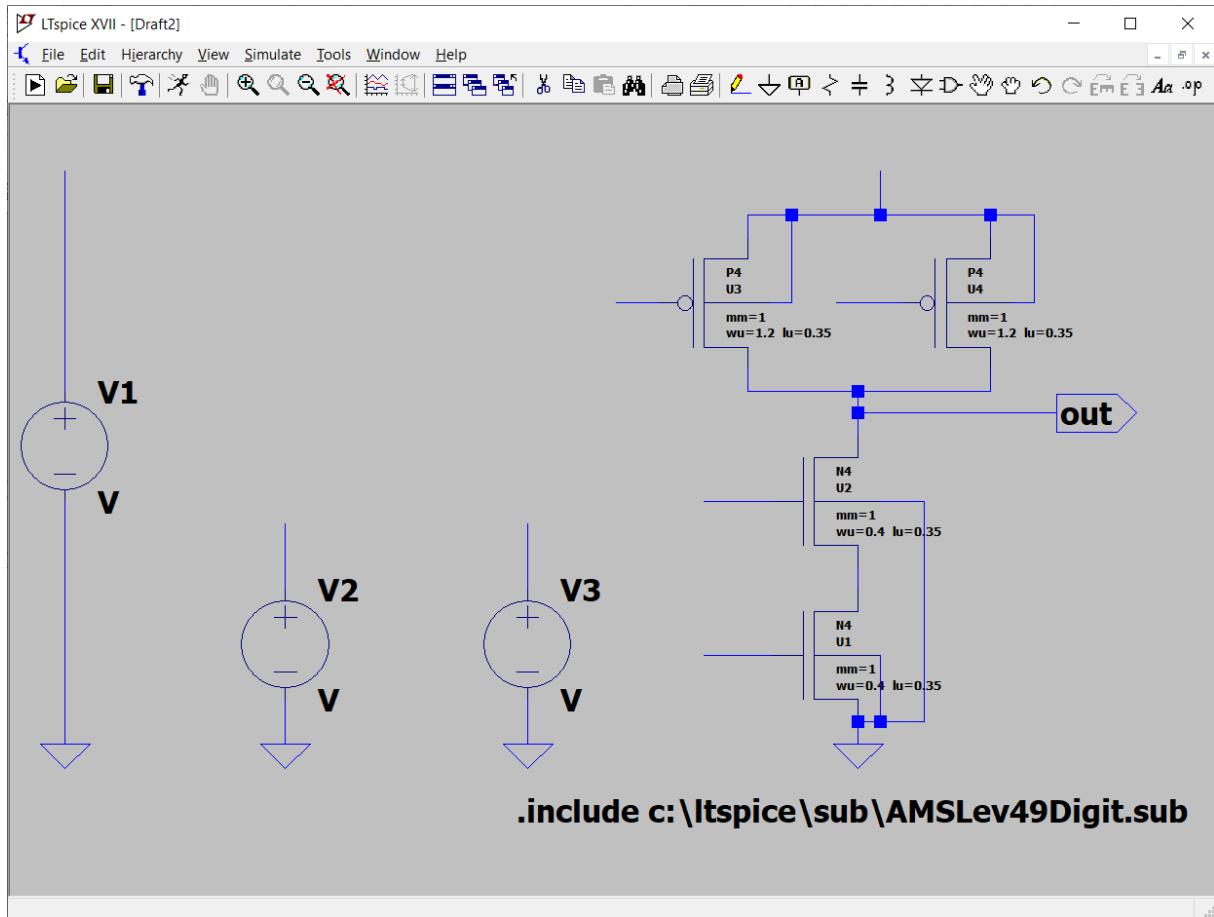


Figure 6. Wiring the schematic

Now we have to add the labels. Two *vdds* for positive power supply, three *As* for input signal A, and three *Bs* for input signal B. To do this, please click on Label Net icon

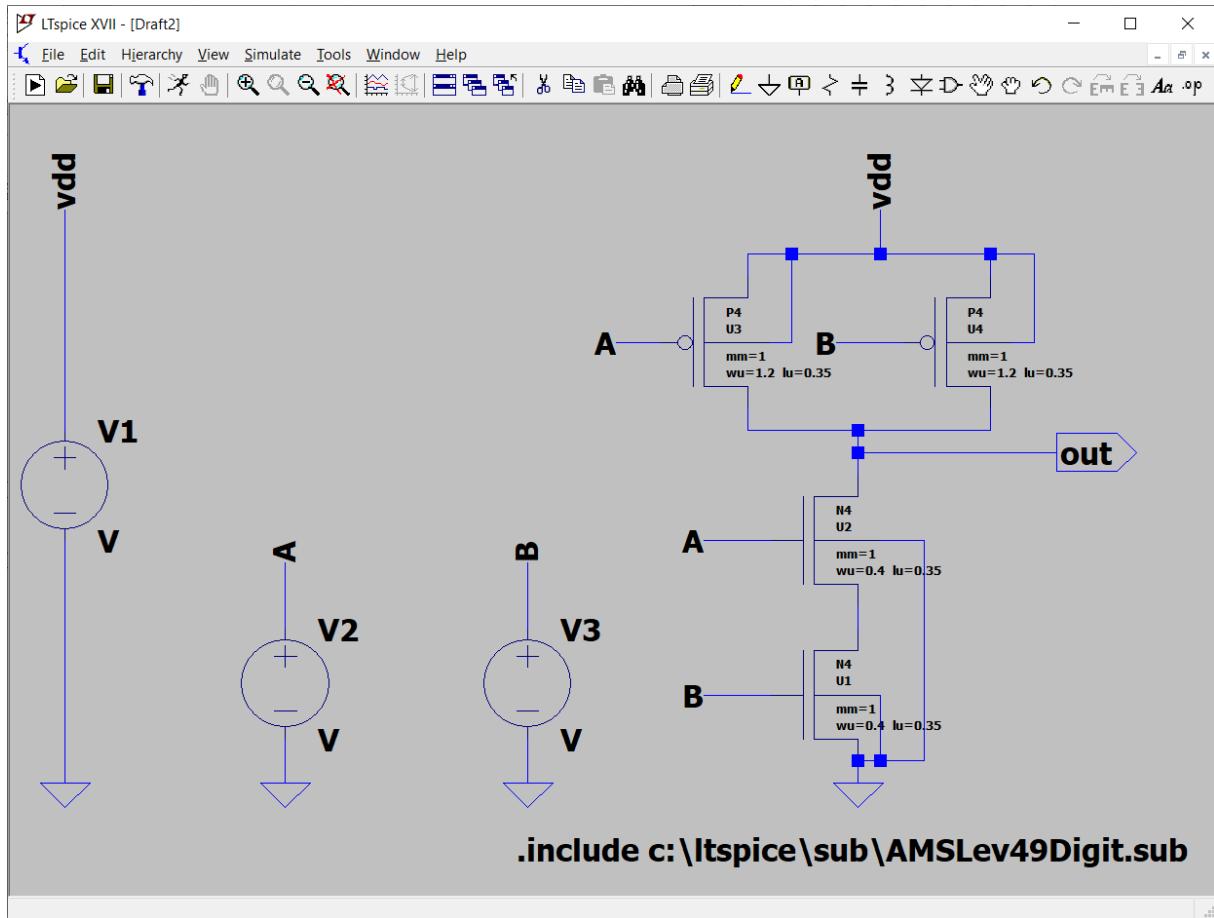


Figure 7. Wired schematic

Now we have to set the voltage of the voltage sources. We can do it by clicking the right mouse button on it. The power supply voltage (vdd) has to be 5 V (see Fig.6.).

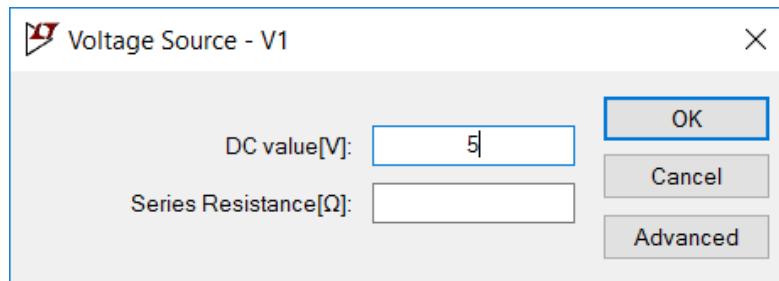


Figure 8. Power supply voltage

At this point, we need to create the input waveform. Let set A as LSB. The frequency is 100 MHz, 50% duty cycle, 5 V of amplitude, 10psec long rising and falling edge.

Right mouse click on V2 source (A signal), Advanced, Function: PULSE and set the parameters according to Figure 9.

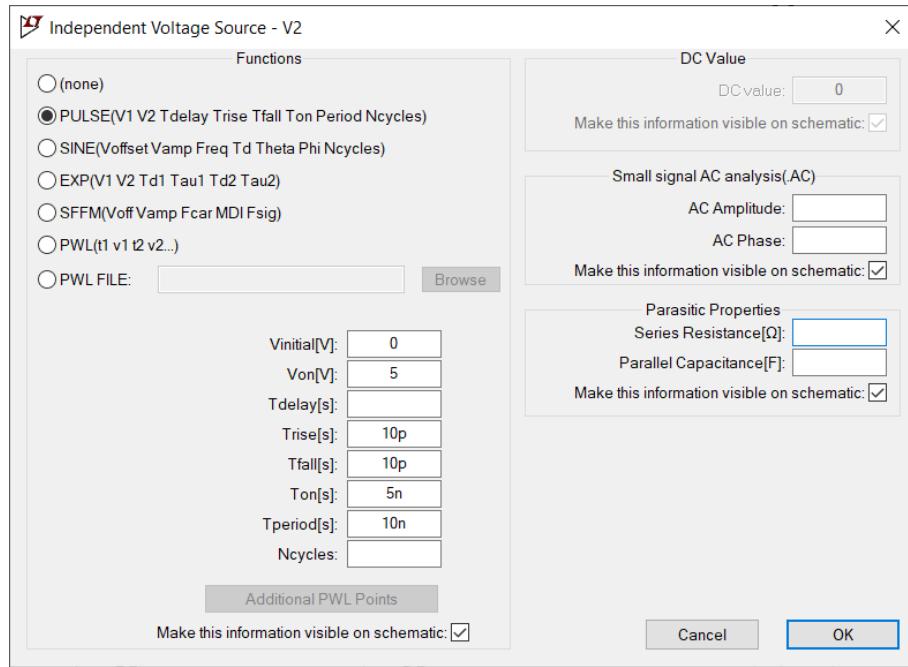


Figure 9. Settings for signal A

For signal B, the period and the T_{on} have to be double (Figure 10).

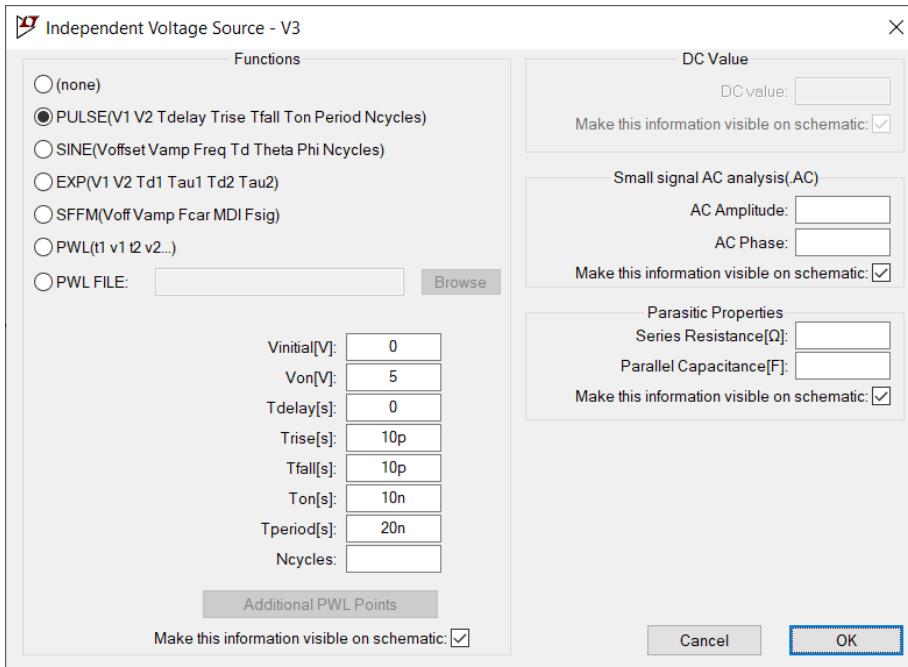


Figure 10. Settings for signal B

Transient simulation

Now we are ready for the simulation. Select Simulate – Edit Simulation Cmd, and choose Transient tab. Fill the input field, as it can be seen in Figure 11.

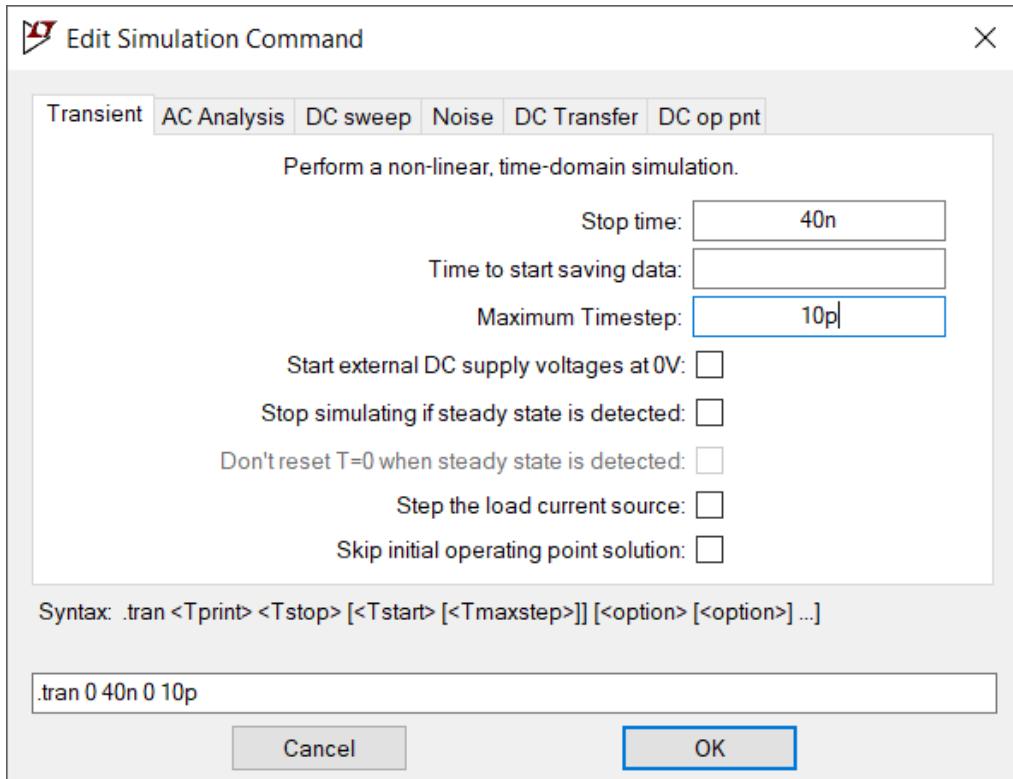


Figure 11. Transient simulation settings

If you are done, click on OK, and place the simulation command on the schematic. Select  button to perform the simulation. If everything is done, you can see an empty diagram like in Figure 12.

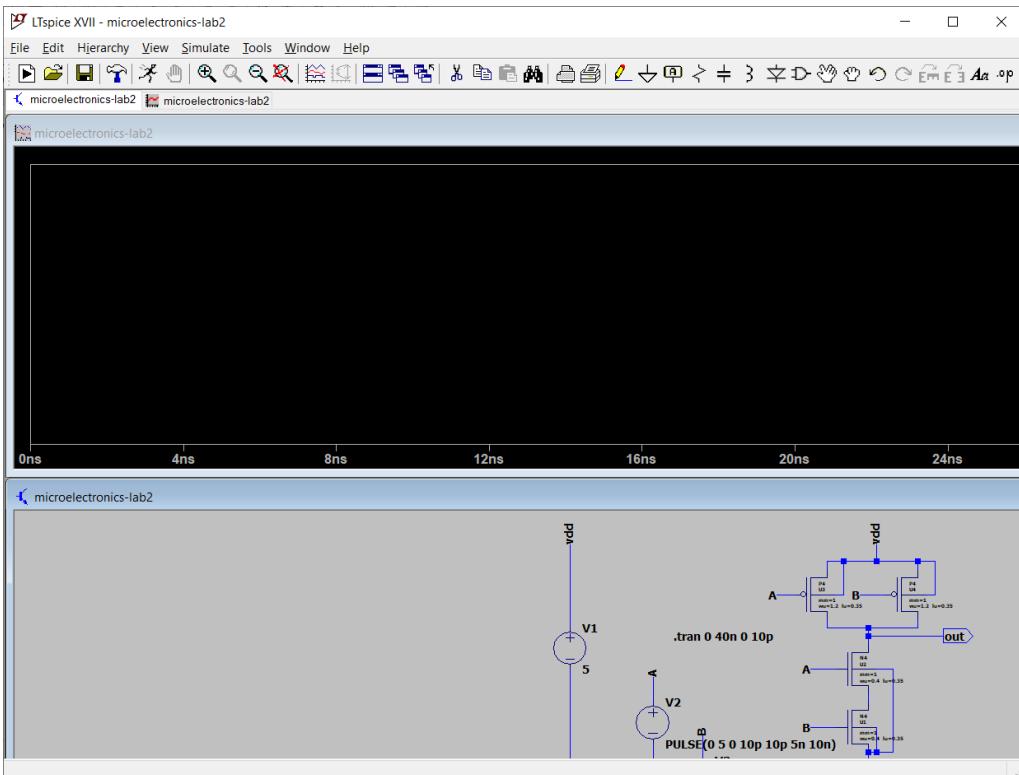


Figure 12. Simulation diagram without waveforms.

If you click on the input wires (the wire from the positive terminals of the V2 and V3 voltage sources) and the output wire, you can get the transient response of the logic gate. Please insert the screenshot of the waveforms.

Decrease the frequency of the input signal to 10 MHz (for the LSB)

- Hints:
 - There are TWO sources.
 - Don't forget to maintain the simulation stop time.

Compare the waveform with the result of the previous simulation.

Simulation and identification of a complex logic gate

Please modify the schematic (or you can start with an empty schematic) to get the same as it can be seen in Figure 13.

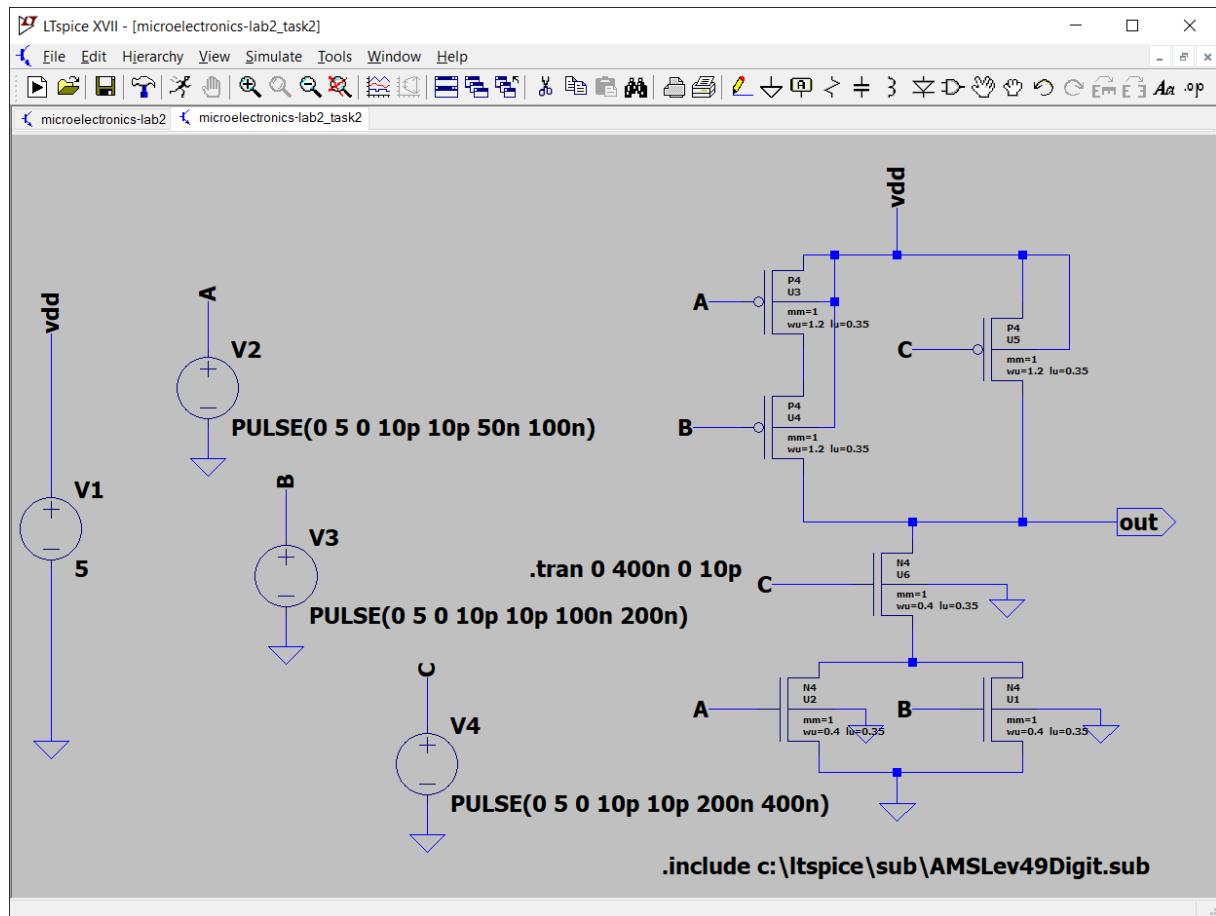


Figure 13. Schematic diagram of a CMOS complex logic gate

Perform a transient simulation with the same settings as in the previous task. Plot the inputs (A, B, and C, and the output waveforms), and insert them to the lab report.

Create a truth table. Try to find out the logic function of the complex gate.

Analog Circuit Design III.

Circuit simulation of a Common Source Amplifier

In this lab, we will use LTspice XVII again with the AMS 350 nm integrated technology library.

Start the circuit design tool by clicking its icon on your desktop or by using the Start menu shortcut (path: "C:\Program Files\LTC\ LTspiceXVII\XVIIx64.exe").

Use the File - New Schematic command to create a new wiring diagram. Components, wiring and simulation commands can be placed here.

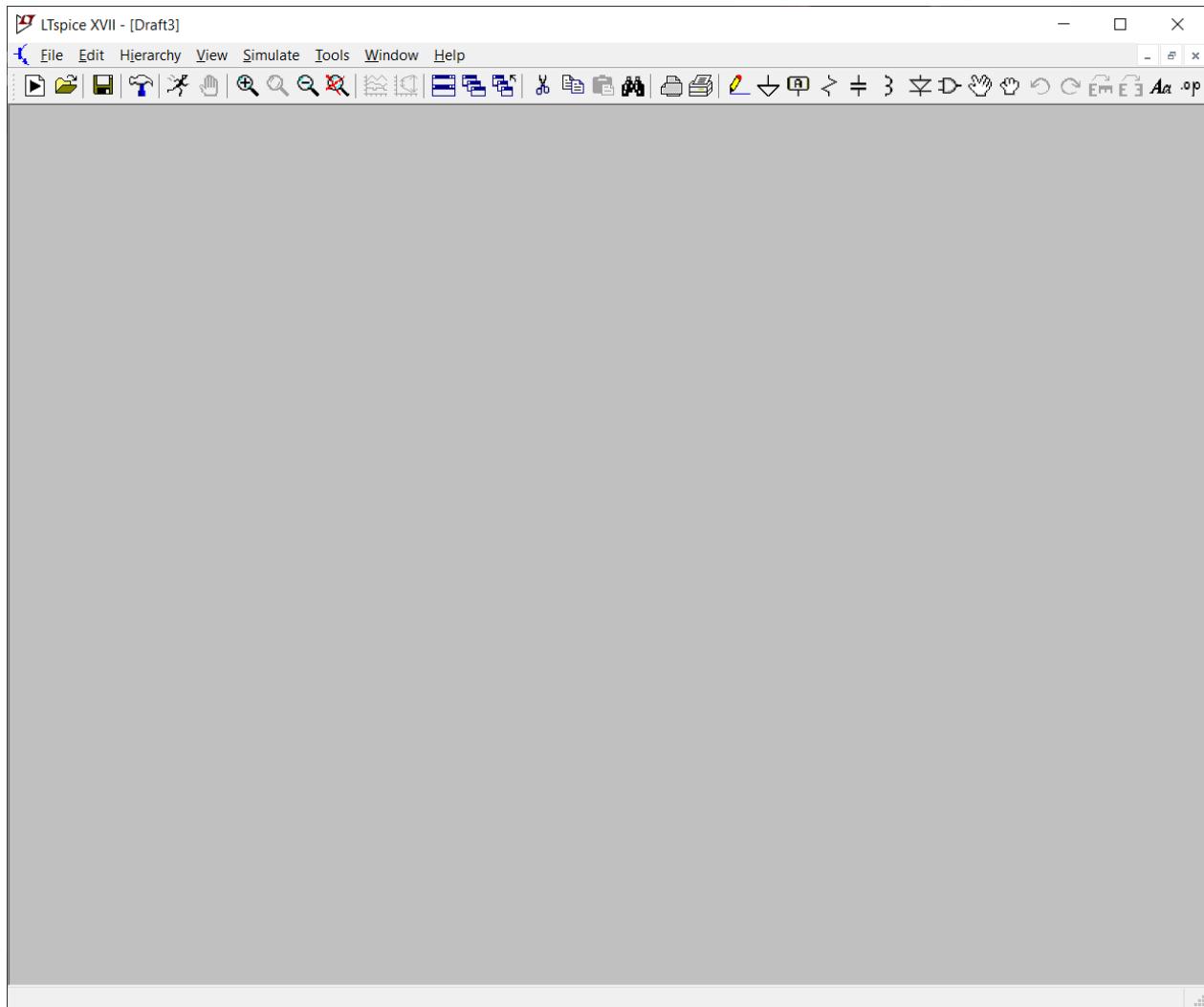


Figure 1. Schematic design window of LTspice

Click on the Edit drop-down menu to see a list of commands needed to create the schematic, with shortcuts enclosed in apostrophes. Some follow logical pattern (e.g. resistance - 'R'), but there are some interesting ones (e.g. wire - F3, undo (Ctrl + Z) - F9).

Simulation of a Common Source amplifier

Select Edit – Component ‘F2’, where we can select a component from the default folder (`C:\Users\USERNAME\Documents\LTspiceXVII\lib\sym`). The folders are in square brackets and they contain more components. Please choose `AMScellsDigit` folder, select `n4` component and place it. (If you cannot find `AMScellsDigit` here, please try to copy the content of the .zip file into the right folder).

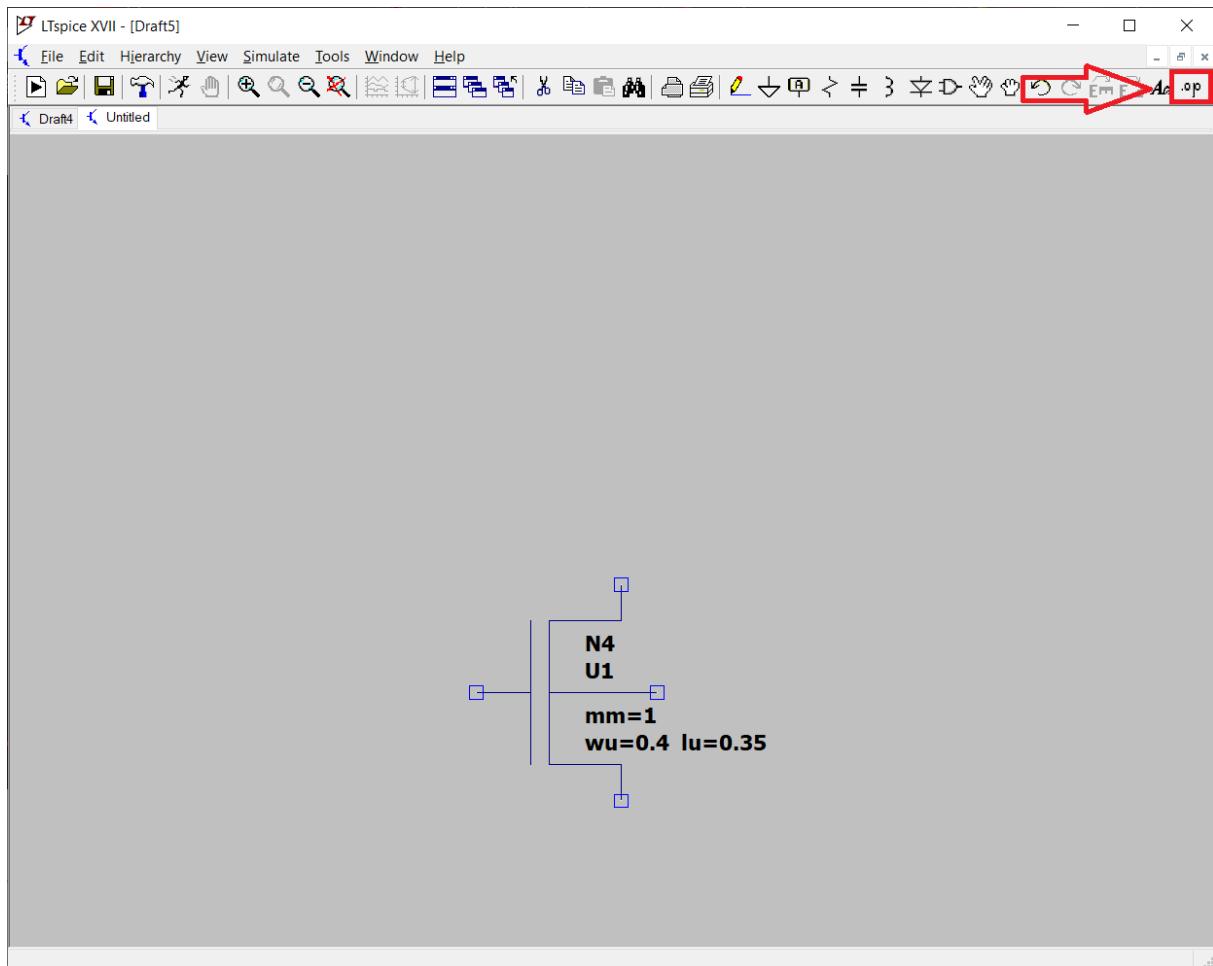


Figure 2. Placement of a MOS transistor

Now we have to click on the .op button (indicated in Figure 2) to create a SPICE Directive. Here we can define the path of the model file. Insert this line below:

```
.include c:\ltspice\sub\AMSLev49Digit.sub
```

Place it somewhere on the schematic (like in Figure 3).

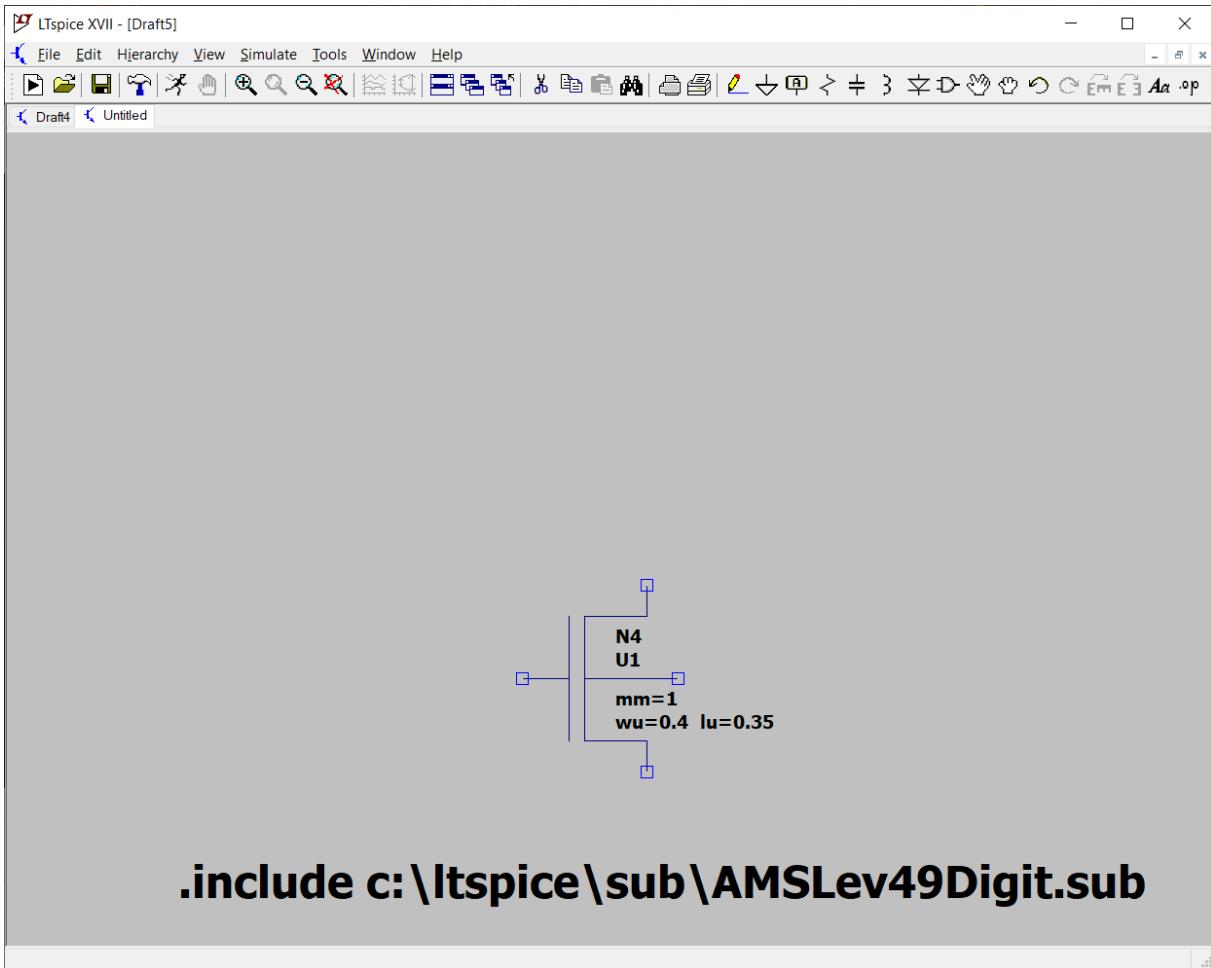


Figure 3. Adding SPICE Directive

Now we have to insert two voltage sources, one for the power supply and one for the input.

From Edit – Component 'F2' choose 'voltage' (you might have to go back from a subfolder clicking on[...]) and place two of it.

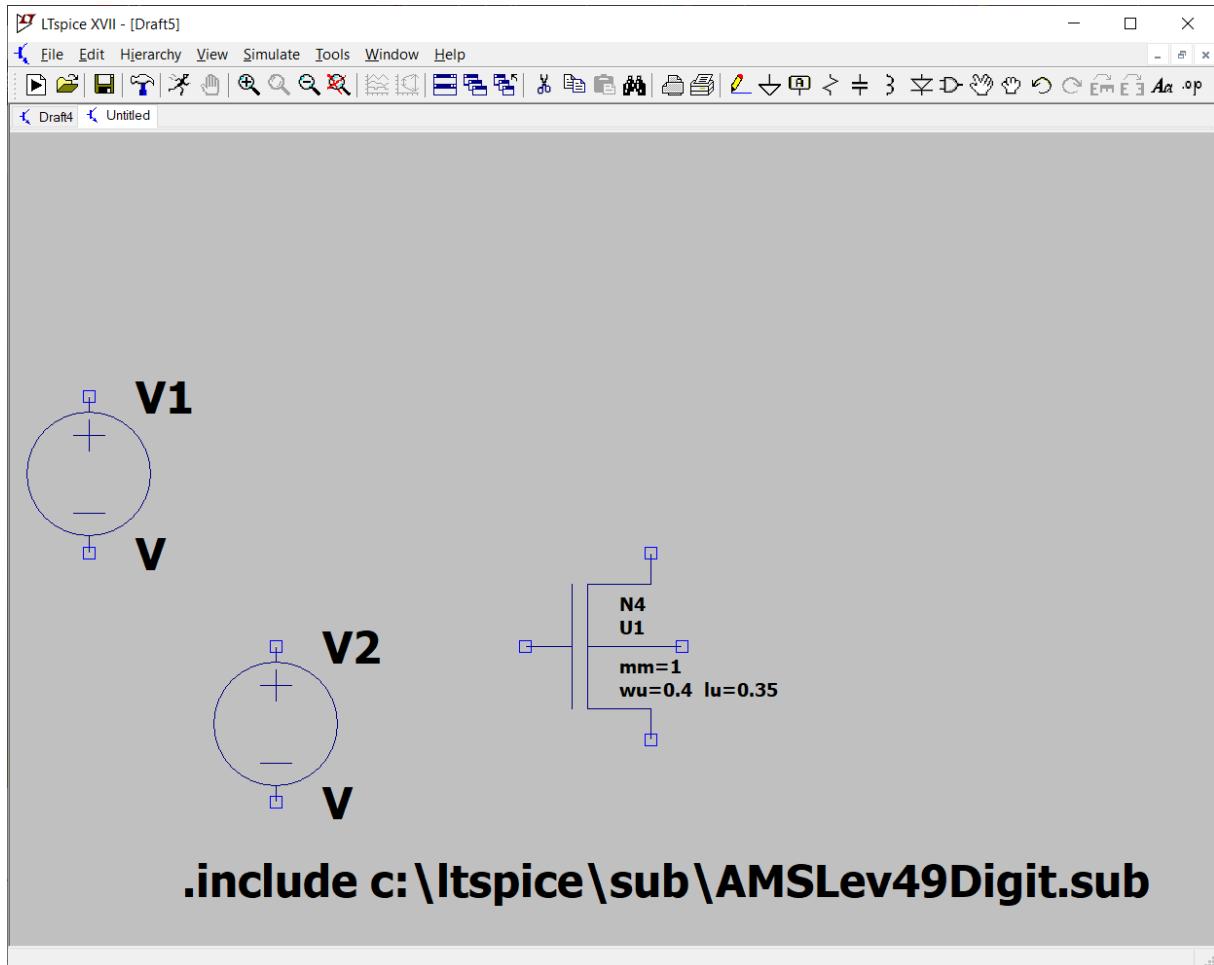


Figure 4. Insert the Grounds

Now place four Ground components (see Figure 5). After you finished it, add an output port to the circuit. Select button, Set the Port Type to Output and write *out* into the input field.

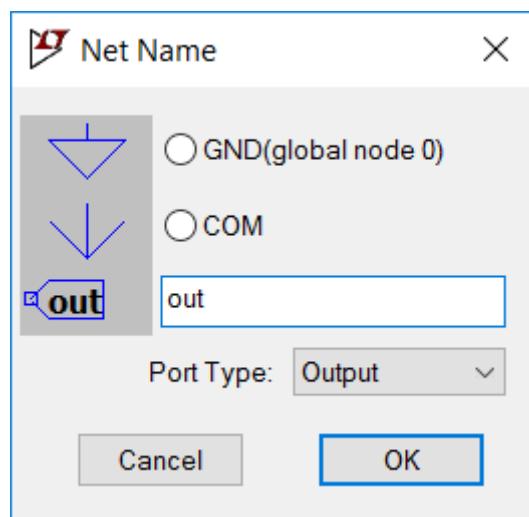


Figure 5. Naming the net

Add a resistor (F2 – res), and wire the circuitry using Edit – Draw wire ‘F3’ command, as it can be seen in Figure 6.

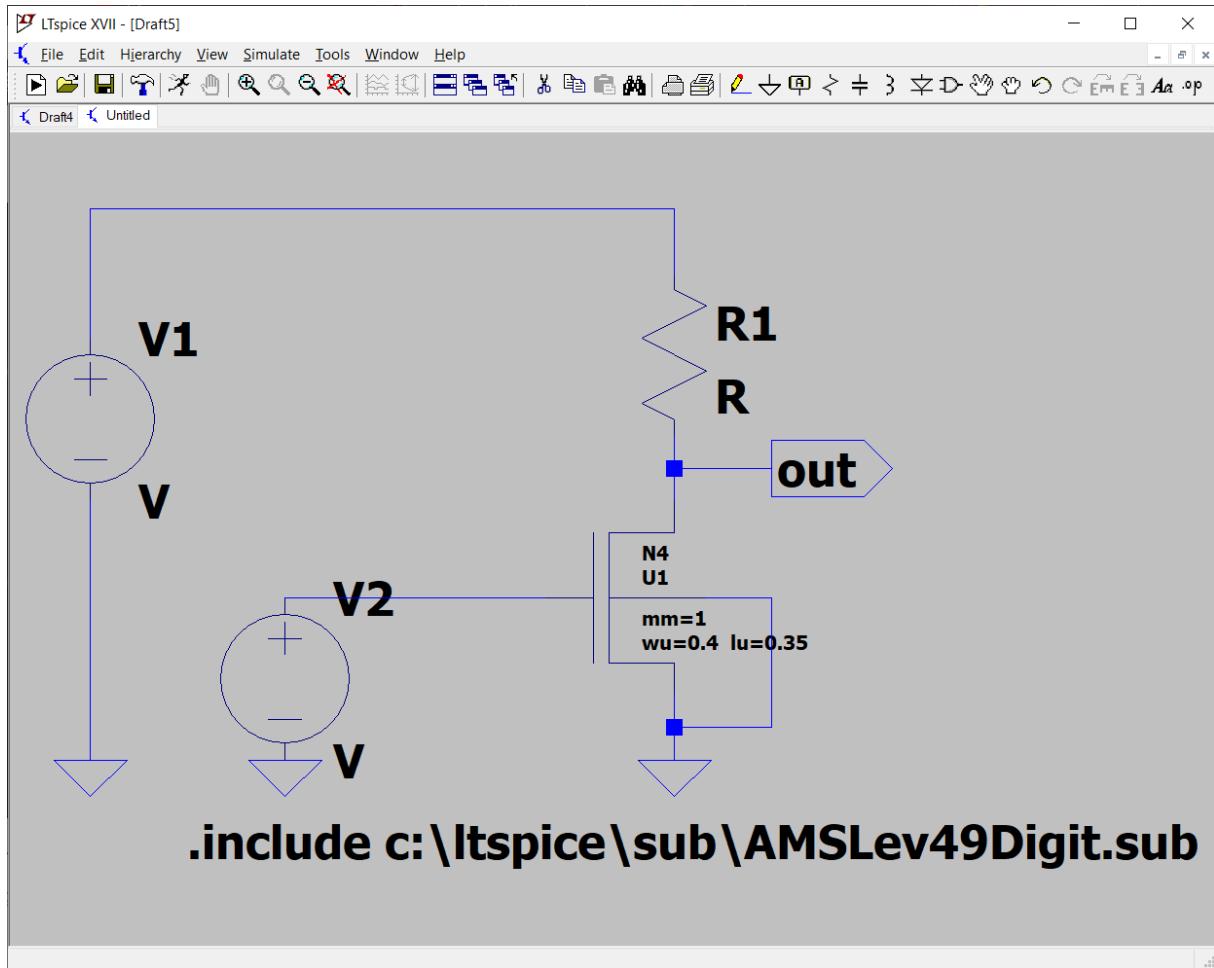


Figure 6. Wiring the schematic

Now we have to add a label for the input wire. To do this, please click on Label Net icon , type 'in' and place it.

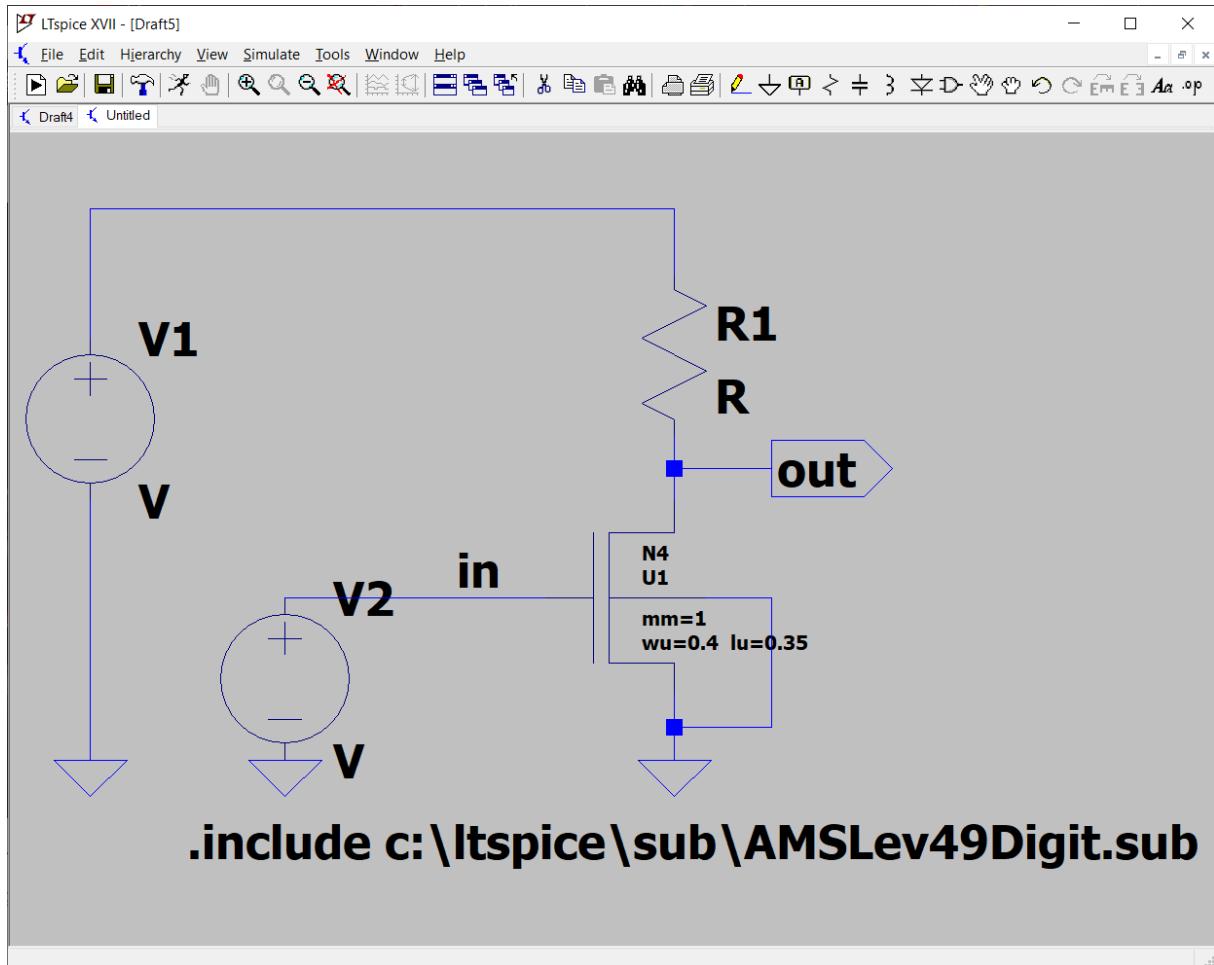


Figure 7. Wired schematic

Now we have to set the voltage of the voltage sources, the resistance of the resistor and the channel dimensions of the MOS transistor. We can do it by clicking the right mouse button on the selected component. The power supply voltage (**V1**) has to be 5 V (see Figure 8.).

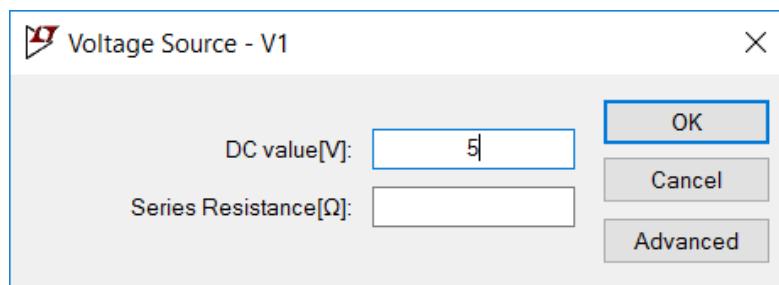


Figure 8. Power supply voltage

For **V2**, please set 0 Volts (we will modify it later), let $R=10k$ and the channel dimension $wu=1$, $lu=1$ (see Figure 9).

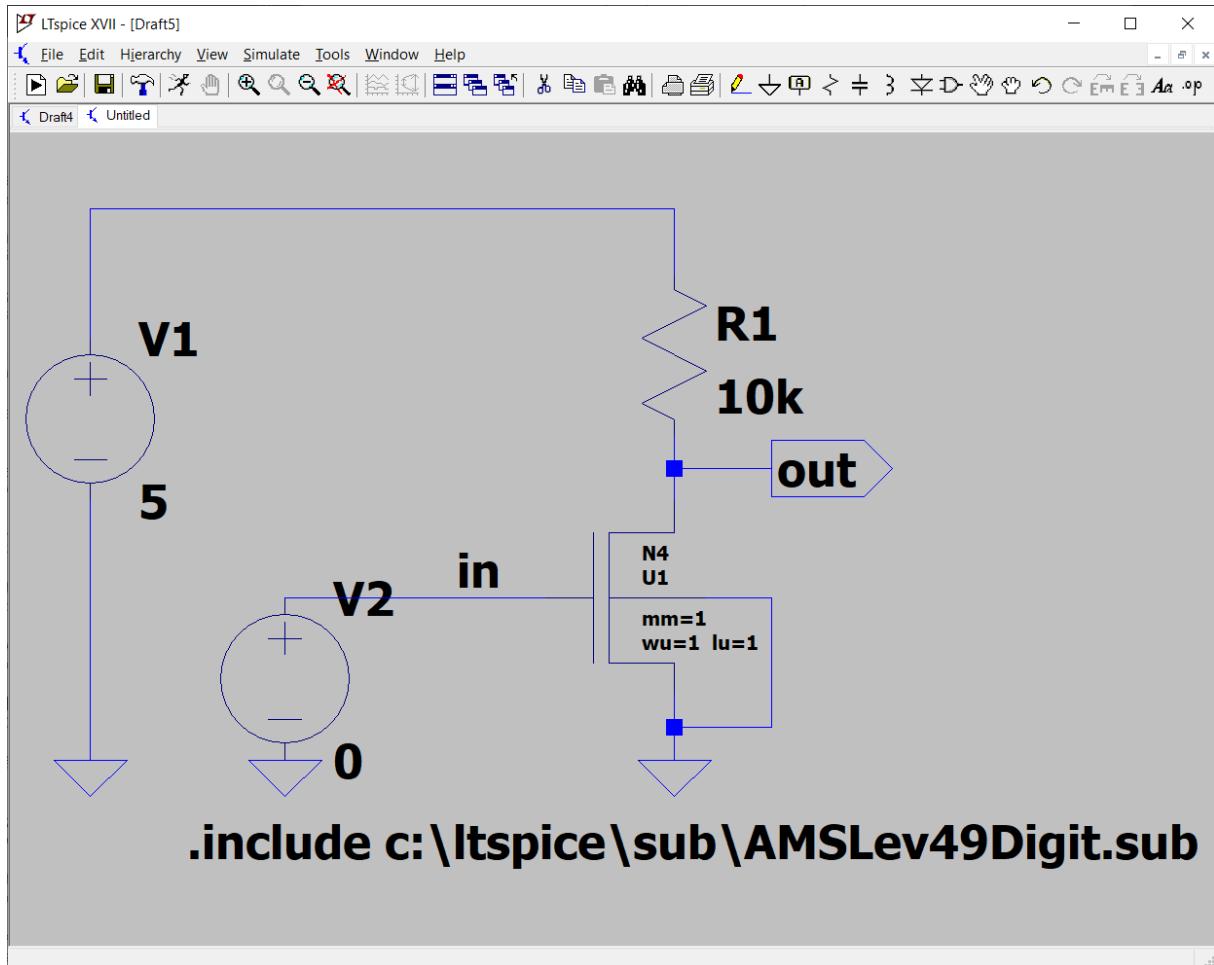


Figure 9. Schematics with component parameters

DC sweep simulation

Now we are ready for the simulation. Select Simulate – Edit Simulation Cmd, and choose DC sweep tab. Fill the input field, as it can be seen in Figure 10.

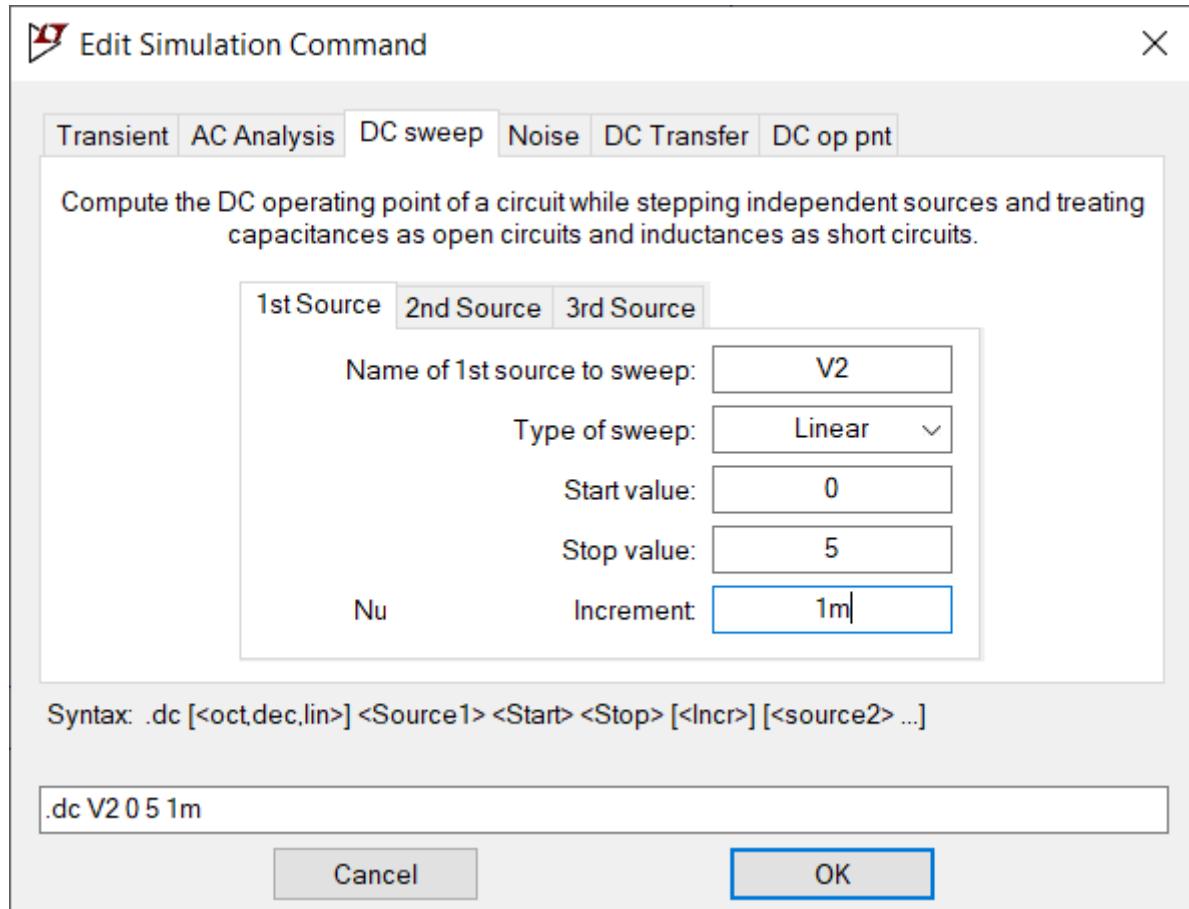


Figure 10. Simulation settings

If you are done, click on OK, and place the simulation command on the schematic. Select  button to perform the simulation. If everything is done, you can see an empty diagram like in Figure 11.

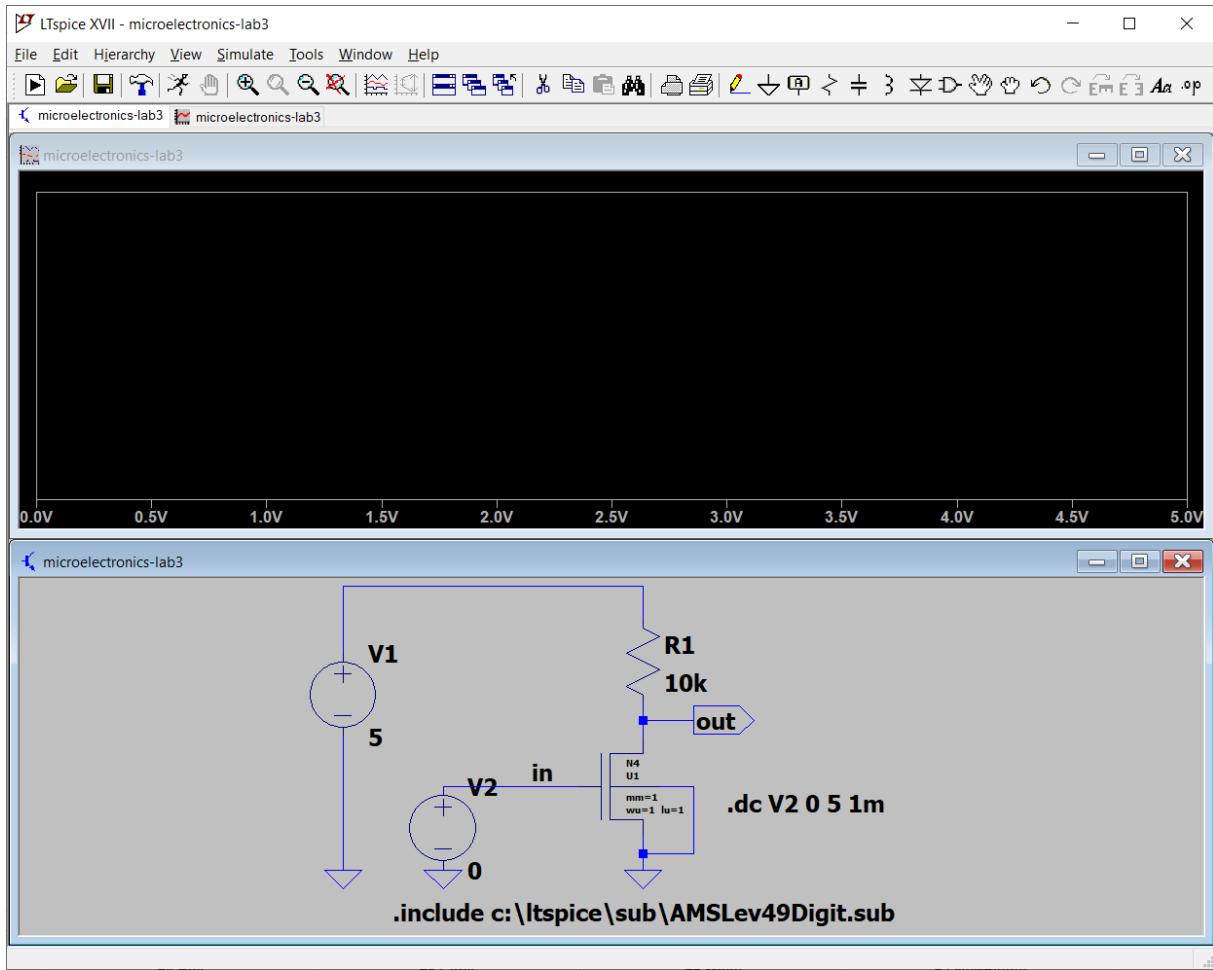


Figure 11. Simulation diagram without waveforms.

If you click on the input and output wires (the wire from the positive terminals of the V2 called 'in'), and the output wire. Please insert the graphs into the lab report.

As you can see, the output voltage cannot reach the 0 Volts, and the slope of the curve is not steep enough. There are two ways to make it better: 1) decreasing the drain current by increasing the resistance of the resistor, 2) increasing the transconductance of the transistor by increasing the width of the channel. Modify the resistance to 50k and the channel width (wu) to 10. Resimulate it. Compare the result with the previous one.

Finding the operating point

Use the cursor to find the operating point when the output node is at half of the power supply voltage (2.5 Volts). Include this value into the lab report, and DC value of V2 voltage source to this value.

AC simulation

First, please deactivate the DC sweep simulation by inserting a semicolon ';' to the beginning of the simulation command (right mouse click on the command to modify). Then, Simulate – Edit simulation cmd, select AC Analysis tab, and set the values as in Figure 12.

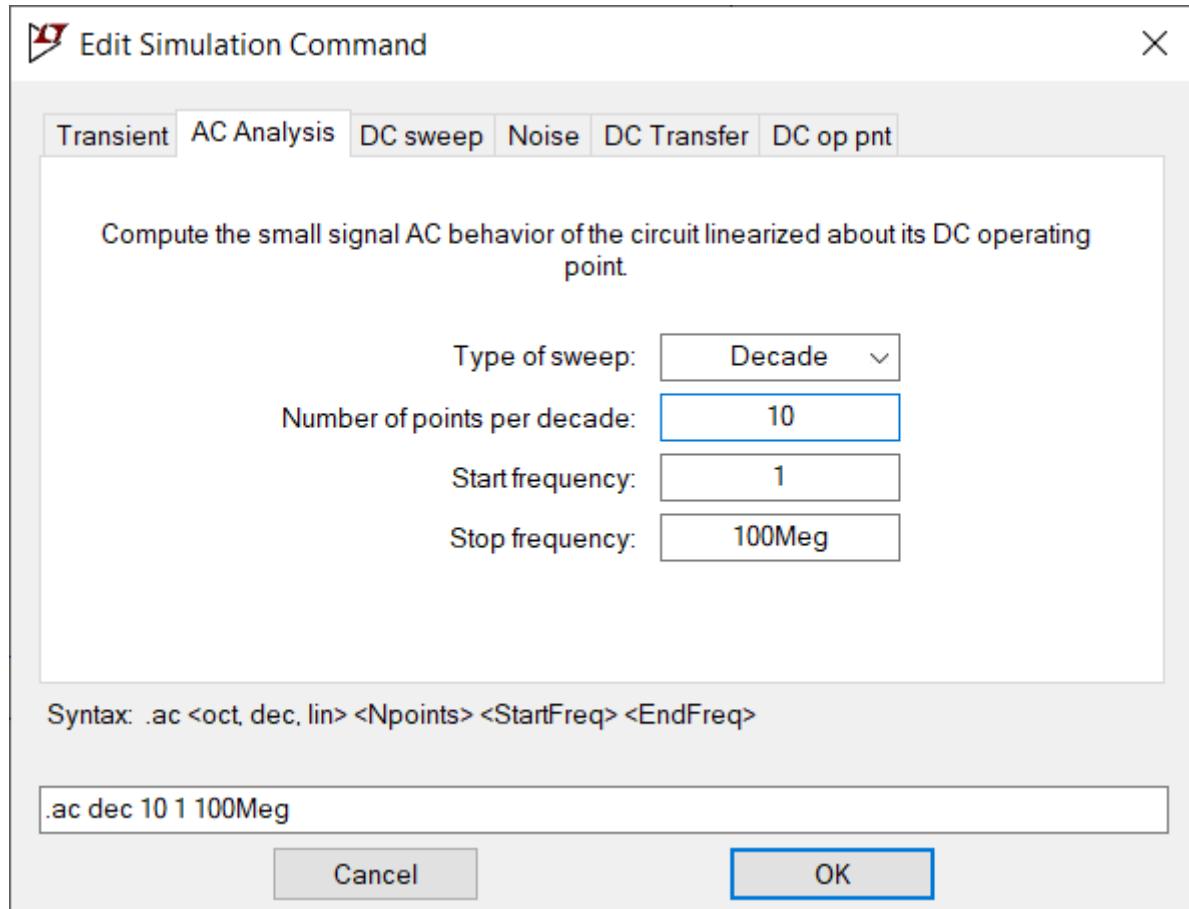


Figure 12. AC simulation settings.

After that, please right mouse click on V2 voltage source, then Advanced and set AC Amplitude to 1. Now please perform the simulation. Select the output port to get the Bode-plot. Please insert it to the lab report, and read the amplification of the amplifier.

Increasing the amplification of the amplifier

In this task, please try to increase the amplification of the amplifier. You can modify the resistance of the resistor and the channel dimensions but do not exceed the following limits:

- 1) The maximum resistance is 100k
- 2) The maximum transistor area (product of the 'wu' and 'lu') 1000
- 3) The minimum channel length is 1

Hint: if you modify at least one of the parameters listed above, the operating point will be different, so first, you have to perform a DC simulation to find the proper DC value of the V2 voltage source.

Include the Bode-plot into the lab report, and indicate the amplification of the amplifier.



Budapest University of Technology and Economics
Department of Electron Devices

BSc Course in Microelectronics

Laboratory Practice: CMOS circuit design and simulation

- Read through this summary carefully and answer the questions listed on the last page (similar questions are expected at the entry exam)
- Learn about how to use the simulator environment. Use this manual in the lab practice to help yourself setting up the simulator.

Introduction to field effect transistors

The metal–oxide–semiconductor field-effect transistor (MOSFET, MOS-FET, or MOS FET) is a transistor used for amplifying or switching electronic signals. Unlike the bipolar junction transistor, the MOSFET is a unipolar device.

The MOSFET is a four-terminal device with source (S), gate (G), drain (D), bulk (B). In practical applications the bulk is shortened to the source, therefore the bulk is not shown on schematic symbols. The MOSFET is by far the most common transistor in both digital and analog circuits, though the bipolar junction transistor was at one time much more common.

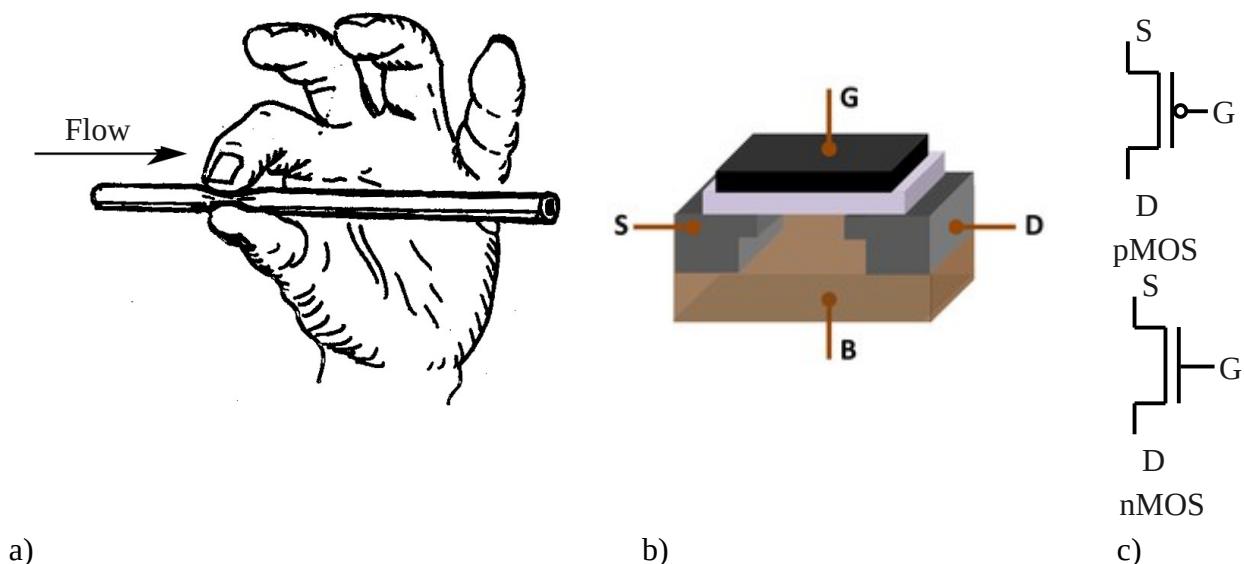


Fig 1. a) understanding the field-effect b) cross section of a MOSFET transistor, c) schematic symbols of pMOS and nMOS transistors

In enhancement mode MOSFETs, a voltage drop across the oxide induces a conducting channel between the source and drain contacts via the field effect. The term "enhancement mode" refers to the increase of conductivity with increase in oxide electric field that adds carriers to the channel, also referred to as the inversion layer. The channel can contain electrons (called an nMOSFET or nMOS), or holes (called a pMOSFET or pMOS), opposite in type to the substrate, so nMOS is made with a p-type substrate, and pMOS with an n-type substrate.

The operation can be imagined by a straw tightened by two fingers. The flow-through rate can be affected by the force of tightening. In field effect transistors the channel can be closed or opened by applying external forces as well, though the external force is the voltage applied to the gate electrode. Therefore the electric current flowing through the channel (source to drain) is affected by the gate voltage applied (drain current is controlled by gate voltage). Note, that in bipolar junction transistors the current flowing through the device (from emitter to collector) is affected by the base current (collector current is controlled by base current).

MOSFETs in digital circuits

MOSFETs are commonly used in digital circuits. For investigating the digital operation, the following rules may apply:

- In switching mode (in digital circuits) only two states of the transistor are utilized: the channel conduct, when we say 'the transistor is opened', or the channel does not conduct, when we say 'the transistor is closed'.
- Binary values are corresponding voltage values.
E.g. binary 1 means 5 V, binary 0 means 0 V.
- From this aspect the operation of the nMOS and pMOS are the opposite.
- The nMOS transistor is normally closed, and opens when a positive voltage (e.g. 5 V) is applied to the gate electrode. When the gate voltage is 0 V, the nMOS transistor is closed.
- However the pMOS transistor closes when a positive voltage (e.g. 5 V) is applied to the gate electrode, but the channel is opened otherwise (e.g. the gate voltage is 0 V).
- Schematically, when the transistor is opened, it can be substituted by a short. When the transistor is closed, it can be substituted by an open.

Digital circuits are commonly built of using both pMOS and nMOS transistors. This type of digital circuits is called cMOS (means complementary MOS). A cMOS circuitry consists of a pMOS circuit block connected to the power supply (V_{dd}) and an nMOS circuit block connected to the ground (V_{ss}). Two important characteristics of CMOS devices are high noise immunity and low static power consumption. Since one transistor of the pair is always off, the series combination draws significant power only momentarily during switching between on and off states. Consequently, CMOS devices do not produce as much waste heat as other forms of logic, for example transistor-transistor logic (TTL) or NMOS logic, which normally have some standing current even when not changing state. CMOS also allows a high density of logic functions on a chip. It was primarily for this reason that CMOS became the most used technology to be implemented in VLSI chips.

CMOS inverter

The simplest digital circuit is the inverter. An inverter has an input and an output. The output is always the opposite value of the input. The figure above describes the way of operation. When the input is '1' (gate voltages are e.g. 5 V) the pMOS closes and the nMOS opens, therefore the output is shorted to the ground. The output voltage equals to the ground potential, the digital value is '0'. If the input is '0' (gate voltages are 0 V) the pMOS opens and the nMOS closes. The output is shorted to the power supply, therefore the output voltage refers to '1' (e.g. 5 V).

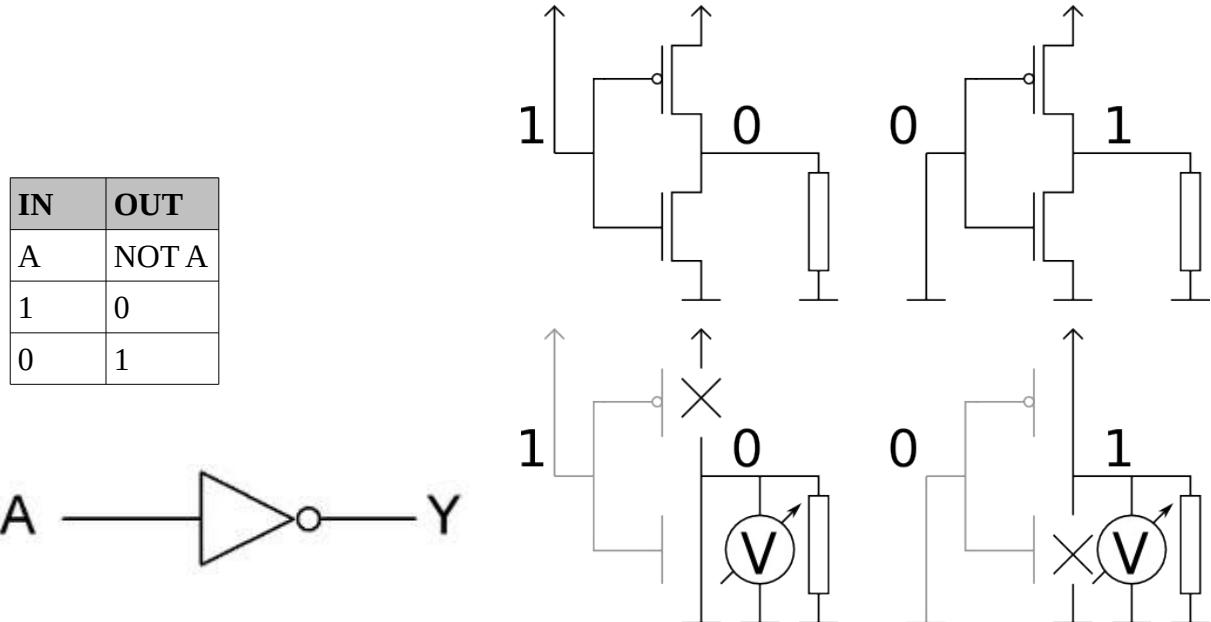


Fig 2. The inverter operation

Questions

1. What MOSFET stands for? What is the difference between nMOS and pMOS? How the terminals are called?
2. Describe the main differences between a bipolar junction transistor and a field effect transistor.
3. What is cMOS? What are the main advantages of cMOS circuits?
4. *Collect information from the internet:* What is the minimal feature size (MFS) for a modern CPU today?
5. Describe the operation of a cMOS inverter, if the input is 1 (/ 0).
6. How an electric circuit is being verified by simulation? Describe the steps shortly!
7. What are the requirements to run a simulation?

Simulation of a cMOS inverter using CAD software¹

Conventions Used

There will be several conventions used in this manual. The mouse has only two buttons but the scrolling wheel between them can be depressed, too, this will be referred to as the middle button. In the following there is some terminology explained which will be used in relation to mouse operations.

<i>click left</i>	quickly press and release the left mouse button
<i>click middle</i>	quickly press and release the middle mouse button (scrolling wheel)
<i>click right</i>	quickly press and release the right mouse button
<i>drag left</i>	press and hold the left mouse button while moving the mouse
<i>drag middle</i>	press and hold the middle mouse button while moving the mouse
<i>drag right</i>	press and hold the right mouse button while moving the mouse

If more than one OPUS window is open then the relevant window will be specified by adding WWW: for the window WWW.

If a double target xxx->yyy is specified with clicking, that may happen to be two separate clicks at xxx and yyy or a drag from xxx to yyy, depending upon how the popup menu for yyy comes up.

<...> depress the key on the keyboard that corresponds to what is inside the brackets (either a character or a special key like CR (carriage return or enter), ESC (escape), SHIFT, CTRL, ALT).

type something you should type (verbatim) whatever is printed boldfaced.

Starting OPUS

The very first start only initializes the design environment. Left click at the icon **IC Design Framework** on the desktop. A new LINUX shell comes up and asks for the design-directory of OPUS. Type **ams37 <CR>**, the name of the recently established subdirectory.

In return OPUS offers the available technologies. **AMS 0.35 um CMOS (c35b3)** has to be used, so type **1<CR>**. Then OPUS reports that several setup files have been created. This happens only at the first start of OPUS. Afterwards send this window to the panel with a left click at the upper right corner.

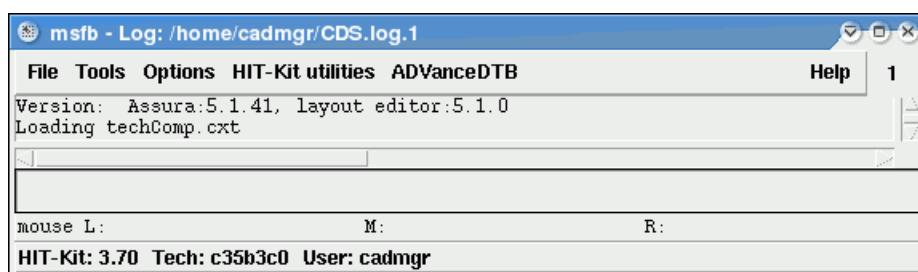


Fig. 1 Command Interpreter window

OPUS goes on. The *Log* window appears with some logging messages, then it changes to the *msfb-Log* window which is called the *Command Interpreter Window (CIW)* because it can accept commands which you type in (Fig. 1.). The library manager window, too, starts automatically, but

¹ Based on P. Gaertner: AMS Hit Kit 3.7 manual, BME DED 2006

here you have to exit OPUS by clicking at **msfb:File->Exit**. So the initialization is done. When you start again, clicking at the icon, then OPUS won't ask any more question and you can go on using the library manager window.

The library manager window can be used for opening existing libraries or cells or creating new ones. The left column of the library manager window is a list of the current (accessible) libraries. Among these **PRIMLIB** contains the transistors you will need for the inverter.

Left click at **PRIMLIB**. The middle column shows the elements of **PRIMLIB**. Left click at **nmos4**. This is the basic n-MOS transistor. In the third, rightmost column you can see several views of **nmos4**. Of these you will need the *symbol* view for the schematic and the *layout* view for building the layout.

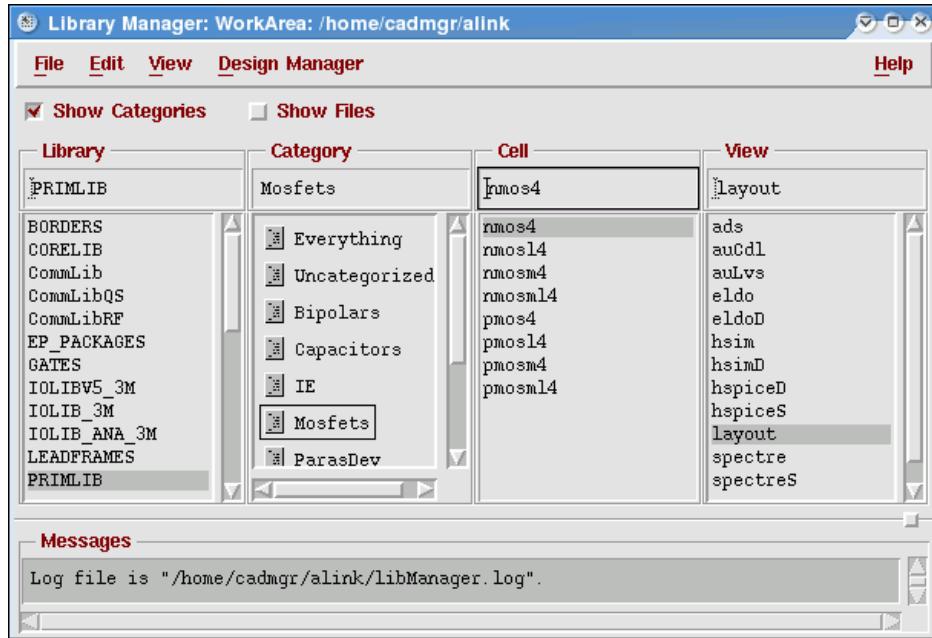


Fig. 2 Library Manager window

Create a new working library

Before building the schematic you have to create a working library. In the *Library Manager* left click at **File->New->Library**. A dialog box appears, asking for the place and name of the new library (Fig. 3.).



Fig. 3 New Library dialog box

Leave the directory at the default, and enter a name for your working library where you are going to design the inverter, such as, for instance, *mylib*. Left click the **OK** button. Cadence now creates a new subdirectory named *mylib* in its home directory (*ams37*). A new window will appear asking information about the technology file. The second option, **Attach to an existing techfile**, will be used, click at it. Then click on the **OK** button. A small dialog box appears asking for the existing techfile (Fig.4.). Left click at the Technology Library button. A list of possible choices pops up. Click at **TECH_C35B3** and then **OK**. Your library is created now and you should be able to locate the new library *mylib* in your Library Manager.

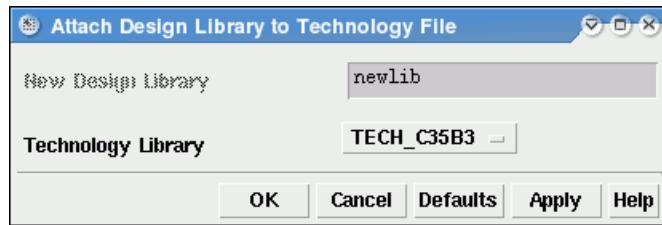


Fig. 4 Choosing the technology for the project

Create the schematic of the inverter

In the *Library Manager* left click on **File->New->Cell view**. The *Create New File* form appears (Fig. 5.).

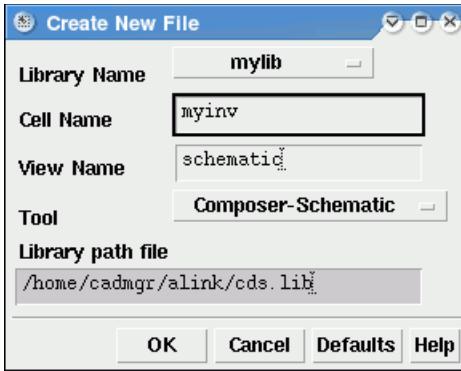


Fig. 5 Specifying the name and view of a new cell

Type a meaningful name in the *Cell Name* block, such as e.g. *myinv*. In the *View Name* block type **schematic** or from the *Tool* menu choose *Composer-Schematic* and the *View Name* block will be automatically filled. Set the library for the would-be cell *mylib*. Left click the **OK** button. The *Virtuoso Schematic Editing window* should show up (Fig. 6:).

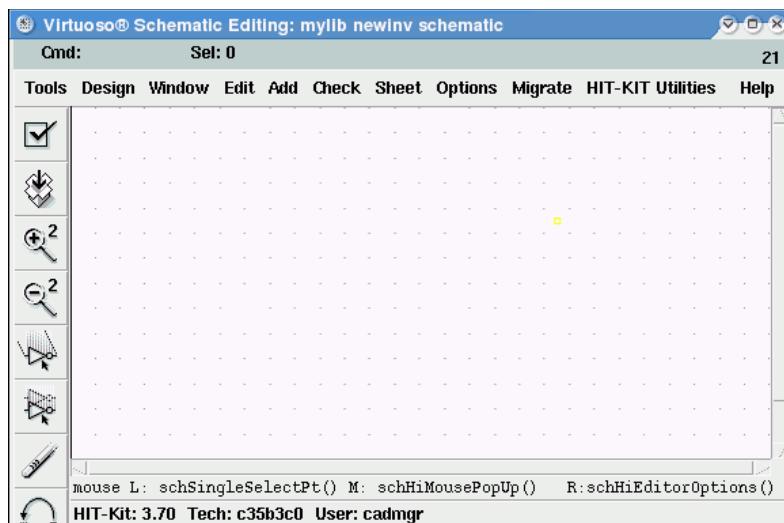


Fig. 6 Schematic Editing window

Left click **Virtuoso Schematic Editing:Add->Instance**. The *Add Instance* dialog box appears (Fig. 7.). Type **PRIMLIB** in the Library field. To choose a four-terminal NMOS transistor type **nmos4** in the *Cell* field and **symbol** in the *View* field. Note that you can use the *Browse* button in order to browse through the libraries and find the cell you want. Generally, typing in known names is faster.

When OPUS learns that you want to place an instance of a transistor then it adds fields to the box for the parameters of the transistor, already containing default values. Just change the *Width* to **2u** (two microns).

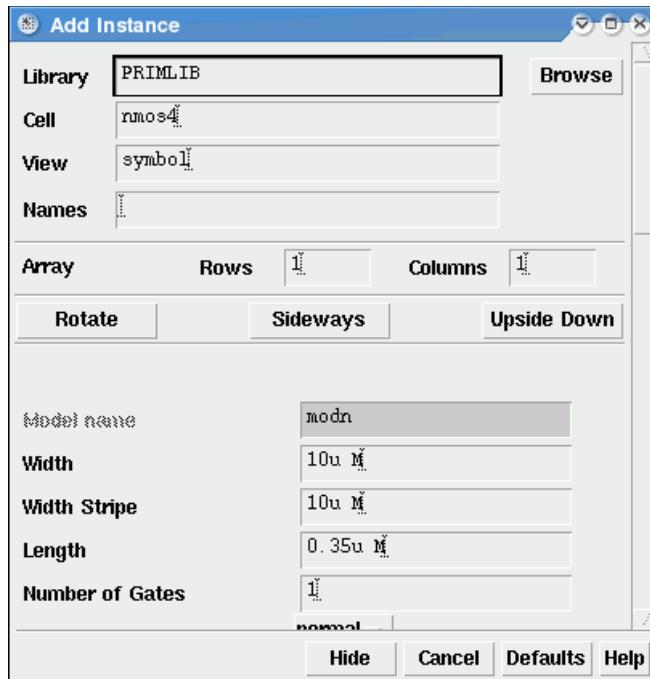


Fig. 7 Add Instance – specifying a transistor

Move the cursor into the editing window. Notice that there is an nmos transistor there instead of the normal cursor. Position it where you want to put the transistor, and place it by a left click. Having placed the first component, change the name of the transistor in the dialog box to **pmos4** and the width to **5u**. Now you have prepared the second half of the inverter and you should place the pmos transistor somewhere over the nmos device so that they can be connected by a straight line. If you type <ESC> then the dialog box disappears and OPUS is waiting for your next command. This will be adding external pins for the inverter.

Left click **Add->Pin**. The *Add Pin* dialog box appears (Fig. 8. next page). Type **in out** in the *Pin Names* field for the pins of the inverter. Set the *Direction* to *input*. (Note that the order of the pins is not important. You may even place one pin at a time and repeat the procedure.)

Move the cursor into the editing window. A pin symbol appears with a small square at the right edge. Place it left of the transistors in the middle with a left click.

In the dialog box *in* disappears from the pin-list, only *out* remains. Change the *Direction* to *output*. In the editing window an output pin symbol appears, having a small square at the left edge. Place it right of the transistors in the middle with a left click.

In a simple case power supply pins (vss, vdd) should be added, too. However, in an IC power supply is provided in a centralized way. In the schematics it is done by the global nodes *gnd!* and *vdd!* while in the layout there are the power rails which do it. Global node names end with an exclamation point (!). Contact to the global nodes *gnd!* and *vdd!* is established by placing instances (small symbols) of the cells *ground* and *vdd*. They are stored in the library *analogLib*. The procedure is the same as for the transistors. Invoke the *Add Instance* window with a left click at **Add->Instance**. Select with the browser the library *analogLib* and then the cells *ground* and *vdd*. Place them underneath and over the transistors, respectively.

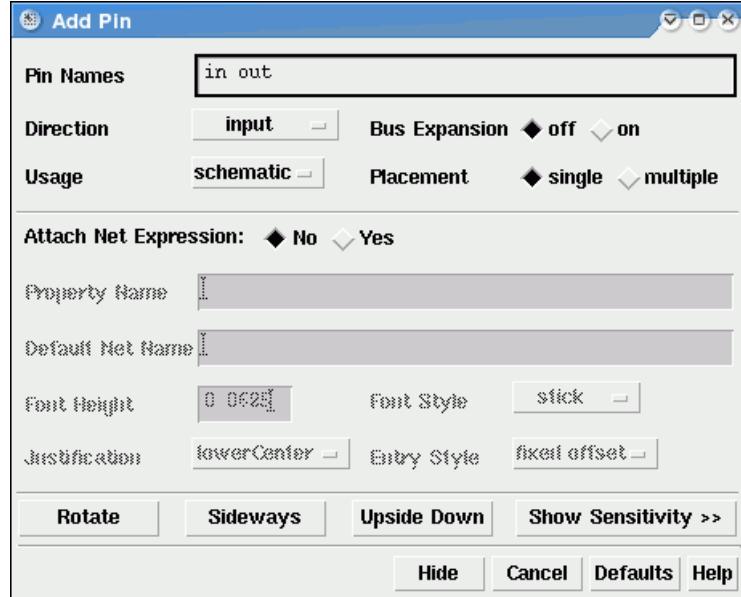


Fig. 8 Add Pin dialog box with pin specification

Now we'll add the wires to make things work. Click **Editing:Add->Wire**. Notice that as you get closer to one pin than to another (including those on devices), a small diamond will show up inside or around that pin. That is where you can click to connect a wire.

To begin with, left click the diamond at the pin of the symbol *vdd*, then left click on the source terminal of the PMOS transistor. The first connection is finished. Now left click on the drain terminal of the PMOS transistor and then on that of the NMOS transistor. Follow that with a wire from the source of the NMOS transistor to the pin of *gnd*. Make one more vertical connection between the gates of both transistors. Now, left click on the diamond in the *in* pin. Move the cursor horizontal to the wire you connected the two gates together with. A diamond will form around the cursor, as long as it is on the wire. Left click. You have just connected the input to the gates of both transistors. Repeat the procedure from the output pin to the wire connecting the drains of both transistors.

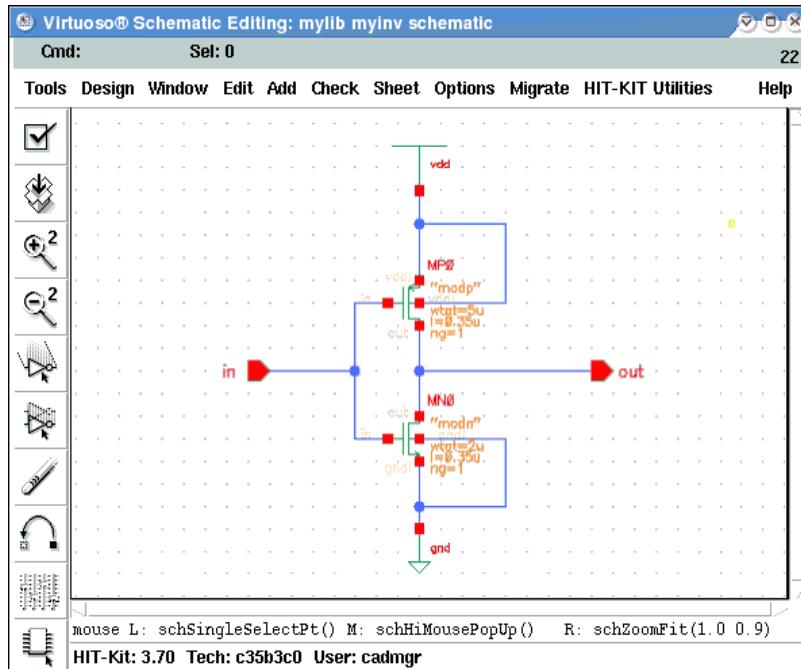


Fig. 9 Complete circuit diagram of the inverter

What remains is connecting the bulk (body) terminals of the transistors. Left click on the bulk terminal of the PMOS transistor. Move the cursor a little right, and left click. The wire will turn here. Now move upwards halfway to *vdd*. Left click again and move to the wire connecting the drain and *vdd*. Connect the bulk of the NMOS transistor to *vss* in a similar manner.

If you happen to put a wire where you don't want it to go, you can delete it by left clicking **Editing:Edit->Delete** and then left click on the object you want to delete (wire, pin, component, etc.).

Once you have done editing, left click the **check mark** (✓) icon on the left side of the screen. This will check your work for connection errors and will save your cell (more exactly: its schematic view!) in the library. You can accomplish the same by left clicking **Editing: Design->Check and save**. Fig. 9 shows what the complete schematic should look like.

This very simple schematic of an inverter will likely be flawless but in more complex designs OPUS may find errors which will be highlighted after the check (blinking). In such a case you may click **Editing:Check->Find Marker**. Then the *Find Marker* window opens (Fig. 10.) and you will find there the list of the highlighted errors and warnings with the reasons stated.

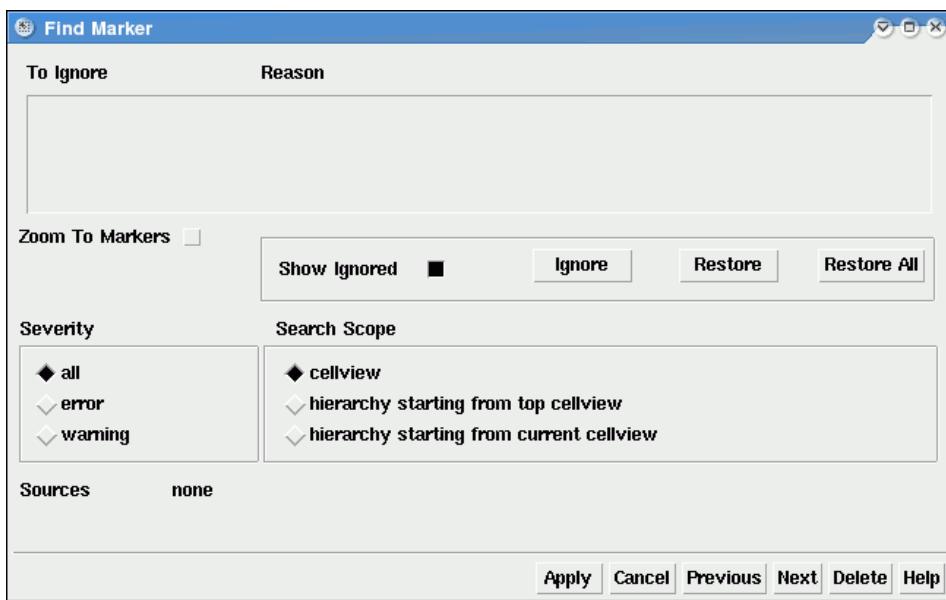


Fig. 10 The window for the list of errors and warnings

Plotting the Schematic of the Inverter

Now that the schematic is complete, you will want to print it out. To do this left click **Editing:Design->Plot->Submit**. The *submit Plot* window should appear (Fig. 11, next page). The default settings usually comprise your schematic and the plotter nearby, so a click on the **OK** button will start plotting. Ensure that the *Header* button is *NOT* selected. This option would produce an extra page with general information on your plot like name and size etc..

If the paper box of the plotter is empty and you happen to want to plot on a sheet of paper which only has one free (empty) side, then make sure that the empty side of the paper looks downwards.

Create a symbol for the inverter

The symbol editor lets you create a "black box" description of a cell using labels, pins, shapes, notes and a selection box. Symbols enable you to introduce hierarchy into your designs. In the *Library Manager* left click on **File->New->Cell view**. The *Create New File* form appears. Ensure that the library name is *mylib*. Fill in the cell name *myinv* and the view name *symbol*. Left click the **OK** button. The *Virtuoso Symbol Editing* window should show up. (Fig. 12. on the next page shows it with the would-be result.).

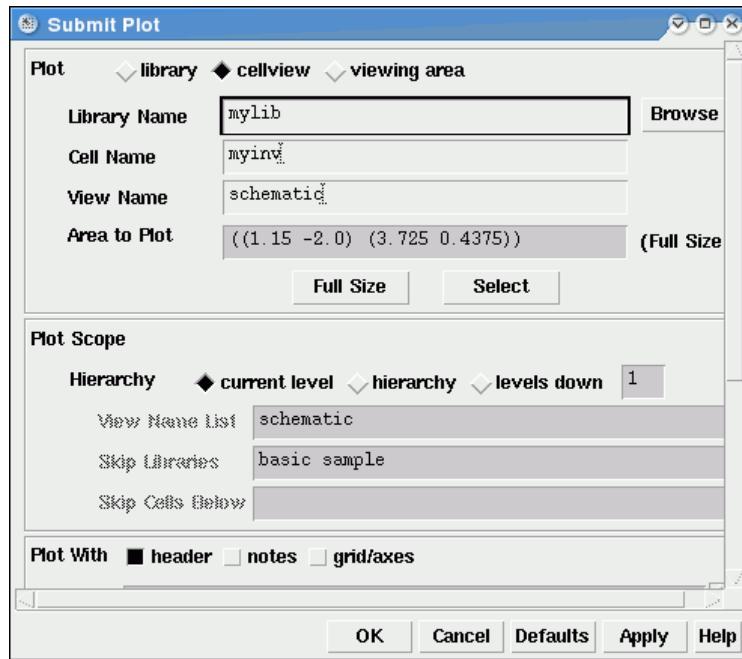


Fig. 11 Sending a circuit diagram to the plotter

Start drawing with a triangle to represent the inverter body. Left click **Editing:Add->Shape->Polygon**. To draw a polygon, left click at a start point and then click at the corners of the shape you want to create. To finish the polygon, click again on the start point. Since we have an inverter, we need an "inverter-like" triangle.

As to the size of the symbol: Note that there are small white dots in the black background of the editing window. If you carefully move the cursor then you will find that its movement is quantized, between two dots it can make 16 small jumps. The triangle should occupy about a "4 by 4 jump" area.

The inverter needs a negation circle at the sideways corner of the triangle, so left click **Editing:Add->Shape->Circle**. Left click at the would-be center of the circle and then at the corner of the triangle. A radius of "one jump" is recommended.

Next you have to create pins for the symbol. It is similar to creating pins in the schematic but the pins look different. They consist of a little red dot and of a piece of line. The dot is the pin itself. The line binds it to the body of the symbol, its length can be adjusted.

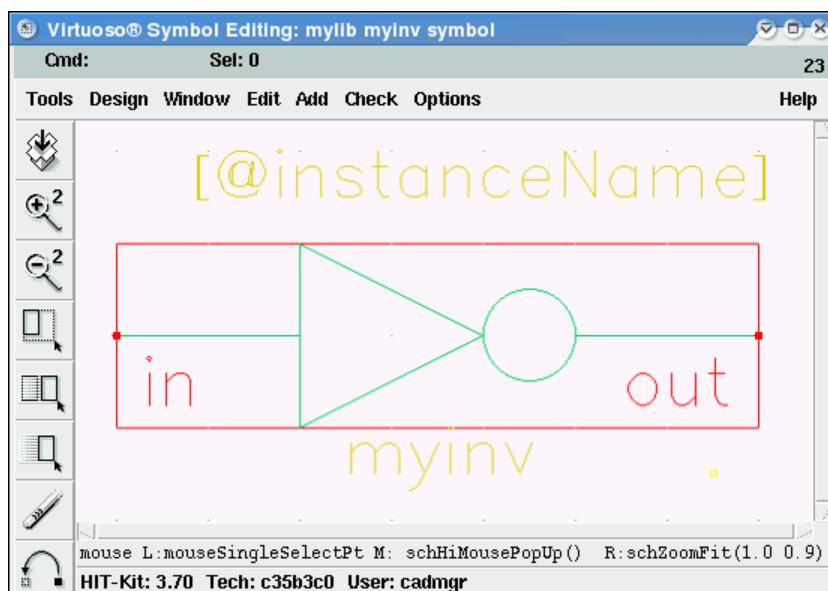


Fig. 12 Symbol Editing window with the symbol

Left click **Editing:Add->Pin**. The *Add Pin* box shows up. Type the pin names, they *must exactly match* those of the schematic: **in out**. Do not forget to set the correct direction for the pins before placing them. Moving the cursor to the editing window the pin appears. With left clicks on **Add Pin:Rotate** you can change the direction of the connecting line. Place the pins so that the red dots are at the far end and the connecting lines join the body of the symbol. At last the position of the pin names have to be adjusted so that the symbol looks nice. Moving the cursor to a name a yellow box appears around it. Now you can left drag the name to its final position.

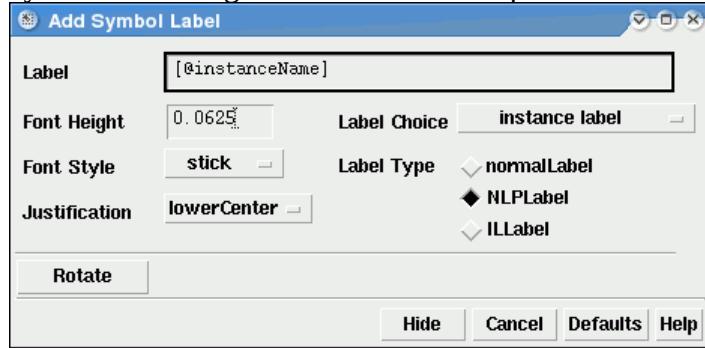


Fig. 13 Adding a label to the symbol

Next we want to add two labels to the symbol. Left click **Editing:Add->Label**. The *Add Symbol Label* dialog box should appear (Fig. 13.). The usual default setting is *[@instanceName]*, Label Choice: *instance label*, Label Type: *NLPLabel*. With this setting you only have to move the cursor to the editing window. The label *[@instanceName]* at once appears and you can place it with a left click. The next label is the name of the cell. Fill into the label field **myinv** and choose *Label Type normalLabel*. Place it again with a left click.

The last thing to add is a selection box. This will tell the software how much of the symbol is actually used. Left click **Editing:Add->Selection Box**. Left click the **Automatic** button. The selection box will be automatically drawn.

The symbol is now finished and you can save it by left clicking **Editing:Design->Save**. If the pin names and attributes do not match those of the schematic then warnings show up and you have to correct the mismatch.

Simulate the schematic

The functionality of an integrated circuit is verified by simulation. For a simulation the following things are needed:

- **netlist**
 - a text file consisting how the components are connected together and what are the component values (e.g. the resistance of a resistor)
- **power supply**
 - the power supply applied to operate the circuit
- **input signal source**
 - the signal source which drives the input(s) of the circuit
- **stimuli**
 - the time function of the signal(s) of the signal source(s) allows to simulate all the functionality of the circuit

The netlist can be extracted from the schematic, it is usually done automatically. Power and input signals are provided by generators. They might be directly added to the schematic but this method is not recommended. Instead, a test bench should be built which takes the cell to be tested as an instance and provides the necessary simulation environment. This has several advantages. The cell remains unchanged and independent of the simulation. It is quite easy to simulate and compare different versions and views of the cell, you only have to specify a cell and a view in the test bench.

Create a test bench

To keep things apart, test benches are usually built in a separate library. For this reason open a new directory *testlib* just the same way you created *mylib* for the schematic of the inverter (page 4). Then open a new cell with schematic view, e.g. *test_inv*. The first element of the testbench is the cell to be tested. Left click **Editing:Add->Instance**. A dialog box comes up and you can either type *Library mylib, Cell myinv*, or you can browse through the libraries and specify it there. Place the symbol of the inverter in the middle of the screen.

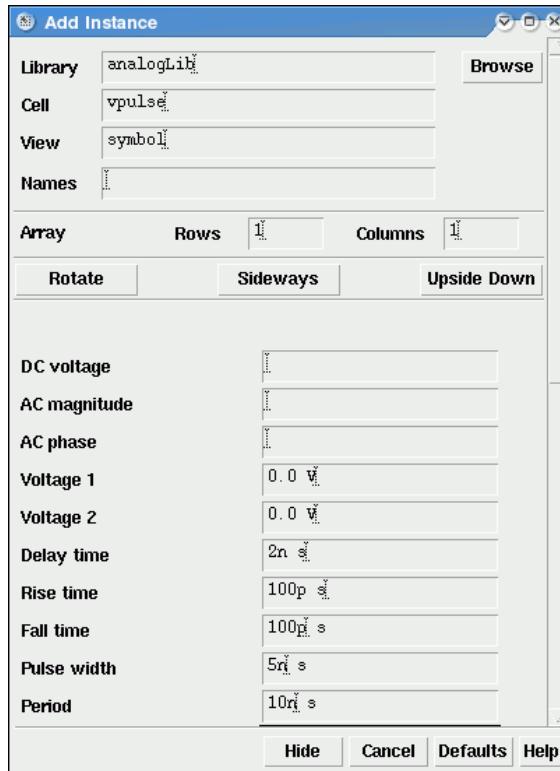


Fig. 14 Specifying a pulse generator

Left click **Add->Instance->Browse**. Select the generator *vdc* from the library *analogLib*. Set the field *DC Voltage* to **5 V**. This will be the power supply, place it far left. Now you might go on browsing for other components but in this case it can be done easier, too. Change the name of the cell to *vpulse* and close it with a <TAB>. The dialog box changes and parameter fields of the pulse generator appear (Fig. 14.). *Voltage 1* is the low level of the pulse, fill in **0**. *Voltage 2* is the high level of the pulse, fill in **5** for 5 Volts. Specify the timing as follows: *Delay time* = **2n**, *Rise time* = **.1n**, *Fall time* = **.1n**, *Pulse width* = **5n**, *Period* = **10n**. This generator will drive the inverter, you may change the Instance Name to *drv*, and place it near the input of the inverter. Change the name of the cell to *cap* and close it with a <TAB>. The dialog box changes and now the value of the capacitive load can be set to **.3p** (0.3 pF). Place the capacitor near the output of the inverter.

Now left click **Add->Wire** or type simply <**w**>. Make the common ground net connecting the lower terminals of the three components. Next connect *vpulse* to the input and the capacitor to the output of the inverter. Place one more piece of wire to the output and add an output pin named *out* (left click **Add->Pin**, etc.). Left click **Editing: Add->Wire Name**. Fill in *Names: uin*. Move the cursor to the wire connecting *vpulse* and the input of the inverter and place the name onto it. In the next step you have to provide for the global *vdd!*. It is done by placing a *vdd* symbol from the library *analogLib* and connecting it to the positive terminal of the *vdc* generator.

It is important for the simulator that the ground net has the name *gnd!* and the internal node number zero. This can be achieved by placing one more component. Left click again **Editing: Add->Instance** and fill in: *analogLib* for the Library and *gnd* for Cell name. Place the ground symbol underneath and connect it to the ground net by a piece of wire. If you happen to get very unusual and unlikely voltage values resulting from the simulation then check if this condition is fulfilled! The test bench is complete, left click **Design->Check and Save** (Fig. 15.).

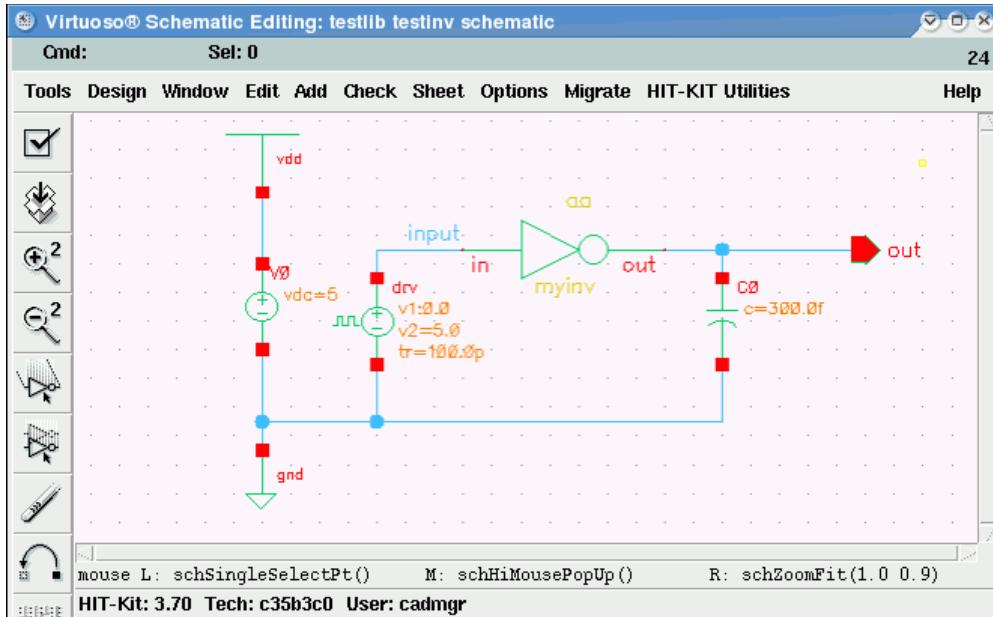


Fig. 15 Testbench – test environment for the inverter

Spice simulation with *eldoD* of Mentor Graphics

We are going to analyze the DC transfer characteristics and the transient behavior of the inverter. Open the schematic *test_inv* in the library *testlib*. Left click **Tools->Analog Environment**. The *Virtuoso Analog Design Environment* window opens (Fig. 16). In the status bar (second from top) the default simulator Spectre is displayed. In order to change the simulator left click **Setup->Simulator/Directory/Host**. The *Choosing Simulator* window opens. The simulator box displays *Spectre*. Left click on the box and select *eldoD*, then click *OK* (Fig. 17.). Check the change in the status bar of the *Design Environment* window.

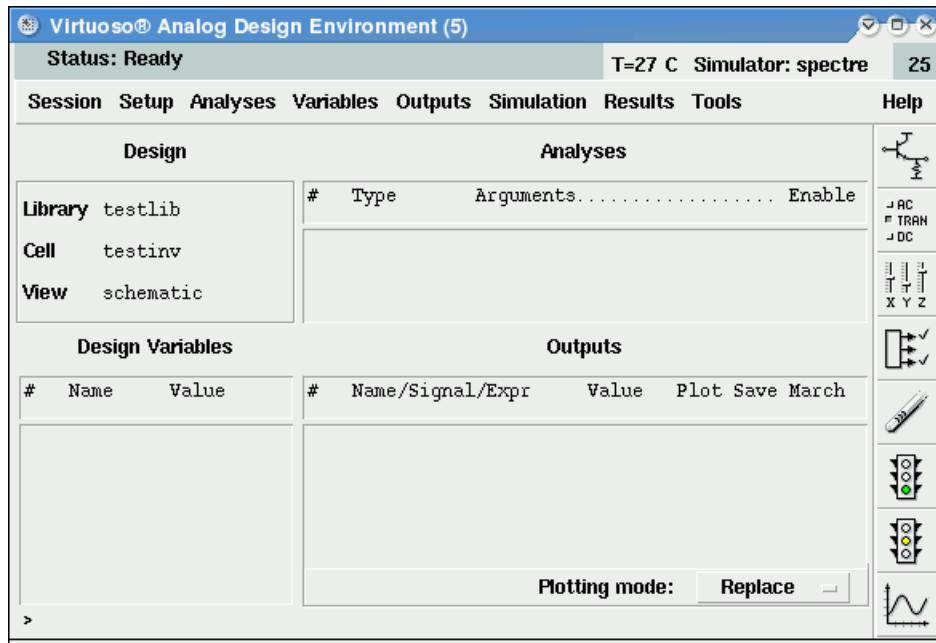


Fig. 16 Analog Environment -- main control panel

Left click **Analyses->Choose**. The *Choosing Analyses* dialog box comes up. Now specify the simulation. Select *Analysis: dc* (Fig. 18, next page), switch off *Print Operating Point*, select *Sweep Variable: Source* (1). Next you specify *Sweep Range*. Fill in *Start 0* and *Stop 5* and *By 0.01*. Left click at the box **Select S1**. Switch over to the schematic and select the the sweeping source by clicking at the pulse generator *drv* at the input of the inverter. Then switch back to the *Choosing Analyses* dialog box. In the *1stSource* field */drv* should appear. See if the *Enabled* checkbox is switched on and click at **Apply**. In the same window you can specify the transient simulation as well. Select **tran** and set *From Time 0*, *To Time 25n*. Switch on the *Enabled* checkbox and click **OK**. In the *Analyses* field of the *Analog Environment* window the specified data appear.

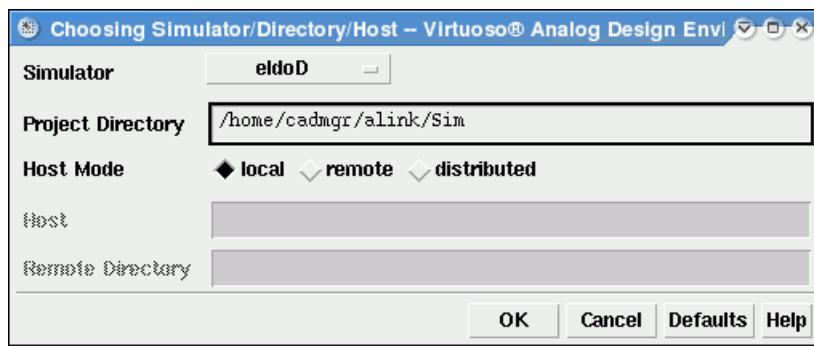


Fig. 17 Choosing the eldoD simulator

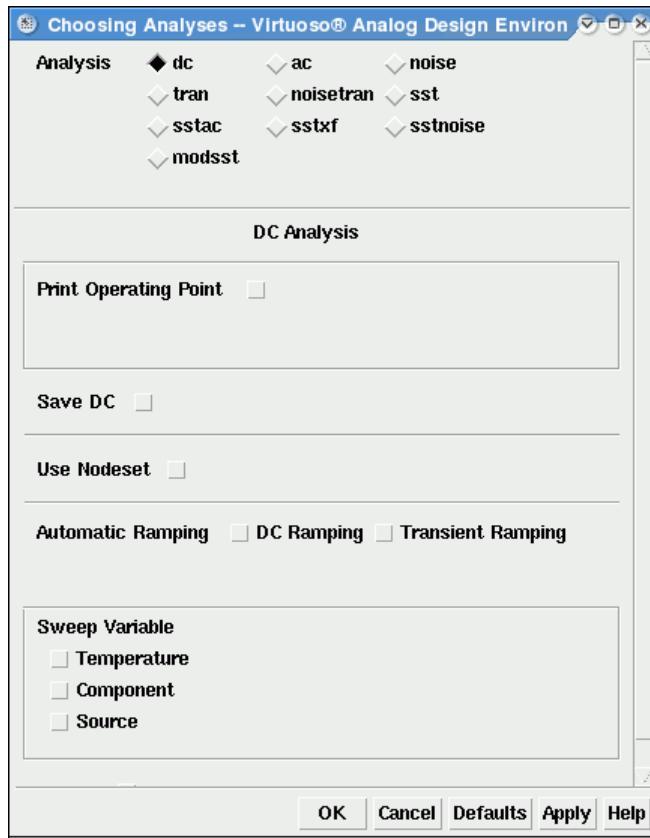


Fig. 18 Selecting simulation parameters

Next we want to select which results should be plotted. Left click on **Analog Design Environment:Outputs->To be Plotted->Select on Schematic**. Click on the wire between your pulse generator and the *in* pin of your inverter bearing the label *input*. Then click on the wire between the *out* pin of the inverter and the *out* pin of the test bench. Both wires should change color indicating that these voltages will be plotted.

Note: if you want to select a current to be plotted then click on the square of a symbol where the current is flowing through. There will be a circle around the square node indicating that a current is selected. Try it with the *vss* and *vdd* terminals of the inverter. A second click on the same square cancels the selection.

Now you are ready to run the simulation. Left click **Simulation->Run** or click on the green traffic light icon on the right side of the *Analog Design Environment* window. If a dialog box appears now requiring your decision whether some simulation results should be saved then your answer *must be Yes!*

The simulation runs and after a while the results will be plotted in two different ways. The first is the *Waveform Window* of Opus, you should ignore it by simply closing. The second one is the window of the display program *EZwave* which we are going to use. Extend this window to the whole screen and arrange the results neatly side by side.

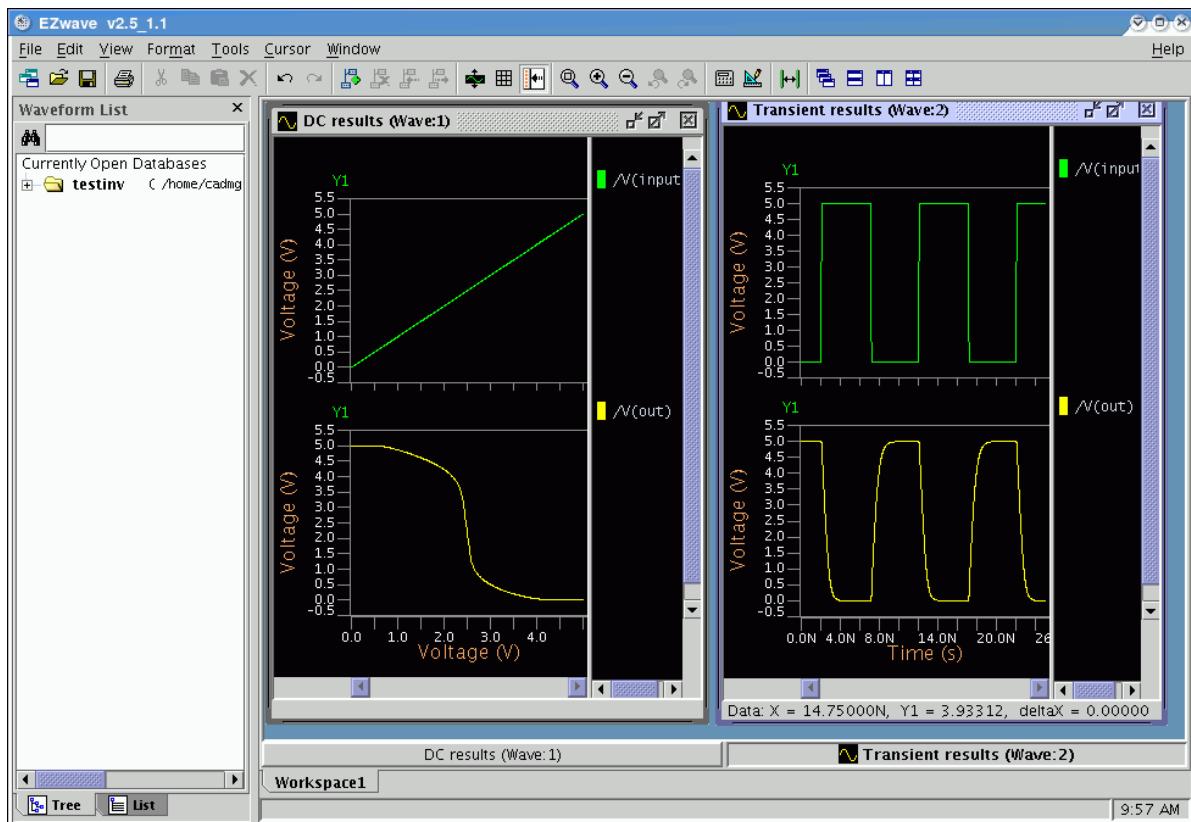


Fig. 19 Simulation results in the EZwave window

The results are plotted in the same coordinate-system. You can separate them by a right click inside the window and activate *Split* in the pop-up menu (Fig. 19). On the right side of the curve you see the colour-code and the name of the curve. To unite curves you can select the curve by left-click at its name and then left-drag the highlighted name to another one.

To finish the simulation exit *eldoD Spice*. Left click on **Analog Design Environment: Session->Quit**. Remember *NOT* to save the current state. If you choose to save then several hundreds of megabytes will be used in order to save your last simulation.

Note: the specified simulations in the *Analyses* field can be enabled and disabled one by one. To do so select the simulation by a left click and then left click **Analog Design Environment:Analyses->Enable** or *Delete* (or even *Delete*).

Exercise using the cursors and the slope function

Read the section **Simulation: EZwave Window, Cursors and Slope** (page 35) and do the following exercise. Start the *DC Sweep* simulation of the inverter again, with the curves *out* and *uin*. When the *EZwave* window opens with the curves make it to form a large square on the screen. Now find three characteristic points of the output curve of the inverter:

1. The *inflection point* where the slope of the curve, and so the *voltage gain* of the stage, is maximal
2. Determine the *noise immunity* of the inverter by finding those two points of the curve where the slope is just -1. The distance of these points from the end of the curve (0 or VDD, respectively) can be regarded as the *noise margin*. (One of them is shown in Fig. 20.)
3. Determine the transient times of the inverter: the propagation delay at 50% signal value and rise and fall time at the output between 10% and 90%. Arrange the transient signals in one coordinate system. So the propagation delay(s) can be determined. Apply one cursor to each curve and bring them to 2.5V. Read the time difference of the cursors at the bottom of the plot. Then delete the input curve and apply both cursors to the output curve. Set them so that they are

at 0.5V and 4.5V, respectively. So you can read the rise and fall time values (Fig. 20.)

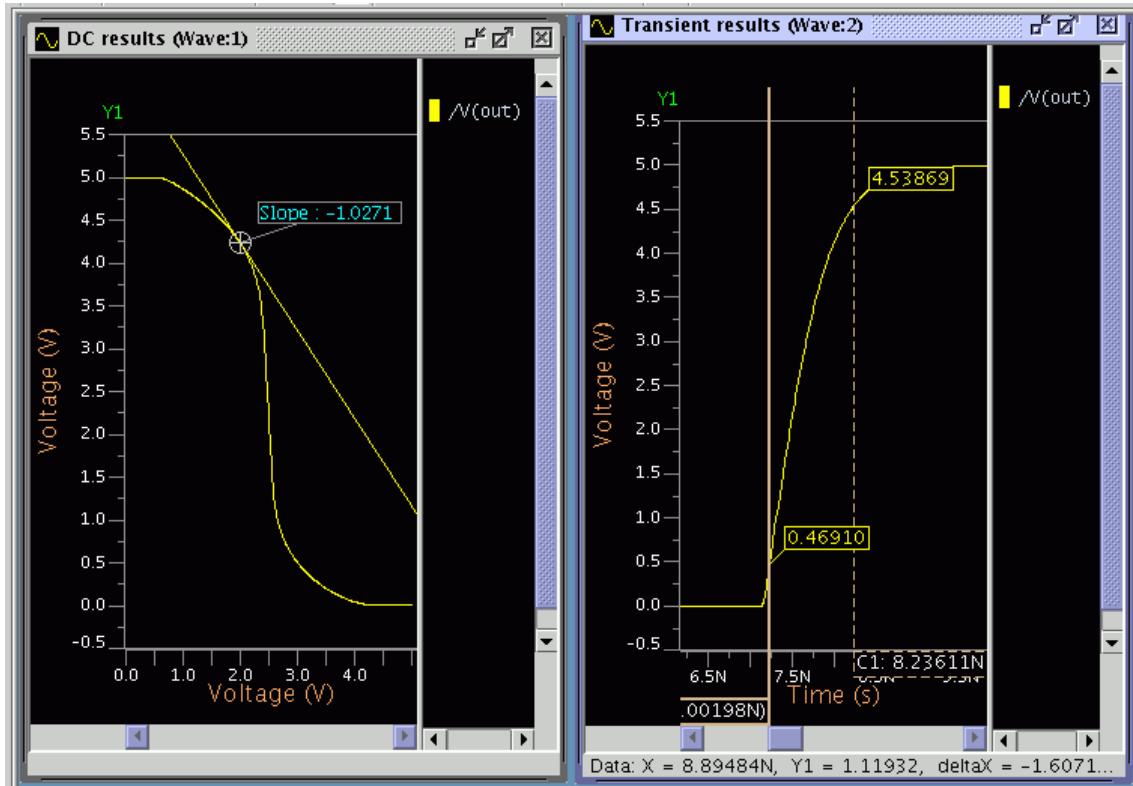


Fig. 20 Finding voltage gain and rise time



*Budapest University of Technology and Economics
Department of Electron Devices*

Lab. 2

Hierarchical design, Testbench, Concurrent statements

Osama Ali

osamaalisalman.khafajy@edu.bme.hu

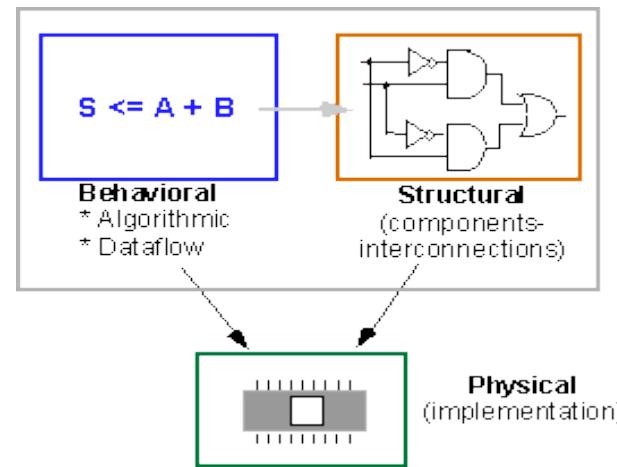
Ahmad Halal

ahmadhalalfesah@edu.bme.hu

13 of May 2022

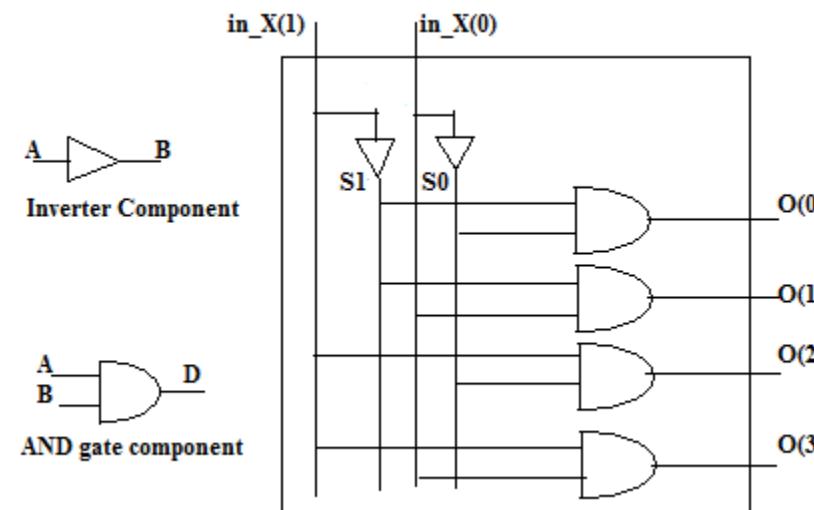


Hierarchical Modeling



- By adopting a more hierarchical design approach it is possible to reuse common elements, and segment a complex design into smaller pieces. Both of these techniques result in a more maintainable design.

- E.g.



Hierarchical Modeling

- To incorporate hierarchy in VHDL we must add **component declarations** and **component instantiations** to the model.
- **Component Represents a precompiled Entity-Architectecture Paire**
- Instantiation is selecting a component and using it as an instance in our design
- we need to declare internal signals to interconnect the components.
 - Format for Architecture body (for internal signals & hierarchy):

```
architecture architecture_name of entity_name is
  signal declarations                                     -- for internal signals in model
  component declarations                                 -- for hierarchical models
begin
  :
  component instantiations
  .
  concurrent statements
  :
end architecture architecture_name;
```



Component Declaration

- Format for component declaration:

```
component component_name is
  :
  port (signal_name(s)): mode signal_type;
  :
  signal_name(s): mode signal_type);
end component component_name;
```

- the component_name is the same as the entity_name from the model being called up.
- component declarations look just like the entity statements for the component being declared but with “component” substituted for “entity”.

Component instantiation

- The component instantiation is the actual call to a specific use of the model.
- A single component declaration can have multiple instantiations.
- each component instantiation must include a unique name (instantiation_label along with the component (component_name) being used.
- There are two methods (and formats) for connecting signals to the port of the component:

Keyword association	Positional association
<pre>instantiation_label: component_name : port map (port_name => signal_name, : port_name => signal_name);</pre>	<pre>instantiation_label: component_name : port map (signal_name, : signal_name);</pre>

Refreshing: Concurrent statement when else & with select

with select

```
entity ex is
  port (a,b,c: in std_logic;
        data: in std_logic_vector (1 downto 0);
        q: out std_logic);
end;
architecture dtf of ex is
begin
  with data select
    q <= a when "00",
      b when "11",
      c when others;
end;
```

when else

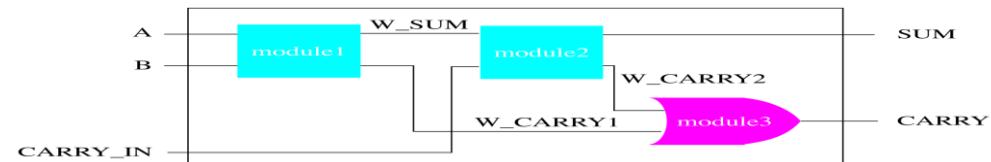
```
entity ex is
  port (a,b,c: in std_logic;
        data: in std_logic_vector (1 downto 0);
        q: out std_logic);
end;
architecture rtl of ex is
begin
  q<=a when data="00" else
    b when data="11" else
    c;
end;
```

Structural style example:

- download those files from Model in a new folder on the desktop.
- Add all files to a new project in Modelsim
- And then compile all and simulate File1. By force the values of A, B and CIN the output should be like that



A	B	CIN	SUM	CARRY
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



File1

```

1 LIBRARY IEEE;
2 USE IEEE.STD_LOGIC_1164.ALL;
3
4 ENTITY FULLADDER IS PORT(A : in STD_LOGIC;
5                             B : in STD_LOGIC;
6                             CARRY_IN : in STD_LOGIC;
7                             SUM : out STD_LOGIC;
8                             CARRY : out STD_LOGIC);
9 END FULLADDER;
10
11 ARCHITECTURE STRUCTURAL OF FULLADDER IS
12   COMPONENT ORGATE PORT(X : in STD_LOGIC;
13                           Y : in STD_LOGIC;
14                           Z : out STD_LOGIC);
15   END COMPONENT;
16
17   COMPONENT HALFADDER PORT(U : in STD_LOGIC;
18                             V : in STD_LOGIC;
19                             SUM : out STD_LOGIC;
20                             CARRY : out STD_LOGIC);
21   END COMPONENT;
22
23   SIGNAL W_SUM : STD_LOGIC;
24   SIGNAL W_CARRY1 : STD_LOGIC;
25   SIGNAL W_CARRY2 : STD_LOGIC;
26
27 BEGIN
28   MODULE1: HALFADDER PORT MAP(A, B, W_SUM, W_CARRY1);
29   MODULE2: HALFADDER PORT MAP(W_SUM, CARRY_IN, SUM, W_CARRY2);
30   MODULE3: ORGATE PORT MAP(W_CARRY1, W_CARRY2, CARRY);
31
32 END STRUCTURAL;
33 
```

File2

```

1 LIBRARY IEEE;
2 USE IEEE.STD_LOGIC_1164.ALL;
3
4 ENTITY HALFADDER IS
5   PORT(U,V: IN STD_LOGIC;
6         SUM, CARRY : OUT STD_LOGIC);
7 END HALFADDER;
8
9 ARCHITECTURE RTL_HALFADDER OF HALFADDER IS
10 BEGIN
11   SUM <= U XOR V;
12   CARRY <= U AND V;
13 END; 
```

File3

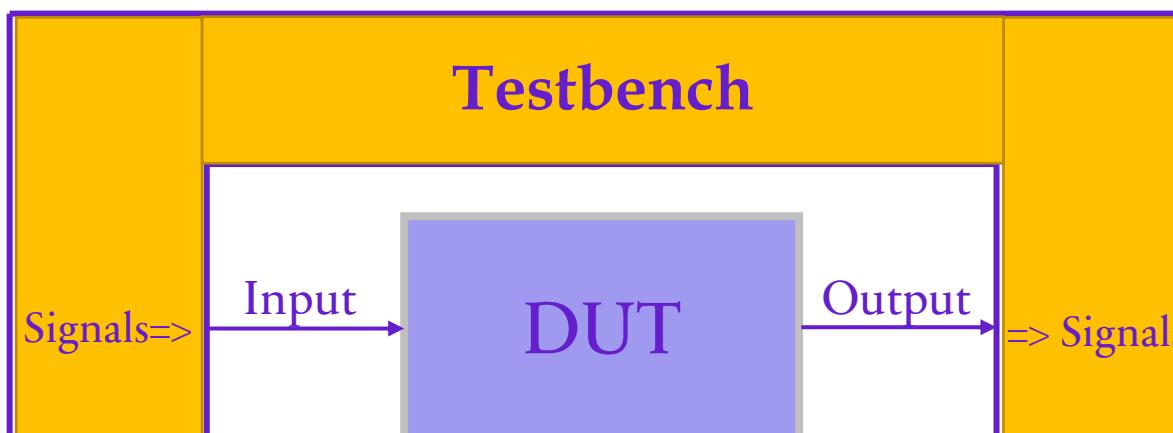
Ln#	Code
1	LIBRARY IEEE;
2	USE IEEE.STD_LOGIC_1164.ALL;
3	ENTITY ORGATE IS
4	PORT(X,Y : IN STD_LOGIC;
5	Z : OUT STD_LOGIC);
6	END ORGATE;
7	
8	ARCHITECTURE RTL_ORGATE OF ORGATE IS
9	BEGIN
10	Z <= X OR Y;
11	END RTL_ORGATE;
12	

Testbench

- Testbench is an important part of VHDL design to check the functionality of Design through simulation waveform.
- Testbench provides a stimulus for **design under test** DUT or **Unit Under Test** UUT to check the output result.

- **A Test Bench consists of:**

- Entity
 - has no ports (empty entity header)
- Architecture
 - declares, instantiates, and wires together the driver model and the model under test
 - driver model provides the stimulus and verifies model responses



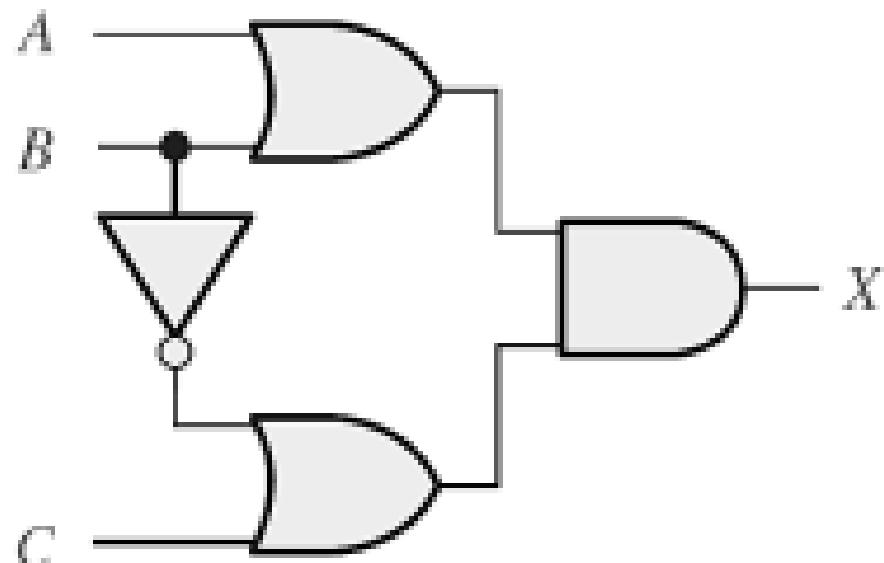
Full-Adder Testbench

```
Ln# |  
1  LIBRARY IEEE;  
2  USE IEEE.STD_LOGIC_1164.ALL;  
3  
4  ENTITY TEST_FULLADDER IS  
5  END TEST_FULLADDER;  
6  
7  ARCHITECTURE TEST_BEHAVIORAL OF TEST_FULLADDER IS  
8  
9      SIGNAL A : STD_LOGIC;  
10     SIGNAL B : STD_LOGIC;  
11     SIGNAL CIN : STD_LOGIC;  
12     SIGNAL SUM : STD_LOGIC;  
13     SIGNAL CARRY : STD_LOGIC;  
14  
15     COMPONENT FULLADDER  
16         PORT (A : in STD_LOGIC;  
17             B : in STD_LOGIC;  
18             CARRY_IN : in STD_LOGIC;  
19             SUM : out STD_LOGIC;  
20             CARRY : out STD_LOGIC);  
21     END COMPONENT;  
22  
23 BEGIN  
24     DUT : FULLADDER  
25         PORT MAP (A, B, CIN, SUM, CARRY);  
26  
27     STIMULUS: PROCESS  
28     BEGIN  
29         A <= '0'; B <= '0'; CIN <= '0';  
30         WAIT FOR 100 NS;  
31         A <= '0'; B <= '0'; CIN <= '1';  
32         WAIT FOR 100 NS;  
33         '  
34         '  
35         '  
36         WAIT;  
37     END PROCESS;  
38 END TEST_BEHAVIORAL;
```



Group 1

- Write a VHDL Code to Implement the below circuit function using hierarchical modelling and data flow style for the components
 - Use Signal for the internal connection between gates
 - Verify your design by simulation, use Testbench to force different input patterns and check the output

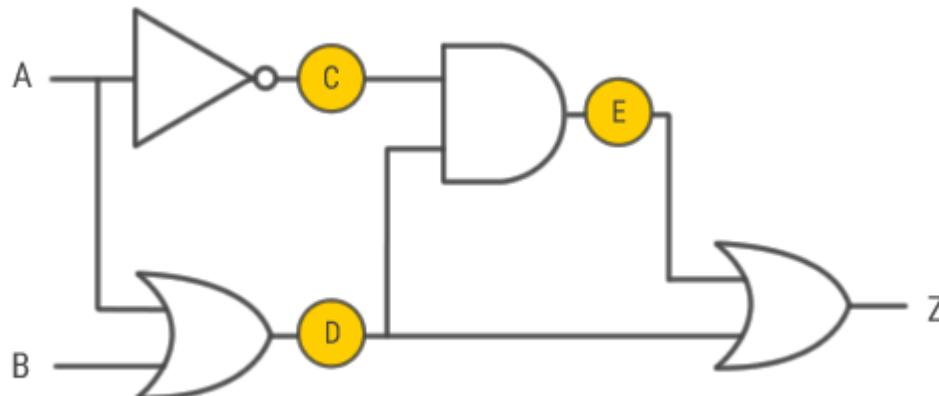


H.W:

Instead of a data flow style use a behavioral style utilizing the concurrent statements.

Group 2

- Write a VHDL Code to Implement the below circuit function using hierarchical modelling and data flow style for the components
 - Use Signal for the internal connection between gates
 - Verify your design by simulation, use Testbench to force different input patterns and check the output



H.W:

Instead of a data flow style use a behavioral style utilizing the concurrent statements.



*Budapest University of Technology and Economics
Department of Electron Devices*

Lab. 1

Digital system design

Introduction, language structure, examples

Osama Ali

osamaalisalman.khafajy@edu.bme.hu

Ahmad Halal

ahmadhalalfesah@edu.bme.hu

6 of May 2022



Outline

- VHDL language structure
- Architecture
- Diferent styles
- Signal & Variables



General Considerations

- Case insensitive
- Comments: ' -- ' until end of line
- Statements are terminated by ' ; '
- List delimiter: ' , '
- Signal assignment: ' <= '
- User defined names:
 - letters, numbers, underscores
 - start with a letter!
 - underscores

It is an example don't write like that in VHDL

```
SIGNAL mySignal : bit;      -- an example signal
Mysignal <= '0'           -- start with '0'
                    '1' AFTER 10 ns; -- and toggle after
                    '0' AFTER 10 ns; -- every 10 ns
                    '1' AFTER 10 ns;
```

```
mySignal_23      -- normal identifier
rdy , RDY , Rdy -- identical identifiers
vector & vector -- special character
last Of Zout   -- white spaces
idle_state      -- consecutive underscores
24th_signal1    -- begins with a numeral
open, register  -- VHDL keywords
\mySignal_23\    -- extended identifier
\rdy\,\RDY\,\Rdy\-- different identifiers
\vector_&_vector\-- legal
\last of Zout\  --legal
\idle_state\    --legal
\24th_signal\   --legal
\open\,\register\ --legal
```

Data Types

- There is a series of pre-defined data types in VHDL through the standard and the IEEE libraries.
- Not all data types are synthesizable.

Package / library	Defined data types
Package standard of library std	BIT, BOOLEAN, INTEGER, and REAL
Package std_logic_1164 of library ieee	STD_LOGIC and STD_ULOGIC
Package std_logic_arith of library ieee	SIGNED and UNSIGNED

STD_LOGIC & STD_ULOGIC	
'1'	Logic 1 or High 1
'0'	Logic 0 or High 0
'Z'	High impedance
'W'	Weak signal, can't tell if 0 or 1
'L'	Weak 0, pulldown
'H'	Weak 1, pullup
'-'	Don't care
'U'	Uninitialized
'X'	Unknown, multiple drivers

- STD_LOGIC (and STD_LOGIC_VECTOR): **8-valued** logic system introduced in the IEEE 1164 standard.
- BOOLEAN: True, False.
- INTEGER: 32-bit integers (from -2,147,483,647 to +2,147,483,647).

```

SIGNAL a: BIT;
SIGNAL b: BIT_VECTOR(7 DOWNTO 0);
SIGNAL c: STD_LOGIC;
SIGNAL d: STD_LOGIC_VECTOR(7 DOWNTO 0);
SIGNAL e: INTEGER RANGE 0 TO 255;
...
a    <= b(5);  -- legal (same scalar type: BIT)
b(0) <= a;    -- legal (same scalar type: BIT)
c    <= d(5);  -- legal (same scalar type: STD_LOGIC)
d(0) <= c;    -- legal (same scalar type: STD_LOGIC)
a    <= c;    -- illegal (type mismatch: BIT x STD_LOGIC)
b    <= d;    -- illegal (type mismatch: BIT_VECTOR x
              -- STD_LOGIC_VECTOR)
e    <= b;    -- illegal (type mismatch: INTEGER x BIT_VECTOR)
e    <= d;    -- illegal (type mismatch: INTEGER x
              -- STD_LOGIC_VECTOR)

```

```

x0  <= '0';      -- bit, std_logic, or std_ulogic value '0'
x1  <= "0001111"; -- bit_vector, std_logic_vector,
                    -- std_ulogic_vector, signed, or unsigned
x2  <= "0001_1111"; -- underscore allowed to ease visualization
x3  <= "101111"   -- binary representation of decimal 47

```

```

SIGNAL X: BIT;
-- x is declared as a one-digit singal of type BIT.

SIGNAL Y : std_logic_vector(7 downto 0);
--7th bit is MSB and 0th bit is LSB here.

SIGNAL W : std_logic_vector(0 to 7);
--0th bit is MSB and 7th bit is LSB here.

```

VHDL Operators

- Operators can be used to implement any combinational circuit.

Operator type	Operators	Data types
Logical	NOT, AND NAND, OR, NOR, XOR, XNOR	BIT, BIT_VECTOR, STD_LOGIC, STD_LOGIC_VECTOR, STD_ULOGIC, STD_ULOGIC_VECTOR
Arithmetic	+, -, *, /, **, MOD, REM, ABS	INTEGER, SIGNED, UNSIGNED
Comparison	=, /=, <, >, <=, >=	All above
Shift	SLL, SRL, SLA, SRA, ROL, ROR	BIT_VECTOR
Concatenation	&, (, ,)	Same as for logical operators, plus SIGNED and UNSIGNED

A VHDL design consist of three fundamental design units:

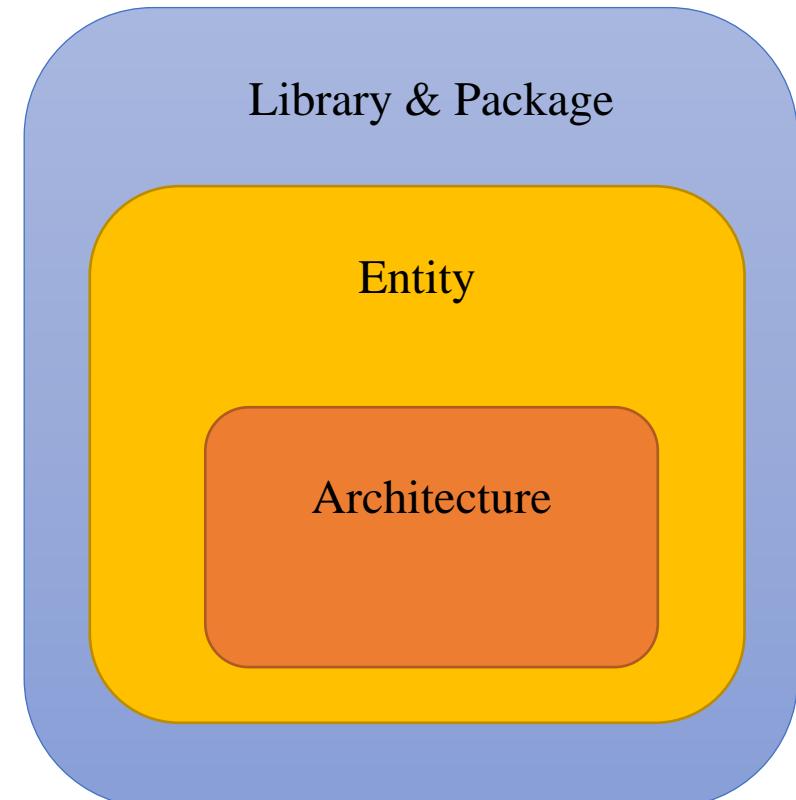
1. Library and Package Declaration

- Library and packages are collection of commonly used items, such as data **types**, **subprograms**, and **components**.

```
library IEEE;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
```

- One can create package that includes several data types, constants, and components. After the IEEE library we can declare the new package.

```
library work;
use work.DataTypes_pkg.all;
```



2. Entity Declaration

- **port name**: used to identify pin(s) and providing the ability to connect it to the design unit or other designs units.
- **mode**: give the direction of the port. It can be **in**, **out**, or **inout**, as will be discussed later
- **data type**: define the data type of the port which can be **bit**, **integer**, **std_logic**, and many other types.

```
Entity Switches_LEDs is
port ( sw_0 : in std_logic;
       sw_1 : in std_logic;
       LED_0 : out std_logic;
       LED_1 : out std_logic
     );
end Switches_LEDs;
```

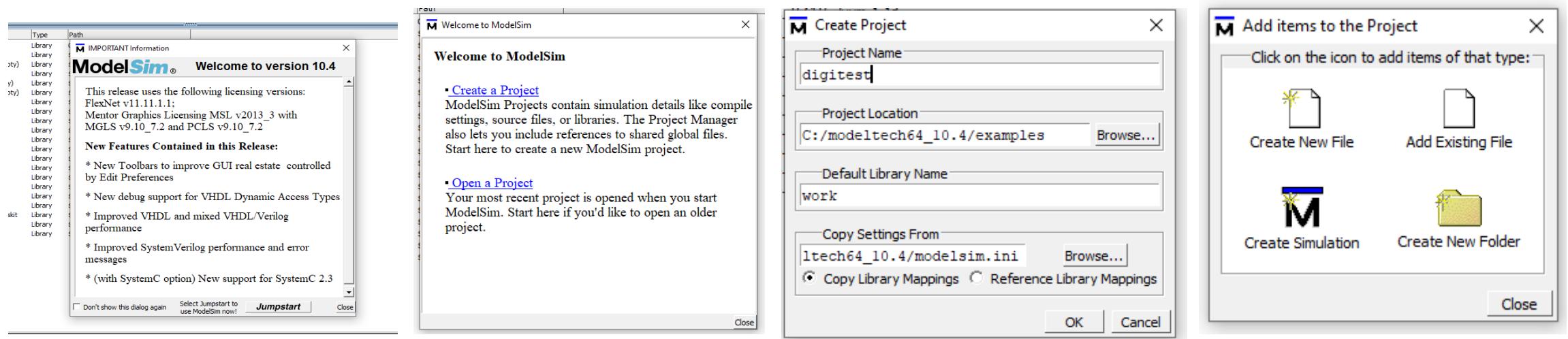
3. Architecture

- Implementation of the design Always connected with a specific entity
 - One entity can have several architectures
 - Entity ports are available as signals within the architecture
- Contains concurrent statements

```
1 LIBRARY IEEE;
2 USE IEEE.STD_LOGIC_1164.ALL;
3
4 ENTITY switch_LED IS
5   PORT(sw_0 : IN STD_LOGIC;
6        sw_1 : IN STD_LOGIC;
7        led_0: OUT STD_LOGIC;
8        led_1: OUT STD_LOGIC);
9 END switch_LED;
10
11 ARCHITECTURE Behav OF switch_LED IS
12   -- in here you can dif. signals and variables
13 BEGIN
14   led_0 <= sw_0;
15   led_1 <= sw_1;
16 END Behav;
```

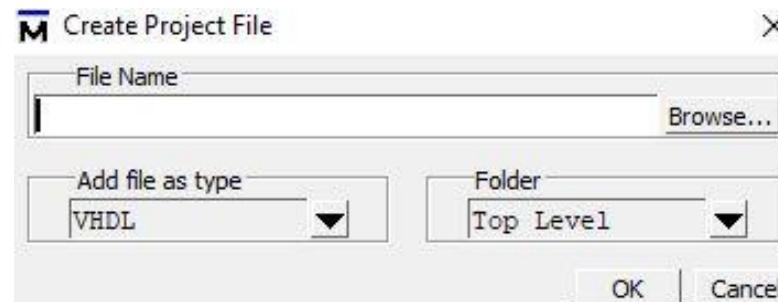
Creating a VHDL Project

- Start the ModelSim
- If the welcome screen popped-up, just press jumpstart.
- Then press Create a Project if you want to create a new one or select Open a Project to continue working on an already existing project. For our first practice we are going to create a new project.
- As shown in the below figure, we have to select the project name and also the project location where it will be saved. We can leave work as the as a library to save out design after compilation.



Adding the source code

- We are free to select any name for the project. However, it is preferred to be a descriptive name.
- It could be more convenient to make a dedicated folder (in advance) for saving the new project all files in a specific location. Let's make a new folder on the desktop, then we choose it as the project location. Then press OK.
- From the next window we add the source VHDL code. We have two options, if we have an already existing VHDL code (.vhd file), we can choose Add Existing File and we browse to it; the other option is to initiate a new source code by selecting Create New File . In the latest case we have to enter a name for the source code.



- If there are no more files, we can close the current window and start editing the source code.

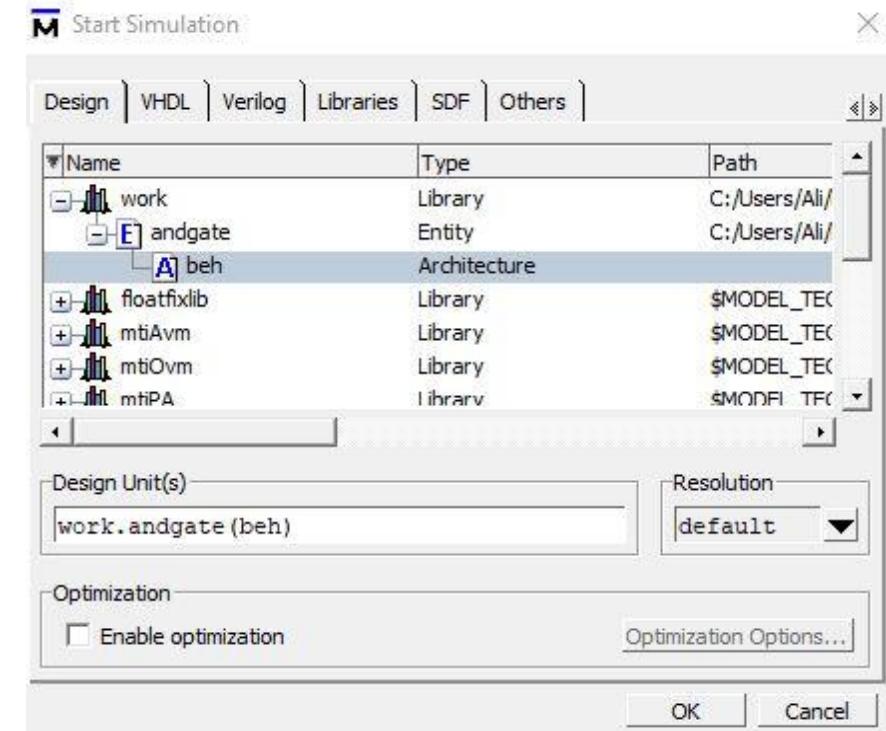
Editing the source code

- For editing the source code, we can use the build in editor of the ModelSim (double click will open the source file) or use any other editor like Sublime.
- After finish editing the code, we have to compile it to make sure that the syntax is correct and also to convert the source code to the objective code and move to the next step.
- For compiling the code there are different ways:
 - right click on the source file and chose compile,
 - from the Compile drop down menu, or
 - press the icon on the tool bar on the top of the window.
- After the first try for compilation, it is probable to get an error message related to some typos. To correct these error, double click on the red message appears in the Transcript sub-window and read the error description, then edit your code.
- After correcting all errors, you should see a message in green that states (Compile of source file was successful).



Start simulation

- For verifying the functional behavior of the design, we have to simulate the compiled design.
- To start the simulation, from the drop-down menu of Simulate we choose **Start Simulation**.
- A new window will appear to select the objective code that we want to simulate from the corresponding library.
- Recall that we have chosen work as the default library. Press on the + beside work and select the name of the entity and under it the name of the architecture that you want to simulate.
- Unselect the option of Enable Optimization and press OK.

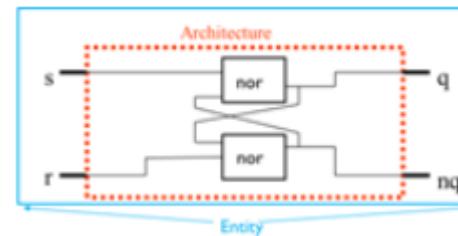
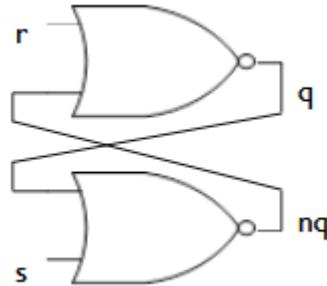


Verifying the design

- To verify the design by applying different input pattern and observe the output we have to add the related signals to the wave window.
- From the Objects sub-window, select the signals that you want to monitor by keep pressing the Ctrl from the keyboard and the lift mouse button, then right click on any of them and choose Add » To Wave » Selected Signals. The wave window will appear with the selected signals.
- To verify the design, we have to provide the input signals with different pattern and check the output. From the wave window, right click on any of the input signals and select Force. In the value field enter the data according to the required test pattern and press OK.
- After doing this for input signals, now its time to start running the simulation by pressing the Run icon from the related tool bar.

Architecture Different styles

- Data flow
 - Describes how data flows from input to output
- Structural
 - How components are put together
- Behavioral
 - Describes the behavior of the circuit most likely within a process



```
process (r,s)
begin
  if (r nor nq) then
    q <= '1';
  else
    q <= '0';
  endif
  ...
end process
```

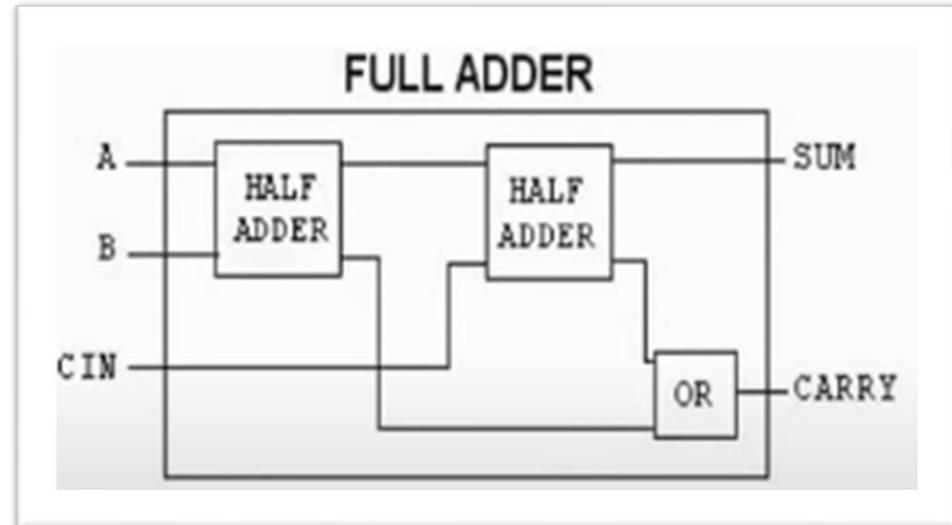
Data flow

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity half_adder_df is
    Port ( a : in STD_LOGIC;
           b : in STD_LOGIC;
           s : out STD_LOGIC;
           c : out STD_LOGIC);
end half_adder_df;

architecture Behavioral of half_adder_df is
begin
    s <= a xor b;
    c <= a and b;
end Behavioral;
```

Structural

- Components from libraries are connected together
- Design are hierarchical
- Each component can be individually simulated



Behavioral

- Also known as High-level Descriptions
- Consists of a set of assignment statements to represent the behavior
- No need to focus on the gate level implementation of a design
- But the execution most likely will be sequential

TRUTH TABLE

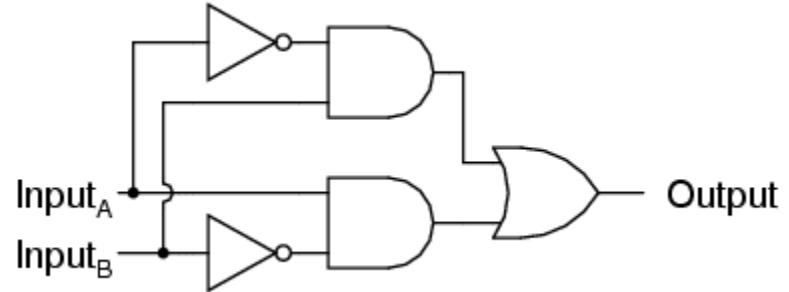
INPUTS		OUTPUT
X	Y	Z
0	0	0
0	1	0
1	0	0
1	1	1

```
Architecture behave of and_gate is
begin
process (a,b)
begin
if a = '1' and b = '1' then
c <= '1';
else
c <= '0';
end if;
end process;
end behave;
```

Signals and Variable

- Represent wires within the circuit.

```
9  ┌─[ ARCHITECTURE AofC1  of C1 is
10     └──[ SIGNAL s1: STD_LOGIC;
11         └──[ SIGNAL s2: STD_LOGIC;
12     ┌─[ BEGIN
13       s1 <= not(InA) and InB;
14       s2 <= InA and not(InB);
15       Ot <= s1 or s2;
16   End AofC1;
```



A	B	Output
0	0	0
0	1	1
1	0	1
1	1	0

SIGNAL IN VHDL

A primary object
describing a hardware
system and are equivalent
to "wires"

An object with a past
history of values

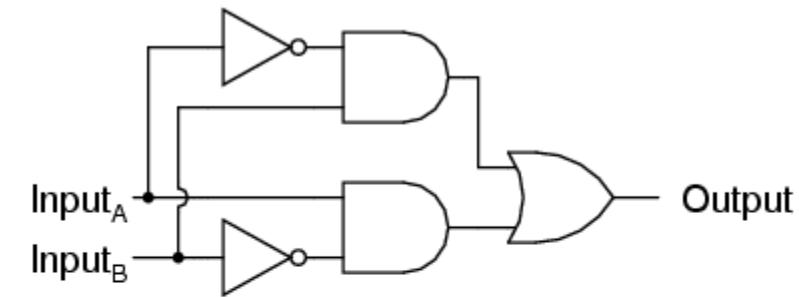
VARIABLE IN VHDL

A variable is an object
which store information
local to processes and
subprograms (procedures
and functions) in which
they are defined

An object with a single
current value

Example: Write this also Test and Verify

```
9  ARCHITECTURE AofC1 of C1 is
10     SIGNAL s1: STD_LOGIC;
11     SIGNAL s2: STD_LOGIC;
12 BEGIN
13     s1 <= not(InA) and InB;
14     s2 <= InA and not(InB);
15     Ot <= s1 or s2;
16 End AofC1;
```



A	B	Output
0	0	0
0	1	1
1	0	1
1	1	0

VHDL online Editor, Compiler and Simulator



If this page reloads when you click "Run" please read [this](#).



Log in with one of the following providers:



Logging in with a social accounts gives you access to all non-commercial simulators and some commercial simulators. If you want to use all the commercial simulators, please register for an account below.

No Google or Facebook account? [Privacy Policy](#)

or

Want full access to EDA Playground?

Username
Password

[Register for a full account](#) [Forgotten password](#)

To run commercial simulators, you need to register and log in simulators.

If you wish to use EDA Playground as a playground, please do

The screenshot shows the EDA playground interface. On the left, there's a sidebar with login fields, registration links, and sections for Languages & Libraries (set to VHDL), Tools & Simulators (set to Aldec Riviera Pro 2019.10), and Examples. The main area has two code editors: 'testbench.vhd' and 'design.vhd'. The 'testbench.vhd' code defines a Fulladder_tb testbench with a PORT section and a component declaration for Fulladder. The 'design.vhd' code defines a Fulladder entity with three inputs (a, b, c) and three outputs (sum, cy). Below the code editors is a log window showing simulation messages. The toolbar at the top includes 'Run', 'Save', and other simulation options. A red arrow points from the 'testbench.vhd' editor towards the 'design.vhd' editor, and a green circle highlights the 'Run' button in the toolbar.

```
testbench.vhd:
1 LIBRARY ieee;
2 USE ieee.std_logic_1164.ALL;
3 ENTITY Fulladder_tb IS
4 END Fulladder_tb;
5 ARCHITECTURE behavior OF Fulladder_tb IS
6 -- Component Declaration for the Unit Under Test (UUT)
7 COMPONENT Fulladder
8 PORT(
9     a : IN std_logic;
10    b : IN std_logic;
11    c : IN std_logic;
12    sum : OUT std_logic;
13    cy : OUT std_logic
14 );
15 END COMPONENT;
16 --Inputs
17 signal a : std_logic := '0';
18 signal b : std_logic := '0';

design.vhd:
1 library IEEE;
2 use IEEE.STD_LOGIC_1164.ALL;
3 entity Fulladder is
4     Port ( a : in STD_LOGIC;
5             b : in STD_LOGIC;
6             c : in STD_LOGIC;
7             sum : out STD_LOGIC;
8             cy : out STD_LOGIC);
9 end Fulladder;
10 architecture Behavioral of Fulladder is
11 begin
12     sum<= a xor b xor c;
13     cy<= (a and b) or (b and c) or (c and a);
14 end Behavioral;
15 
```

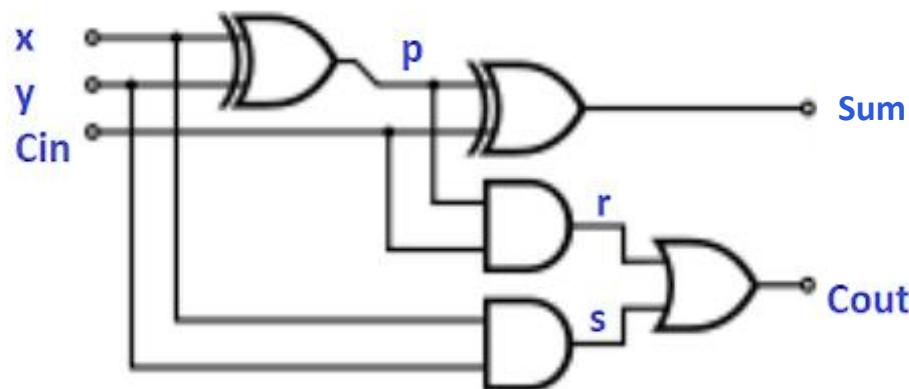
KERNEL: warning: You are using the Riviera-PRO EDU Edition. The performance of simulation is reduced.
KERNEL: Warning: Contact Aldec for available upgrade options - sales@aldec.com.
KERNEL: Kernel process initialization done.
Allocation: Simulator allocated 7447 kB (elbread=1450 elab2=5853 kernel=142 sdf=0)
ASDB file was created in location /home/runner/.dataset.asdb
KERNEL: PLI/VHPI kernel's engine initialization done.
PLI: Loading library '/usr/share/Riviera-PRO-2019.10-x64/bin/libsysytf.so'
KERNEL: Simulation has finished. There are no more test vectors to simulate.
VSIM: Simulation has finished.
Finding VCD file...
.dump.vcd
[2020-07-05 21:25:11 EDT] Opening EPWave...
Done

Activate Windows
Go to Settings to activate Windows.



Task Group1

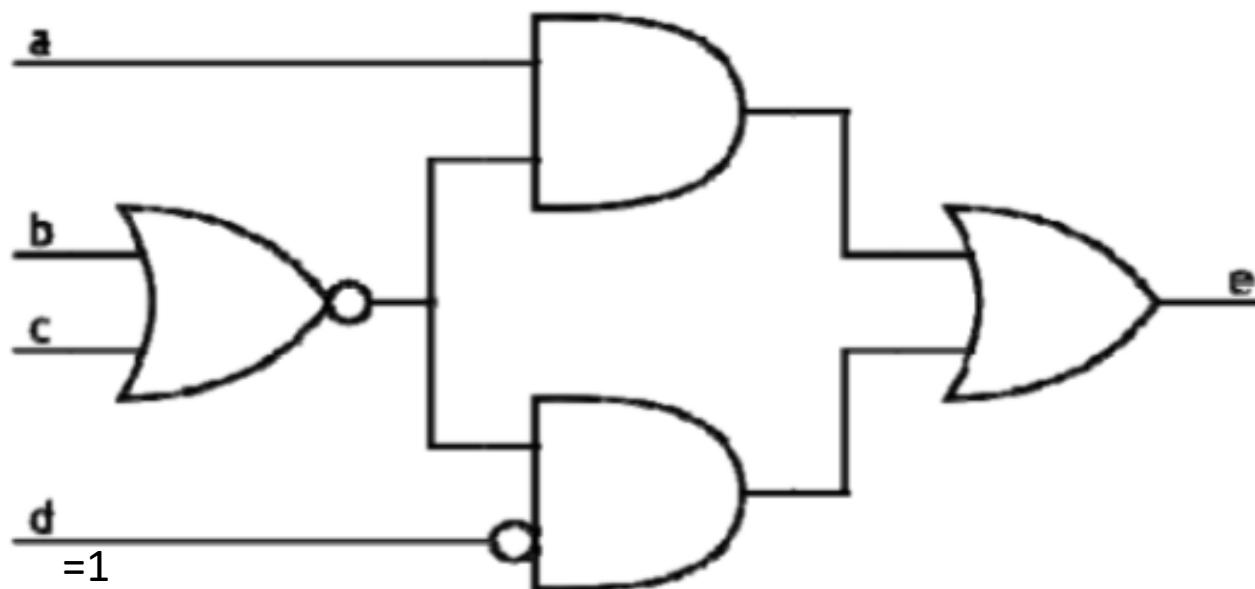
- Write a VHDL Code to Implement the below full-adder circuit using **data flow style**.
- Use **Signal** for the internal connection between gates.
- **Verify your design by simulation**, force different input patterns and check the output.



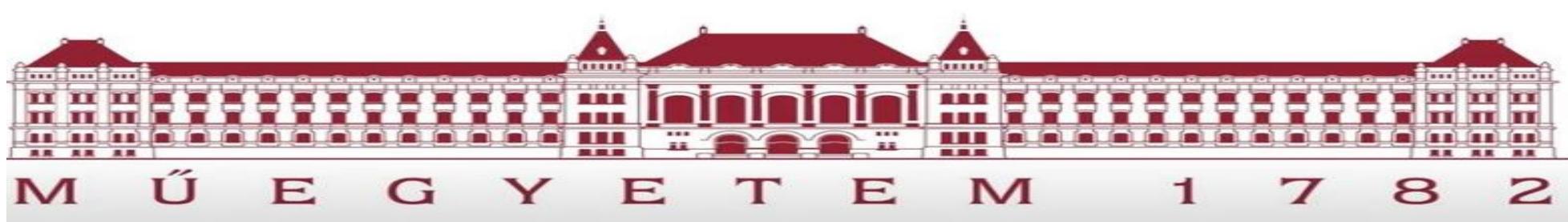
X	Y	Cin	Sum	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Task Group2

- Write a VHDL Code to Implement the below circuit function using **architecture data flow style**.
- Use **Signal** for the internal connection between gates.
- **Verify your design by simulation**, force different input patterns and check the output.



A	B	C	D	E
0	0	0	1	1
0	0	1	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	1	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	0



*Budapest University of Technology and Economics
Department of Electron Devices*

Lec. 1

Introduction to system-level design

Osama Ali

osamaalisalman.khafajy@edu.bme.hu

2 of May 2022



Department of Electron Devices

Microelectronics - BMEVIEEAB00-EN

Outline

- Analog Vs. Digital
- Synchronous and asynchronous
- What is a System
- Specification and Implementation of digital system
- System Level design
- Managing a complex design
- Abstraction Levels
- Design flow of digital systems
- Small intro about VHDL



ANALOG COMPUTER

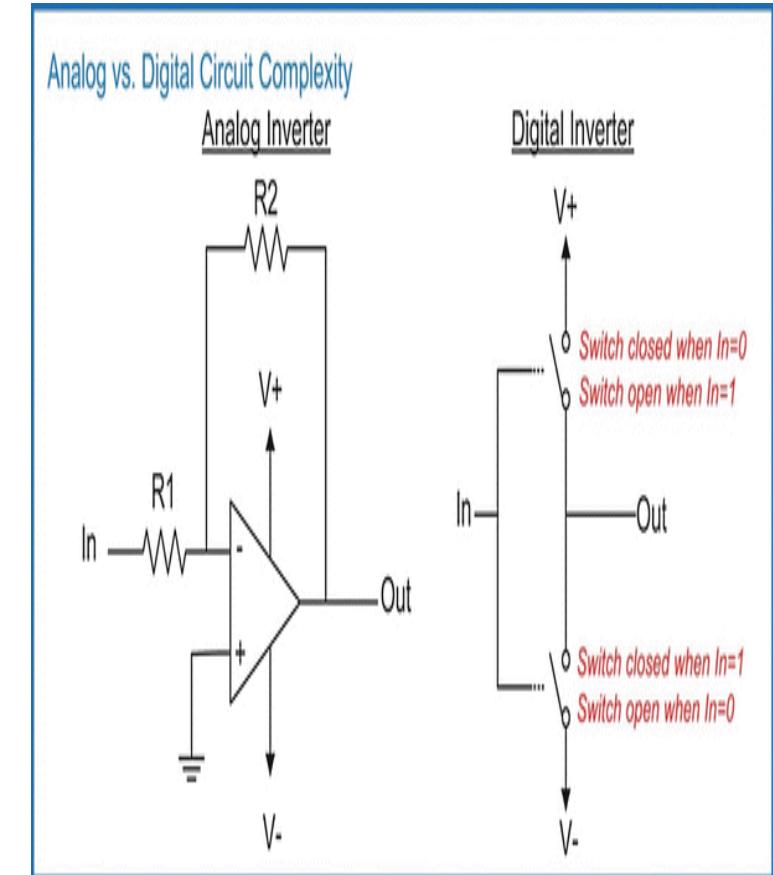
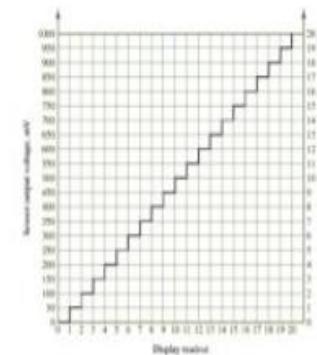
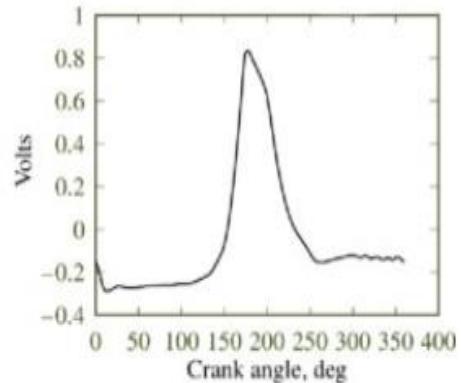
VS

DIGITAL COMPUTER



Analog Circuits vs. Digital Circuits

- An **analog** signal is an electric signal whose value varies continuously over time
- A **digital** signal can take on only finite values as the input varies over time

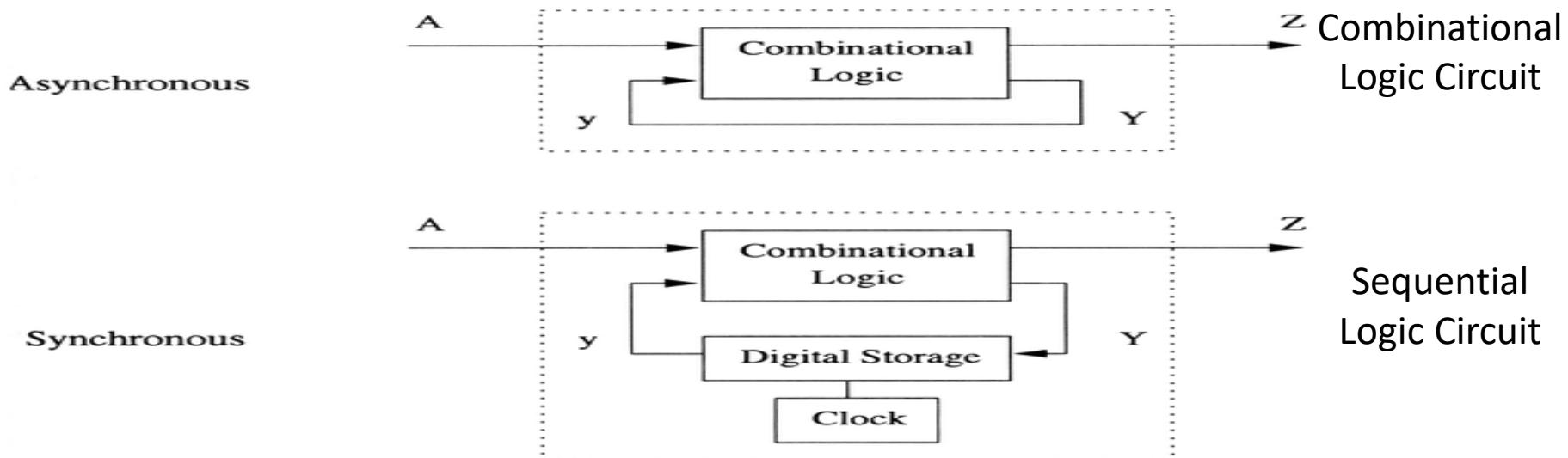


Analog Circuits	Digital Circuits
Analog circuits operate on continuously variable signals also known as Analog Signals.	Digital Circuits operate on discretely variable signals or Digital signals i.e. the signal exists only in two levels: 0 and 1 (binary digital signaling).
Depending the efficiency and precision, it is quite difficult to design Analog Circuits.	Digital Circuits are relatively easy to design with many automated tools available for various stages of design and analysis.
When interacting with the physical world, analog circuits can directly accept the signals from outside as the data is already analog.	If a digital circuit has to acquire data from physical world, the analog signals must be converted to digital signals first.
As there is no need for data conversion, there is ideally no loss of information .	During the process of converting analog signals to digital signals, there might a significant amount of data loss , which can result in loss of information.
If precision and accuracy are not a criterion , then analog circuits can be simple and inexpensive.	Even with simple design techniques and at low cost, the digital circuits can provide good accuracy and precision .
Due to the lack of skilled engineers and the complexity of the designs, analog circuits can turnout to be quite expensive .	Advanced Integrated Circuits technologies and many other factors help the digital circuits to be reliable, lower in cost and smaller in size.



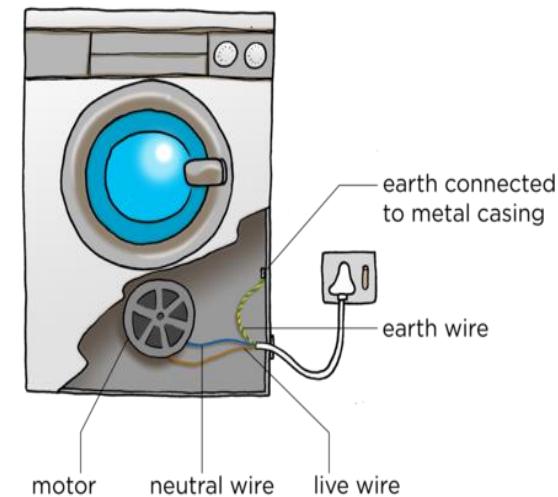
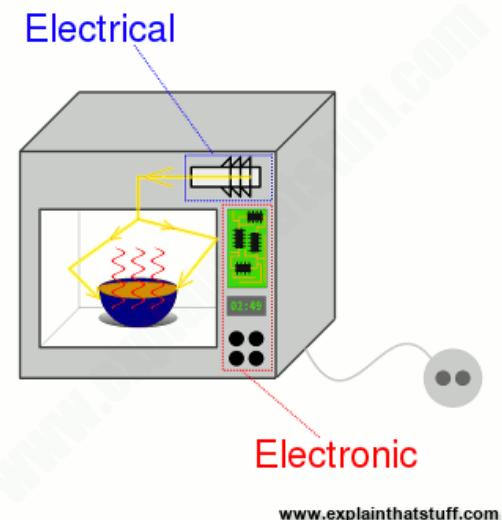
(Synchronous and asynchronous)

- Sequential logic is a type of logic circuit whose output depends not only on the present value of its input signals but on the sequence of past inputs, the input history as well.
- On the other hand, the combinational logic, whose output is a function of only the present input.
- That is, sequential logic has state (memory) while combinational logic does not.



What is a System

- A system is a group of **interacting** or **interrelated** elements that perform according to a set of **rules**.
- A system, **surrounded** and **influenced** by its **environment**, is described by its **boundaries**, **structure** and **purpose** and expressed in its functioning



Digital system specification and Implementation

- **Specification** of a system is the description of its **function** and other **characteristics required** for it, for example **speed**, **cost** and **power**. On other hand you need to define **input** and **output port** and **signals**.
- **Implementation** means how the system is constructed from **smaller** and **simpler components** called **modules**. The modules can vary from simple **Gates** to complex **processors**



System Level design

- In same cases a system can be a **Heterogenous/Mixed system** (sub-systems).
- It could be Multi-Domain or Multi Physics System like System that contains **analog** and **digital** components and **non-electrical** components
- Domain/ Areas
 - electrical - digital and analog
 - magnetic - also related to electrical domain
 - mechanical rotational
 - mechanical translational
 - hydraulic/fluidic
 - radiation/optical
 - thermal



Cont.

- System level design is a **methodology** where the designer (engineer) accounts for all the components of a system.
- Traditionally, you may have a specialized designer for each domain of the system each one of them know little about others domain.
- This can lead to **long** and **frustrating** design iterations to reach a successful design
- The solution for that a tool not only provides multiple levels of abstraction but also it have the ability for integration between hardware and software that will be a very good system-level design tool.



Cont.

- This tool must have ability to express
 - mathematical equations
 - execute them in a real-time

Operation:

Algebraic symbols:

OR (logical sum)	AND (logical product)	NOT (negation)
---------------------	--------------------------	-------------------

$$\begin{array}{l} X + Y \\ X \vee Y \\ X \cup Y \\ X \text{ or } Y \end{array}$$

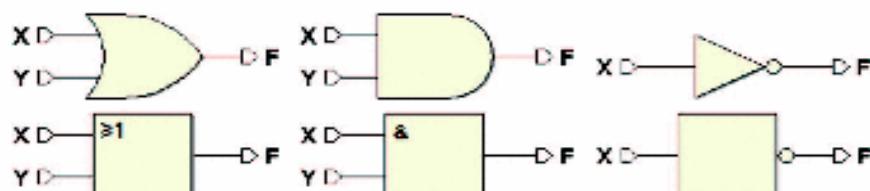
$$\begin{array}{l} X \cdot Y = XY \\ X \wedge Y \\ X \cap Y \\ X \text{ and } Y \end{array}$$

$$\begin{array}{l} \overline{X} \\ !X \\ -X \\ \text{not}(X) \end{array}$$

Truth table:

X	Y	$X + Y$	X	Y	$X \cdot Y$	X	\overline{X}
0	0	0	0	0	0	0	1
0	1	1	0	1	0	1	0
1	0	1	1	0	0	0	1
1	1	1	1	1	1	1	0

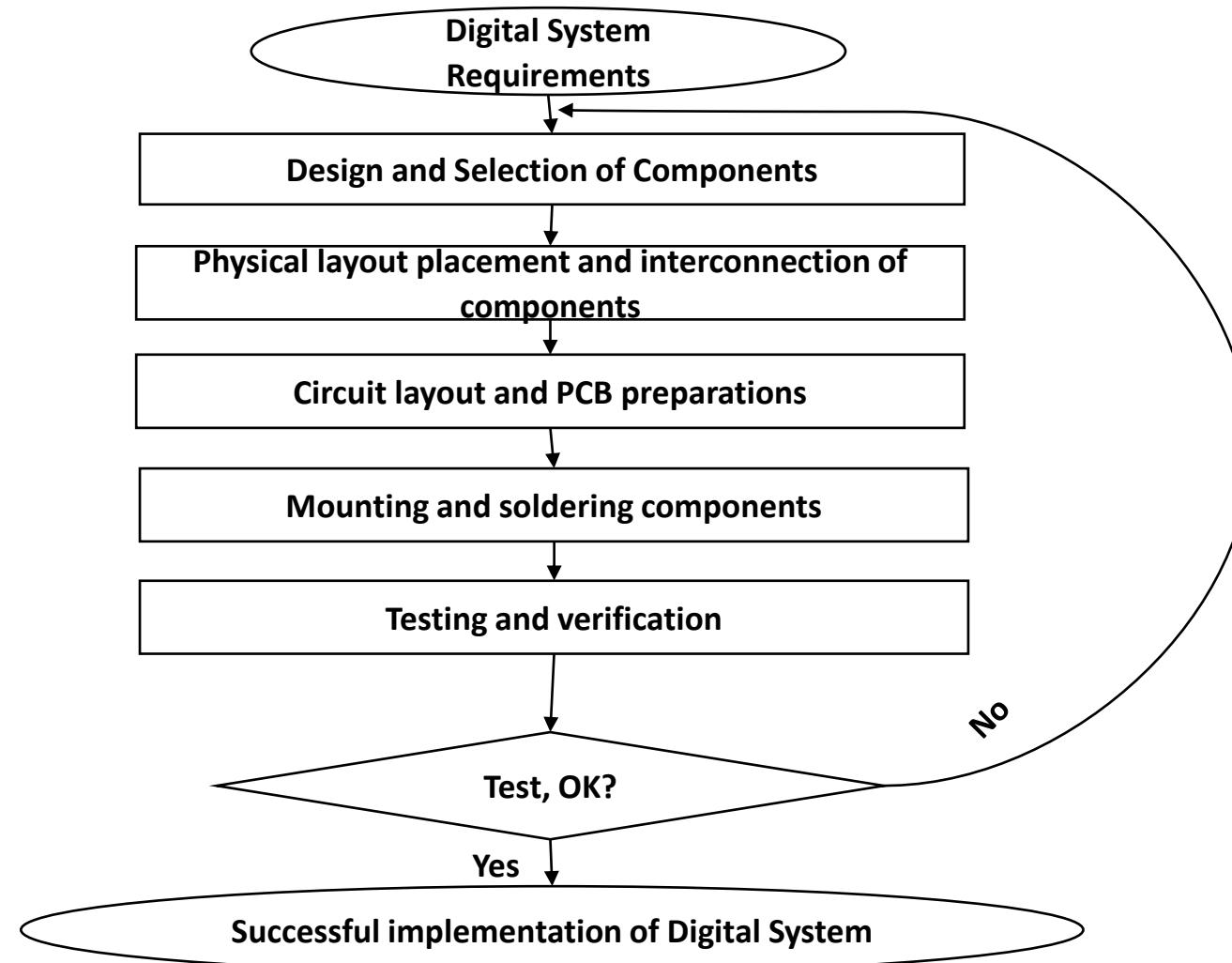
Circuit diagram symbols:



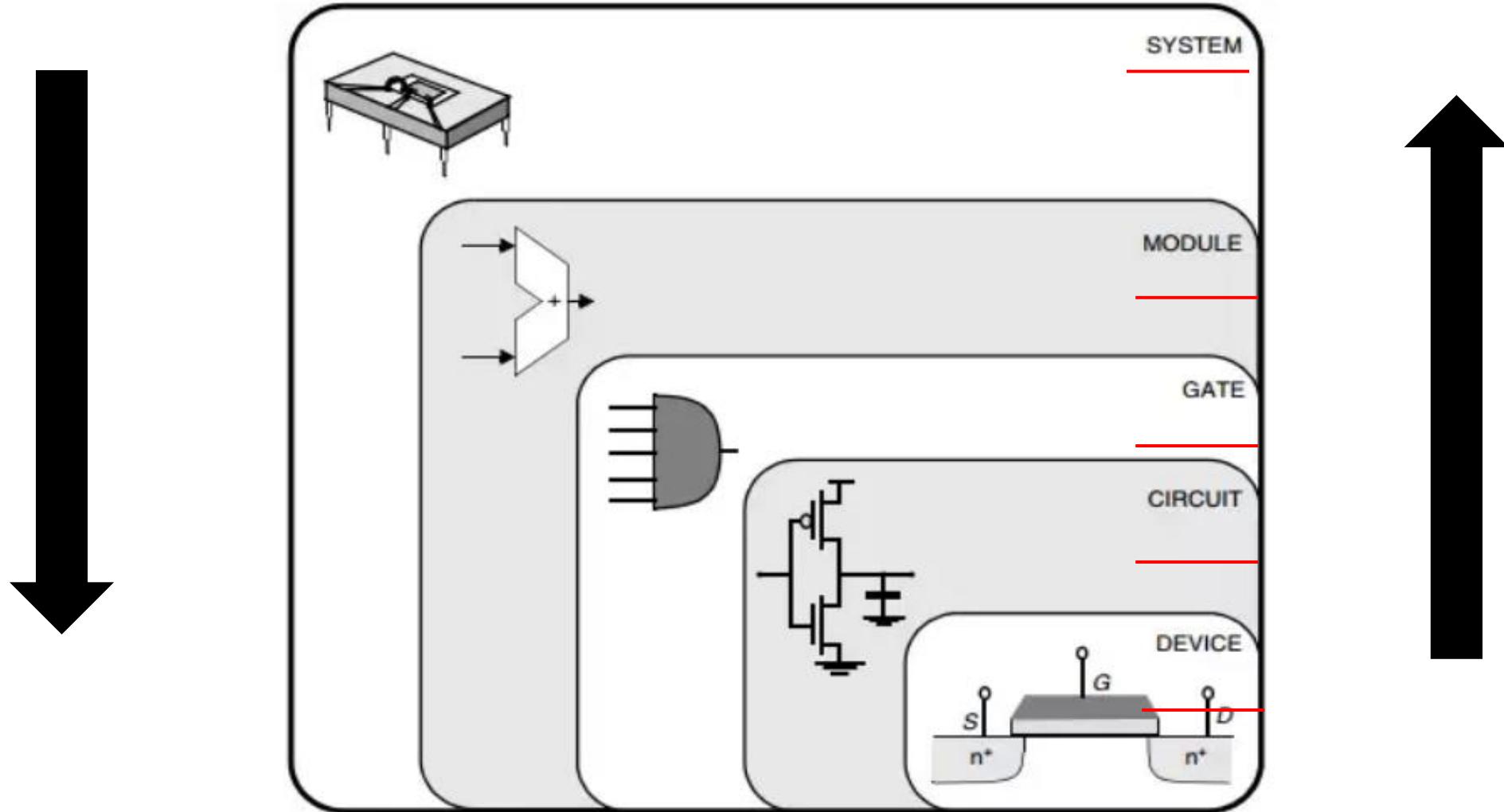
Managing a complex design



Sequence of steps in conventional digital design

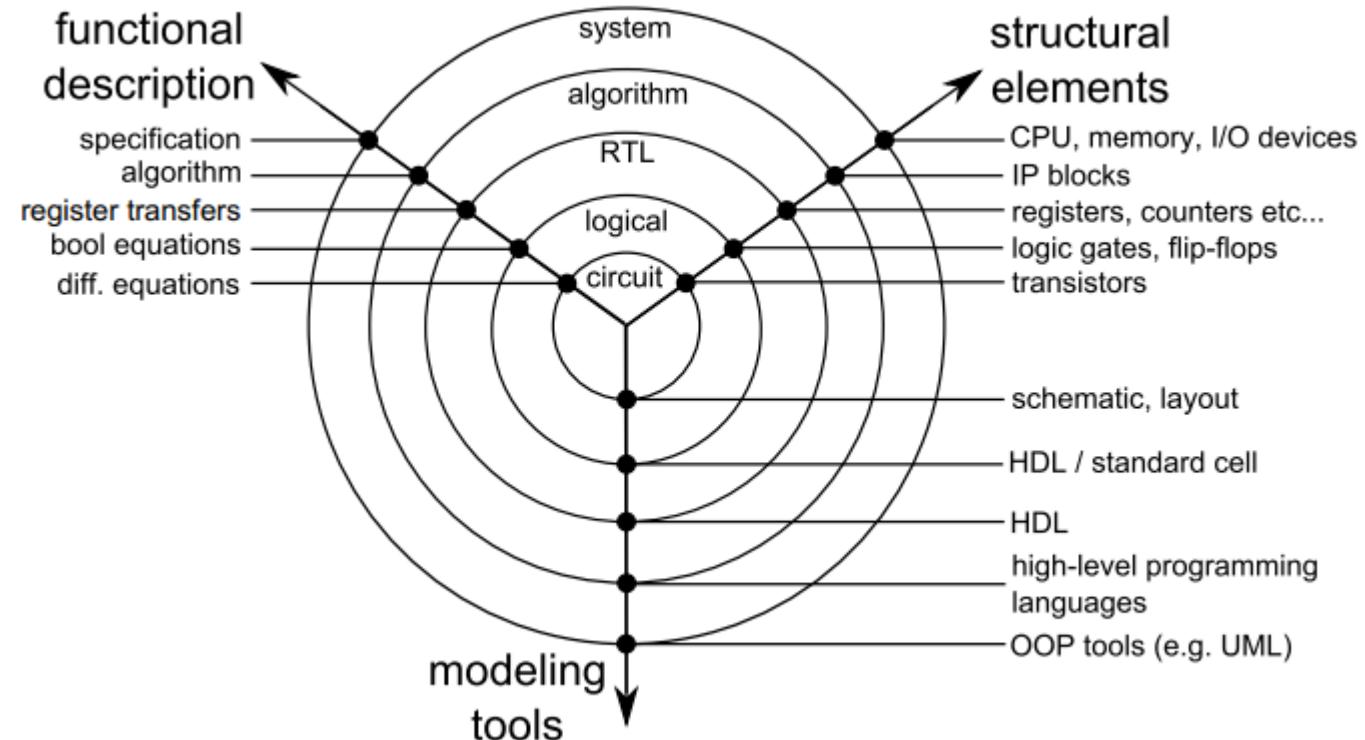


Abstraction Levels in Digital System



Abstraction Levels\Gajski-Kuhn Y-diagram

- The GK diagram is an expressive representation of the abstraction levels.



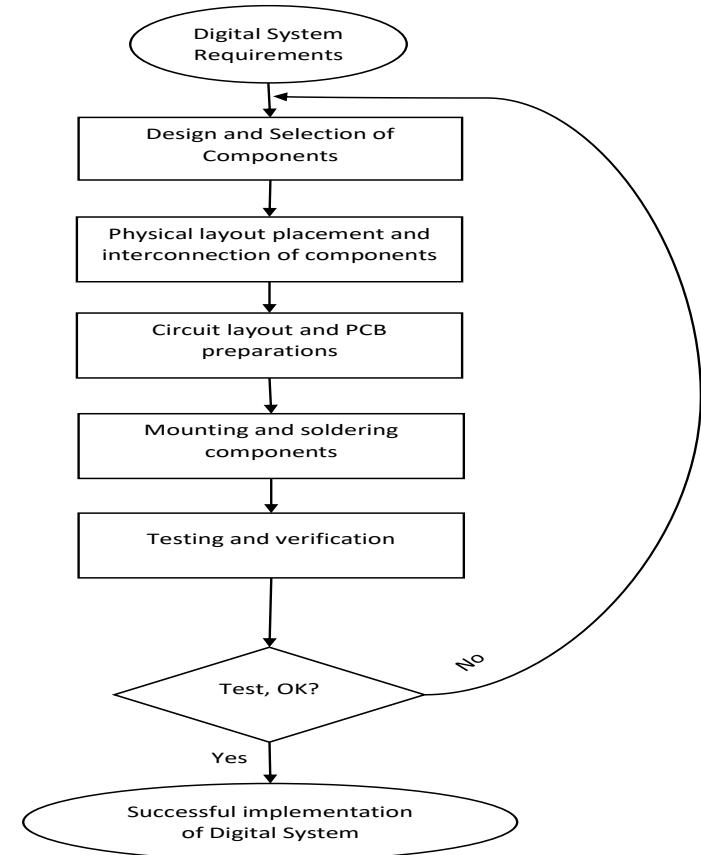
Ref: Dr. Péter Horváth

Different abstraction levels

- **System** – defining design **partitions** and their **interfaces**. In the case of VLSI design, these can be description languages like VHDL, Verilog or SystemC.
- **Algorithm** – behavioral modeling with high-level programming languages. The tool that could be used here C\C++.
- **RTL (register-transfer level)** – defining "**microarchitecture**". So, it capture the functionality in the ships of register-transfer operations on ALUs, registers, multiplexers
- **Gate** – defining the behavior of **RTL** components with **Boolean Equations**
- **Circuit** – implementing the behavior of the logic gates with **transistor-based structures**

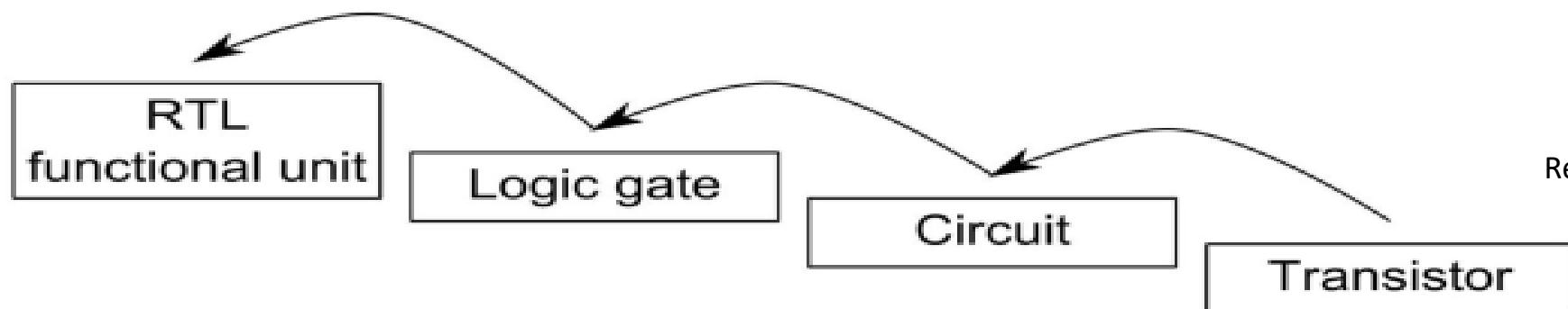
Design flow of digital systems

- Not the conventional one
- Mainly there are two approaches that can be taken out in designing a digital system
 - Bottom-up design method
 - Top-down design method



Bottom-up design method

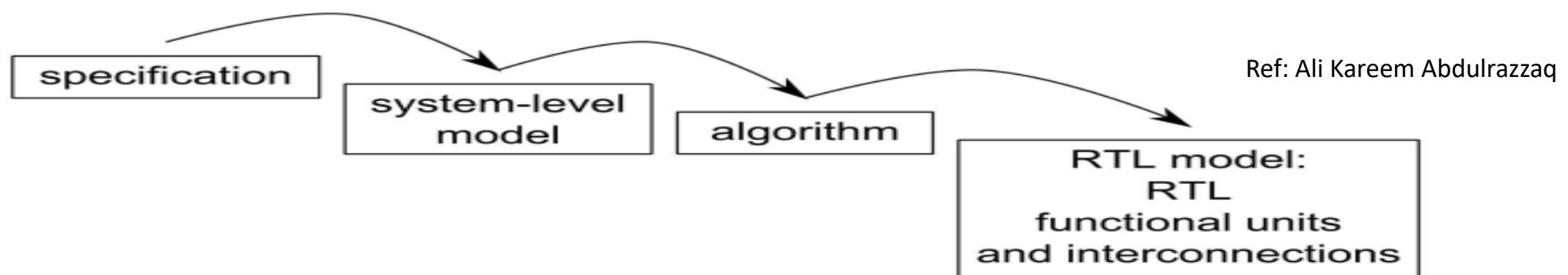
- In the bottom-up approach the designer creates basic functional units with very simple tasks. Once a sufficient set of elementary functionalities is constructed, a more complex model can be prepared with the combination of the simple ones. The design process stops when the increasingly complex model is able to implement the desired functionality defined in the specification.



Ref: Ali Kareem Abdulrazzaq

Top-down design method

- The design process starts with a high-level representation of the system.
- The high-level model includes partitions (subsystems) with a specific task.
- During the design process the implementations of the subsystems are elaborated; they are split into components with more specific sub-tasks and more detailed implementations.
- The process stops when the components of the refined design are simple enough to substitute them with an existing model (practically with an RTL functional unit).



Small introduction in the descriptive language (VHDL)

Introduction to VHDL

Languages for designing hardware

- Higher-level computer languages are used to describe algorithms
 - Sequential execution
- Hardware Description Languages (HDL) are used to describe hardware
 - Not for programming, but for designing hardware
 - Most popular: VHDL, Verilog
 - Parallel (concurrent) execution
 - **Instructions are all executed at the same time**

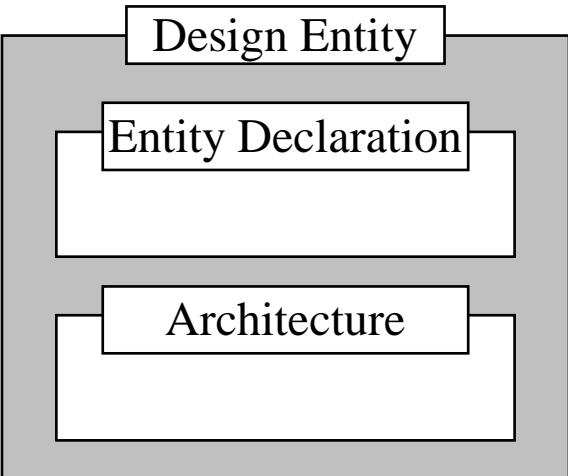


Abstraction levels in VHDL

- VHDL is rich in language abstractions, in addition to which the language can be used to describe different abstraction levels
 - Dataflow
 - Structural
 - Behavioural
- Abstraction levels are means of concealing/hidden details
- The design of VHDL components described on higher abstraction levels can be **technology-independent**
 - It is usually a requirement to determine the abstraction level at which the information is to be described
 - If a short development time is required, a high abstraction level should be chosen as the way of modelling

- **Dataflow** : The dataflow view describes a network of signals in which the flow of signal values is supervised by a set of control elements
- **Structural** : Structural design is the closest to schematic capture and utilizes simple building blocks to compose logic functions
- **Behavioural** : It describes hardware behaviour in terms of **circuits** and **signal**
 - It accurately models what happens on the inputs and outputs of the black box
 - No matter what is inside and how it works
 - Function is defined algorithmically with timing and node loading largely ignored

Primary language abstraction



- The primary abstraction is the design entity
- It is the basic unit of hardware description
- The design entity can represent a **cell, chip, board, or subsystem**
- Aspects of modelling a system: *interface* and *function*
- An entity declaration defines the interface between the entity and the environment outside of the design entity
- An entity can be linked to several architectures
 - E.g., one architecture may model an entity at a behavioural level, while another architecture may be a structural model

Design entity and component

- Entity is a component of a design
 - Component reusability
 - A component can be saved in a component library this will enabling it to be copied as many times as required
 - Ports
 - The inputs and outputs of the circuits
 - They are special programming objects
 - Ports are signals
 - Ports are the means used by the circuit to communicate with the external world, or with other circuits
 - Each port must be declared to be of a particular type

Fundamental terms

Architecture

- It is a code that specifies the behaviour of a component
- There can be more than one architecture for an entity
- Each architecture would specify some different modelling levels

Configuration

- It specifies the entities, architectures to use for components within a particular model
- In hardware design tools there is not configuration at all, or the integrated tool has configuration functionality

Package

- A collection of type definitions, procedures, functions, and component declarations

Library

- A collection of entities, architectures, configurations, packages
- Most designs import library modules

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
ENTITY HALFADDER IS
    PORT(U, V : IN STD_LOGIC;
         SUM, CARRY : OUT STD_LOGIC);
END HALFADDER;

ARCHITECTURE RTL HALFADDER OF HALFADDER IS
BEGIN
    SUM <= U XOR V;
    CARRY <= U AND V;
END RTL HALFADDER;
```

Example of entity declaration and architecture

```
entity andGate is
    port (a, b: in bit;
          q: out bit);
end andGate;

architecture str of andGate is
    -- declarations here
begin
    -- statements here
end str;
```

- Two names are specified in the architecture declaration
 - *Component name* describes which entity the architecture belongs to
 - *Architecture name*
- The entity name in the architecture has to be the same as the identifier (entity name) of the corresponding entity declaration

- The highlighted (bold face) words are key words in VHDL
- The other words are user given





Budapest University of Technology and Economics
Department of Electron Devices

BSc Course in Microelectronics

LABORATORY PRACTICE: THERMAL ISSUES OF INTEGRATED CIRCUITS

- Read through this summary carefully and answer the questions listed on the last page (similar questions are expected at the entry exam)
- Visit edu.eet.bme.hu, download and watch the video guide about how to use the THERMAN software.

Heat transfer methods

It is well known that heat transfer may be occurred by three different physical methods: *conduction*, *convection* and *radiation*. As a result of the heat transfer heat current is developed. Heat current density, denoted by \mathbf{q} is the heat current flowing through the unit area during the unit time. It has a dimension of $[W/m^2]$. The heat current density values of the different heat transfer mechanisms can be calculated as follows:

The conductive heat transfer

$$q = -\lambda \cdot \text{grad}T$$

where T is the temperature, λ $[W/m^2K]$ is the heat transfer coefficient. Note, that λ is a material property.

The convective heat transfer

$$q = h(T - T_\infty)$$

where T is the temperature of the surface which loses heat while T_∞ is the ambient temperature, which is treated to be constant, h is the convective heat transfer coefficient.

Convective heat transfer can be divided into two typical forms:

- *Natural convection* is a mechanism, or type of heat transport, in which the fluid motion is not generated by any external source (like a pump, fan etc.) but only by density differences in the fluid (liquid or gas) occurring due to temperature gradients. In natural convection, fluid surrounding a heat source receives heat, becomes less dense and rises. The surrounding, cooler fluid then moves to replace it. This cooler fluid is then heated and the process continues, forming a convection current; this process transfers heat energy from the bottom of the convection cell to top.
- *Forced convection* is a mechanism, or type of transport in which fluid motion is generated by an external source (like a pump, fan etc.). It should be considered as one of the main methods of useful heat transfer as significant amounts of heat energy can be transported very efficiently and this mechanism is found very commonly in everyday life, including central heating, air conditioning, steam turbines and in many other machines.

Note: The current generated by forced convective heat transfer is caused by an external potential difference (e.g. pressure difference), just like the drift current in the semiconductors is generated by the potential of external voltage.

The Radiation

$$q = \varepsilon\sigma T^4$$

where $\sigma = 5.67 \cdot 10^{-8} W/m^2 K^4$ the Stefan-Boltzmann constant, ε is the emissivity coefficient. If the surface is a perfect mirror ε equals to zero, while the perfect black body has an ε of 1.

In case of investigation the thermal transfer behavior of electronic devices, the following rule of thumbs may be applied:

- Within the IC package (chip, chip assembly, bonding wires or bonding balls) only conductive heat transfer should be considered.
Heat generation within the chip depends on the function. Small signal analog circuits (e.g. operational amplifiers) typically generate heat below 500 mW. Simple digital circuits (TTL or CMOS logic gates) never exceed the 2 W dissipation. However analog switches may dissipate tens of Watts, as well as complex digital circuits, like a CPU.
- Outside the package both conduction and convection are present. Heat transfer occurs through the package (natural convection) and through the device pins across the PCB (conduction).
High power devices (high current switching transistors, CPUs, GPUs) request cooling in order to keep the temperature of the device in the safe operation area. In case of non-military devices this temperature typically falls in the 0 to 70°C range.
- Without a heatsink only 2 – 5W of power can be dissipated. With a heatsink the dissipable power may increased up to 10W (mainly natural convection).
- Higher powers (up to 100W) can only dissipated by using cooling fans. In this case the forced convection dominates over the heat transfer.
- Radiation should be take into consideration when heat transfer occurs through big surfaces (e.g. LED lighting panels).

Boundary conditions

Heat transfer issues can be investigated by computer simulation methods. The problem should be isolated at first which consists the following restrictions:

- It should be decided, what is the physical volume to be considered. With other words, where are the borders of the simulation domain (volume) where the effects of the outside world can be neglected therefore the domain is separated from the outside world.
- It should be decided, how the borders of the simulation domain connect to the outside world. These are called the boundary conditions. In case of thermal investigation these are typically Neumann, Dirichlet or Robin cases.
- Note, that in many cases despite the simulation gives some results the real world measurements may show quite different values. These errors are usually consequences of badly changed simulation domain or boundary conditions rather than the fault of the simulation software itself.

The boundary conditions can be divided into three groups

First order or Dirichlet boundary condition. The temperature function is prescribed at the boundary. If $T(x, t) = \text{const}$ the boundary condition is isometric.

$$T(x, t) = f(x, t) \quad (1)$$

Second order or Neumann boundary condition. The heat current density function is prescribed at the boundary. If $q(x, t) = 0$ the boundary condition is adiabatic (no heat loss from the system).

$$-\lambda \frac{\partial T}{\partial n} = q(x, t) \quad (2)$$

where n is the normal vector.

Third order or Robin boundary condition. The convection heat current density function is prescribed at the boundary.

$$-\lambda \frac{\partial T}{\partial n} = h(T(x, t) - T_{\infty}) \quad (3)$$

Compact models

There are some cases when simulations are not necessary to describe the thermal behavior of a system, obviously not in simple cases (e.g. a simple transistor in the circuit dissipates 5W, how big heatsink is requested?). Complex heat transfer methods may be simplified under given circumstances. If the path of the heat transfer may be treated to be one dimensional, the conduction heat transfer can be described by thermal resistances. Consider a brick shaped heat conductor body with a temperature difference between its ends of $\Delta T = T_H - T_C$. Because of the temperature gradient heat current of P develops (remember, how the charge carrier concentration gradient causes electric current!).

$$R_{th} = \frac{\Delta T}{P} = \frac{L}{\lambda A} [K/W] \quad (4)$$

where $L[m]$ is the length and $A[m^2]$ is the cross-sectional surface of the brick.

Obviously the temperature of a body cannot change immediately, because the body itself should be filled of heat. The ability of a body to store heat is the heat capacitance. To rise up the temperature of the body by ΔT , a sum of W energy is requested:

$$C_{th} = \frac{W}{\Delta T} = c_v \cdot A \cdot L \quad (5)$$

where $c_v[W/(m^3 * T)]$ is the volumic heat capacity.

The thermal time constant is analogous the time constant of an R-C circuit as follows:

$$\tau_{th} = R_{th} C_{th} \quad (6)$$

The heat which is generated within the chip structure should be transferred to the ambient otherwise the temperature of the chip rises above the safe operation area. It depends on two factors:

- How the heat can be transferred from the chip (or more precisely, the place where the heat is generated: the p-n junction) to the chip package (or case): R_{thjc} (read: thermal resistance junction to case).
- How the heat can be transferred towards from the case to the ambient (it depends on the size of the heatsink, convective heat transfer etc.): R_{thca} (read: thermal resistance case to ambient)

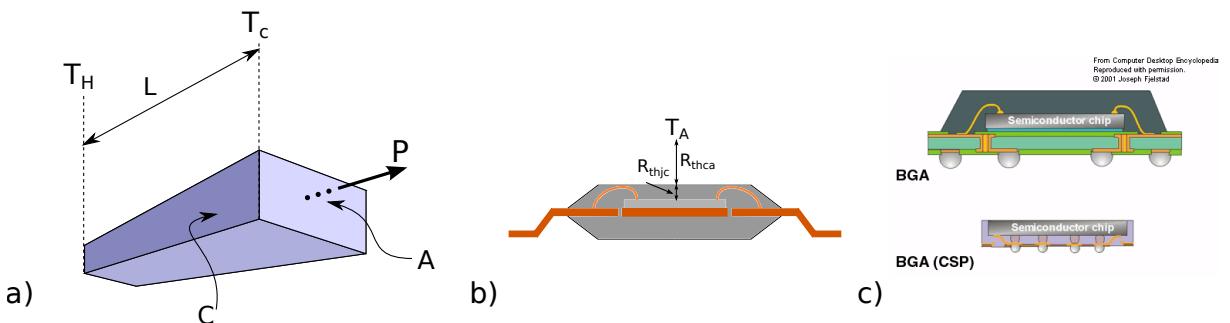


Figure 1: a) Understanding thermal resistance b) DIP package cross section and R_{thca} , R_{thjc} values c) modern BGA and BGA (CSP) flip chip cross sections

R_{thjc} and R_{thca} are shown on the device's datasheets. Let's consider a device dissipating 100mW in DIP package, having $R_{thjc} = 37K/W$ and $R_{thca} = 70K/W$. What will be the junction temperature, if the ambient temperature is 25°C?

$$T_j = T_{amb} + P(R_{thjc} + R_{thca}) = 25 + 10.7 = 35.7^\circ C \quad (7)$$

Temperature aware design of integrated circuits

It is a well known fact that the concentration of minority carriers have a strong temperature dependence through the temperature dependence of n_i^2 (where n_i is the intrinsicsic electron concentration).

Based on these effects the forward voltage characteristic of the p-n junctions (diodes, bipolar transistors) show

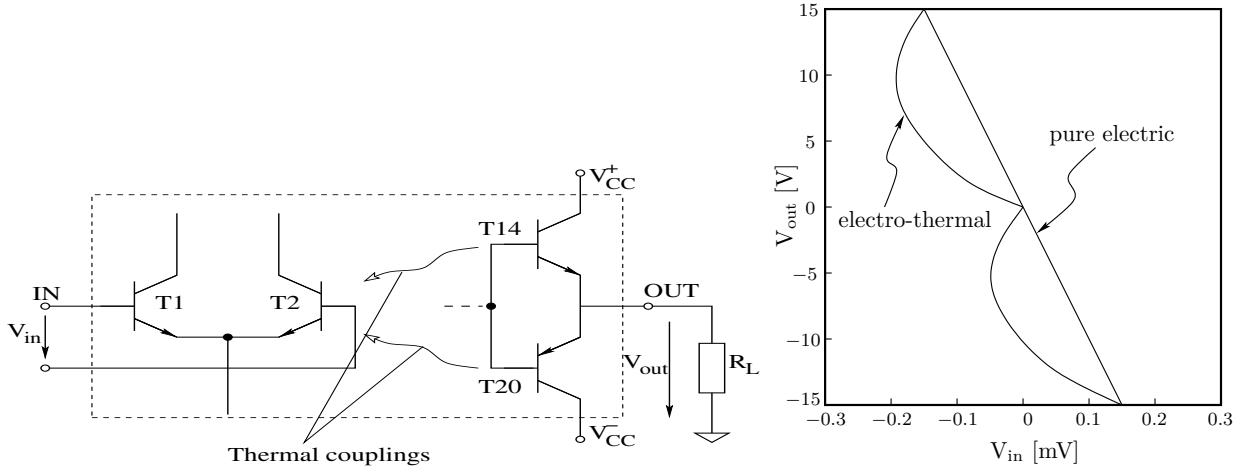


Figure 2: Thermal feedback in an op-amp – a typical situation, where there is a significant difference in the results between “electrical-only” and electro-thermal simulations

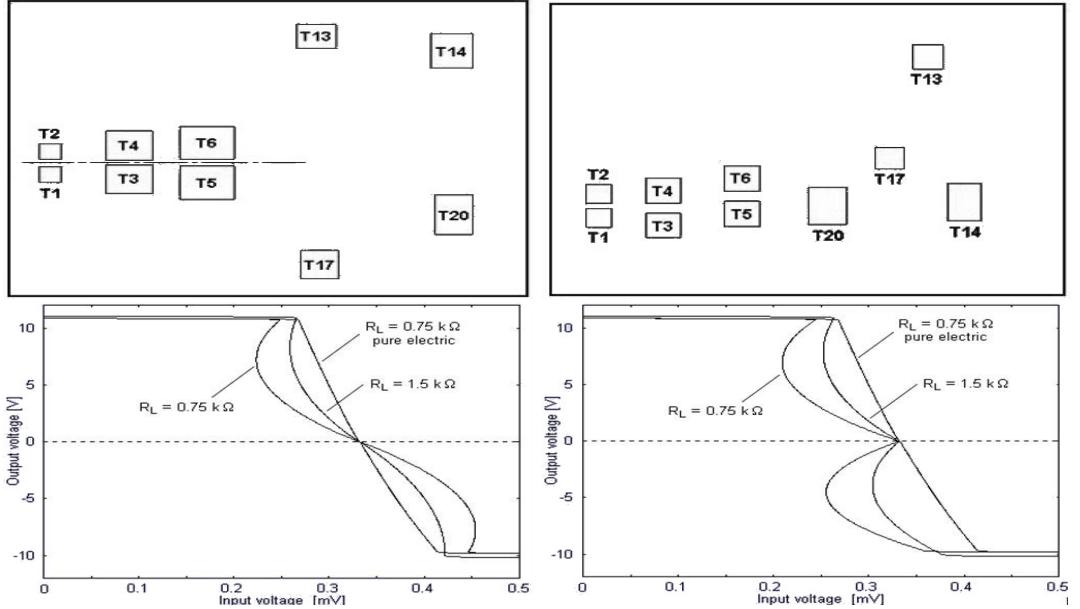


Figure 3: The two different layouts for the μ A741 and transfer characteristics with different loads

also a non-negligible temperature dependence.

The forward voltage of the silicon diode shows about $-2mV/K$ temperature dependence.

Symmetric devices, like operational amplifiers request very good matching components in order to work correctly (imagine a badly balanced teeter: in order to work correctly, the two sides should be equal). As the characteristics of the components have temperature dependence, they should be at the same temperature otherwise they don't match anymore.

The importance of thermal aware design at chip level can be easily demonstrated with the example of a simple op-amp (Fig. 2) – which is an ideal benchmark circuit for electro-thermal circuit simulators because it is highly sensitive to thermal offset. The relatively high dissipation of the output stage (T14, T20) warms up the input stage (T1, T2) which moves the operating points and thus changes the transfer characteristics. This effect strongly depends on the layout of the circuit.

The layout of the same circuitry is shown in two different version which gave us the ability to compare the results at different thermal scenarios. The simulated open-loop transfer characteristics can be seen with different loads. These results demonstrate that the structure of the physical layout has a major effect on the electrical behaviour through thermal coupling. Using the symmetric layout the output characteristics remains symmetric regardless the load applied while the asymmetric layout causes asymmetric output characteristics.

1 Questions

1. What are the main heat transfer methods? Describe the conductive (/convective/radiation) heat transfer and relevant equations.
2. What are the main types of convective heat transfer? What is the main difference between them?
3. What types of heat transfer should be considered at different stages of an electronic system?
4. What are boundary conditions? Why they should be used?
5. Describe the first (/second /third) order boundary condition and relevant equations.
6. What are compact models? Under what circumstances should they be used?
7. Describe the compact model approach of thermal resistance (/thermal capacitance) and relevant equations.
8. What is the thermal time constant and how to calculate it?
9. What are main factors which describe the heat transfer ability of a package?
10. A transistor dissipates 1 W, the ambient temperature is $25^{\circ}C$, $R_{thjc} = 40K/W$. What is the requested value of R_{thca} if the junction temperature should not exceed $60^{\circ}C$?
11. Why the temperature aware design of analog circuits is important?

References

- [1] A. Szalai, Z. Czirkos, V. Szekely: A quasi-SPICE electro-thermal simulator, 18th International Workshop on Thermal investigations of ICs and Systems, 2012, Budapest, Hungary



*Budapest University of Technology and Economics
Department of Electron Devices*

Lec. 2

Introduction to HDL/VHDL Coding

Osama Ali
9 of May 2022



Outline

- Introduction to VHDL
- Abstraction levels
- Design hierarchies
- Data types
- Language elements
- Finite State Machine (FSM)
- Simulation results

Introduction to VHDL

Languages for designing hardware

- Higher-level computer languages are used to describe algorithms
 - Sequential execution
- Hardware Description Languages (HDL) are used to describe hardware
 - Not for programming, but for designing hardware
 - Most popular: VHDL, Verilog
 - Parallel (concurrent) execution
 - Instructions are all executed at the same time

What is VHDL?

- Standardization
 - VHDL: IEEE Std-1076-1987, 1993, 2000 (Digital systems)
 - VHDL-AMS: IEEE Std-1076.1-1999, 2001 (Analog, digital, and mixed-signal systems)
- VHDL features
 - Formal language for modelling a digital circuit
 - Modelling concepts are derived from the operational characteristics of digital circuits
 - Source code is interchangeable among the different tools
- Main steps of a VHDL-based design procedure:
 - Code writing, compiling, simulation, & synthesis
 - Synthesis is a process where a VHDL is compiled and mapped into an implementation technology such as an FPGA or an ASIC. Not all constructs in VHDL are suitable for synthesis.

VHDL vs. Verilog

VHDL	Verilog
All abstraction levels	All abstraction levels
Complex grammar	Simple grammar
Lots of data types	Few data types
User-defined type definition	No user defined type definition
User-defined libraries, packages, configurations	No user-defined libraries, packages, configurations
Full design parameterization	Simple design parameterization
Very consistent language (e.g. strong typing rules). Code behaves exactly the same in every simulator	Less consistent language. If the designer doesn't follow ad-hoc coding styles, it executes differently on different platforms
Case insensitive	Case sensitive

Abstraction levels in VHDL

- VHDL is rich in language abstractions, in addition to which the language can be used to describe different abstraction levels
 - Dataflow
 - Structural
 - Behavioural
- Abstraction levels are means of **concealing/hidden** details
- The design of VHDL components described on higher abstraction levels can be **technology-independent**
 - It is usually a requirement to determine the abstraction level at which the information is to be described
 - If a short development time is required, a high abstraction level should be chosen as the way of modelling

Dataflow

```
1. library ieee;
2. use ieee.std_logic_1164.all;
3.
4. entity half_adder is
5.     port (a, b: in std_logic;
6.             sum, carry_out: out std_logic);
7. end half_adder;
8.
9. architecture dataflow of half_adder is
10. begin
11.     sum <= a xor b;
12.     carry_out <= a and b;
13. end dataflow;
```

Behavioural

```
1. library ieee;
2. use ieee.std_logic_1164.all;
3.
4. entity half_adder is
5.     port (a, b: in std_logic;
6.             sum, carry_out: out std_logic);
7. end half_adder;
8.
9. architecture behavior of half_adder is
10. begin
11.     ha: process (a, b)
12.     begin
13.         if a = '1' then
14.             sum <= not b;
15.             carry_out <= b;
16.         else
17.             sum <= b;
18.             carry_out <= '0';
19.         end if;
20.     end process ha;
21.
22. end behavior;
```

Structural

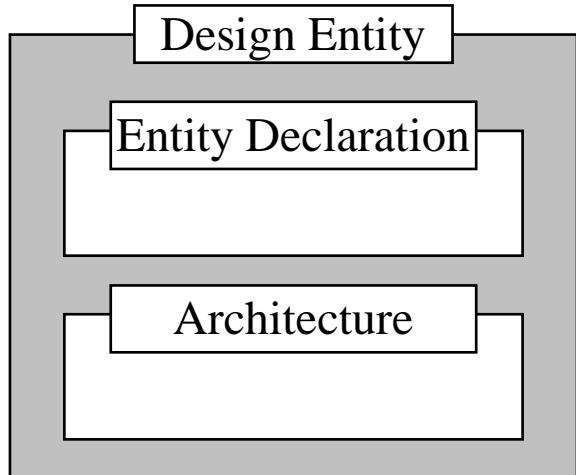
```
1 library ieee;
2 use ieee.std_logic_1164.all;
3
4 entity half_adder is -- Entity declaration for half adder
5   port (a, b: in std_logic;
6         sum, carry_out: out std_logic);
7 end half_adder;
8
9 architecture structure of half_adder is -- Architecture body for half adder
10
11 component xor_gate -- xor component declaration
12   port (i1, i2: in std_logic;
13         o1: out std_logic);
14 end component;
15
16 component and_gate -- and component declaration
17   port (i1, i2: in std_logic;
18         o1: out std_logic);
19 end component;
20
21 begin
22   xor_gate port map (a, b, sum);
23   and_gate port map (a, b, carry_out);
24 end structure;
```

Design hierarchies

Mechanisms to reduce complexity

- Language abstractions are useful to describe complex matters without having to describe small details
- Black box principle
- In some cases there is no need to know how the component is internally structured
 - The designer is usually only interested in
 - Inputs and outputs
 - Specification function
 - Delay times

Primary language abstraction



- The primary abstraction is the design entity
- It is the basic unit of hardware description
- The design entity can represent a **cell, chip, board, or subsystem**
- Aspects of modelling a system: *interface* and *function*
- An entity declaration defines the interface between the entity and the environment outside of the design entity
- An entity can be linked to several architectures
 - E.g., one architecture may model an entity at a **behavioural level**, while another architecture may be a **structural model**

Design entity and component

- Entity is a component of a design
 - Component reusability
 - Ports
 - The inputs and outputs of the circuits
 - They are special programming objects
 - Ports are signals (**But?**)
 - Ports are the means used by the circuit to communicate with the external world, or with other circuits
 - Each port must be declared to be of a particular type

Syntax of the entity declaration

```
entity <identifier_name> is  
port( [signal] <identifier>:[<mode>] <type_indication>;  
      ...  
      [signal] <identifier>:[<mode>] <type_indication>);  
end [entity] [<identifier_name>];
```

The word **signal** is normally left out of the port declaration, as it does not add any information

<mode> = **in, out, inout**

- **in**: Component only read the signal
- **out**: Component only write to the signal
- **inout**: Component read or write to the signal - bidirectional signals

Mode **in** and the name of the **entity** after **end** can also be left out



Example for simplification of an entity declaration

```
entity gate1 is
  port(signal a: in bit;
       signal b: in bit;
       signal c: out bit);
end gate1;
```

```
entity gate1 is -- Identical with the above example
  port( a,b: bit;
        c: out bit);
end;
```



Syntax of the architecture

```
architecture <architecture_name> of <entity_identifier> is  
[<architecture_declarative_part>]  
begin  
<architecture_statement_part>  
end [architecture] [<architecture_name>];
```

- The architecture declaration part must be defined before first begin and can consist of, for example:
 - Types
 - Subprograms
 - Components
 - Signal declarations

CSA: Concurrent Signal Assignment

- Each architecture consists of concurrent statements, which are executed concurrently with respect to simulated time
 - The order of execution of the statements is dependent upon the flow of values and not on the textual order of the program
- CSA: Concurrent Signal Assignment
 - This is a major difference between VHDL and ordinary computer languages Types
 - Simple CSAs
 - Conditional CSAs: **when** statement
 - Selected CSAs: **with – select** statements

```
b <= "1000" when a = "00" else
      "0100" when a = "01" else
      "0010" when a = "10" else
      "0001" when a = "11";
```

```
with a select b <=
      "1000" when "00",
      "0100" when "01",
      "0010" when "10",
      "0001" when "11";
```

Objects

Data types Objects and containers

- Objects are containers for values of a specified type
- Once an object is declared of a certain type, operations can be performed on the object within the bounds set in the type declaration
- If objects of different types are mixed or exceed boundaries set by the type declaration, an error is displayed
- Type of objects
 - *Variables*
 - Sequential objects
 - *Signals*
 - Concurrent objects
 - Unlike variables, signals have an associated time value
 - The signal retains this value until it is assigned a new value at a future point in time
 - *Constants*
- Declarations of signals, variables and constants need specify their corresponding type or subtype

Classes of types

- *Scalar (range) types*
 - Integer types, floating point types, enumeration types
 - E.g. of enumeration types “**type** wireColor **is** (red, black, green);”
- *Composite types*
 - Array, record
 - Access type “These types provide access to objects” **Not supported by synthesis tools**

Scalar types: Integer

- Sets of positive or negative whole numbers
- Their range are machine dependent, typically $\pm 2.147.483.648$ for **32-bit** systems

-- Declaration in standard.vhd package

```
type integer is range -2147483647 to 2147483647;
```

-- User declared integer type

```
type testInteger is range -100 to 100;
```



Scalar types: Floating point

- Defines a collection of numbers that provide an approximation to real numbers
- Problem: it is not possible for hardware to handle **infinitely** long real numbers
 - Declaration in the package standard.vhdl

type real is range -1.7e38 to 1.7e38

-- Bounds are implementation dependent

type half_hour is range 0.0 to 29.99;



Composite types: Array

- A named array is a collection of elements that are of the same type
- Arrays may be configured in one or more dimensions
- Each array element is referenced by one or more index value

```
type array10 is array (0 to 9) of integer;
```

-- Std_ulogic_vector is defined as:

```
type std_ulogic_vector is array (natural range <>) of std_ulogic;
```

-- Std_logic_vector is defined as:

```
type std_logic_vector is array (natural range <>) of std_logic;
```

-- Bit_vector is defined as:

```
type bit_vector is array (natural range <>) of bit;
```

Comments, spaces & labels

- Comments follow (two consecutive dashes or hyphens) '--' and instruct the analyser to ignore the rest of the line
 - The **parser** ignores anything after the two dashes and up to the end of the line in which the dashes appear
 - There are no multiline comments in VHDL
- VHDL is not sensitive to white spaces (**spaces and tabs**)
 - Tabs improve readability, but it is best not to rely on a tab as a space in case the tabs are lost or deleted in conversion

Process

- A *process* is a concurrent statement
 - All statements in a process are executed sequentially until the process is suspended via a *wait* statement
 - Within a process, procedures and functions can partition the sequential statements
- A process can be a single signal assignment statement or a series of sequential statements
- Upon initialization, all processes are executed once
- Processes are executed in a data-driven manner, and activated
 - by events on signals in the process *sensitivity* list or
 - by waiting for the occurrence of specific events using the *wait* statement
- The **sensitivity list** – being next to the process keyword – is a list of those input signals to the component to which the process is sensitive

```
31
32 entity XuLA_2 is
33     Port ( PB1 : in STD_LOGIC;
34             PB2 : in STD_LOGIC;
35             PB3 : in STD_LOGIC;
36             PB4 : in STD_LOGIC;
37             LED1 : out STD_LOGIC;
38             LED2 : out STD_LOGIC;
39             LED3 : out STD_LOGIC;
40             LED4 : out STD_LOGIC);
41 end XuLA_2;
42
43 architecture Behavioral of XuLA_2 is
44 begin
45
46     process (PB1)
47 begin
48         if PB1 = '1' then
49             LED1 <= '1';
50             LED2 <= '0';
51             LED3 <= '1';
52             LED4 <= '0';
53         else
54             LED1 <= '0';
55             LED2 <= '1';
56             LED3 <= '0';
57             LED4 <= '1';
58         end if;
59     end process;
60
61 end Behavioral;
```



An infinite loop

- When a process is completed, it enters in *suspend* mode until the next change in its sensitivity list
- If there is no sensitivity list, then the process will run forever

```
process
begin
    signalName <= '1';
end process;
```



The wait statement

wait_statement ::=

[label :] **wait** [sensitivity_clause] [condition_clause] [timeout_clause] ;

- It is permissible to have several *wait* commands in the same process

wait for time expression;

- This causes suspension of the process for a period of time given by the evaluation of the time expression

wait for 10 ns;

- The simulator waits for 10 ns before continuing execution of the process
- The starting time point of the waiting is significant and not the actual changes of any signal value

The wait statement (continued)

wait on *signal1, signal2, ...;*

- The process is interrupted until the value of one of the signals changes
- An event on any of the signals, causes the process to resume execution at the next statement after the *wait* statement

wait on *a, b;*

- It suspends execution until a change (event) occurs on either signal *a* or *b*

wait until *condition;*

- The Boolean expression is evaluated whenever one of the signals in the expression changes, and the process continues execution when the expression evaluates to true

wait until *a='1';*

- It is satisfied when signal *a* has an event (changes value), and the new value is '1', i.e. a rising edge for signal *a*

wait until *signalName='1'* **for** 10 ns;

- The wait condition is satisfied when *a* changes value or after a wait of 10 ns (regarded as an or condition)

wait;

- The process is permanently suspended



Sensitivity list vs. wait statement

- If a process has a sensitivity list, then it cannot contain a *wait* statement
- Using the sensitivity list is identical to a *wait on* statement at the end of the process, if the group of signals are the same
 - The both processes are executed at first time
 - The both processes are triggered each time that signal *s1* or *s2* changes value

```
p0: process (s1, s2)
begin
  if s1>s2 then
    q<='1';
  else
    q<='0';
  end if;
end process;
```

```
p1: process
begin
  if s1>s2 then
    q<='1';
  else
    q<='0';
  end if;
  wait on s1, s2;
end process;
```

Examples of wait statement

signal a: in bit; c1, c2, c3, c4, c5, c6, c7: out bit;

-- Example 1

```
process (a)
begin
  c1<= not a;
end process;
```

-- Example 2

```
process
begin
  c2<= not a;
  wait on a;
end process;
```

-- Example 3

```
process
begin
  wait on a;
  c3<= not a;
end process;
```

-- Example 4

```
process
begin
  wait until a='1';
  c4<= not a;
end process;
```

-- Example 5

```
process
begin
  c5<= not a;
  wait until a='1';
end process;
```

-- Example 6

```
process
begin
  c5<= not a;
  wait for 10 ns;
end process;
```

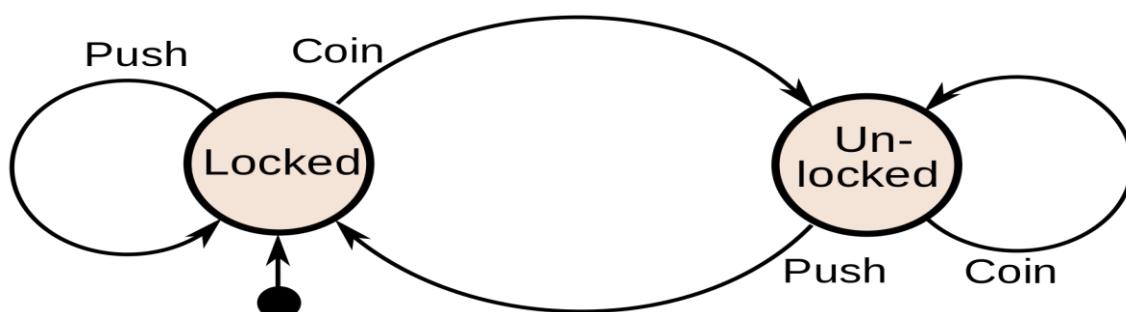
-- Example 7

```
process
begin
  c5<= not a;
  wait until a='1' for 10 ns;
end process;
```

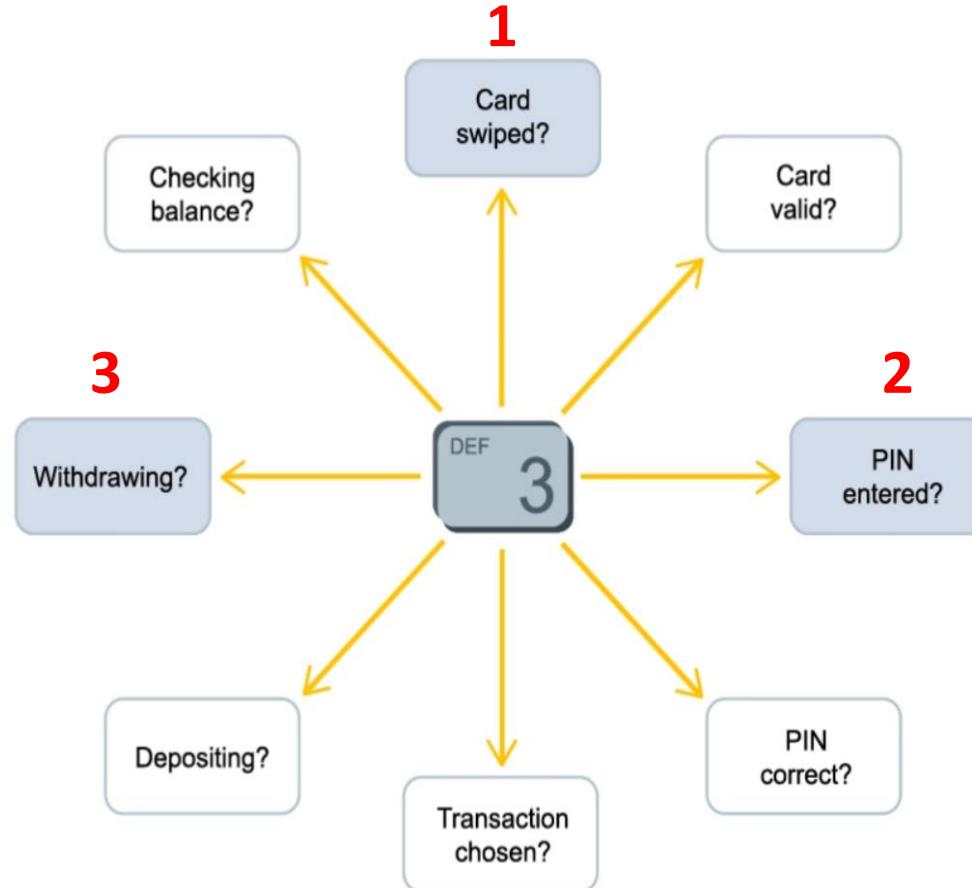
- Note that the process does not wait for event at the signal assignment

FSM : Finite State Machine

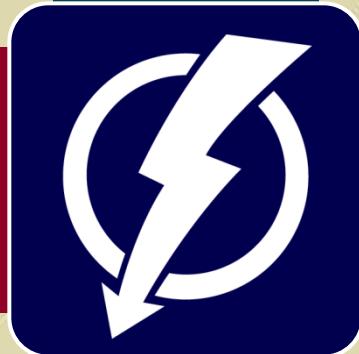
- A sequential circuit is a circuit with memory
- FSM is a mathematical model of the sequential circuit with discrete inputs, discrete outputs and finite number of internal configuration or states.
- So, it is special modelling technique for sequential logic circuit.
- Example of FSM



FSM: example and benefits

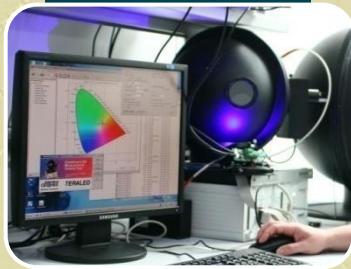


```
if (card_swiped){  
    if (pin_entered){  
        if (pin_validated){  
            if (transaction_selected){  
                if(cash_withdrawal){  
                    if(amount_selected){  
                        confirm_amount(a,b,c);  
                    }  
                    else {  
                        continue_entry(d,e,f);  
                    }  
                }  
                else if (depositing_cash){  
                    if(amount_selected){  
                        confirm_amount(g,h,i);  
                    }  
                    else {  
                        continue_entry(j,k,l);  
                    }  
                }  
                else if (depositing_check){  
                    if(amount_selected){  
                        confirm_amount(m,n,o);  
                    }  
                    else {  
                        continue_entry(p,q,r);  
                    }  
                }  
            }  
        }  
    }  
}
```



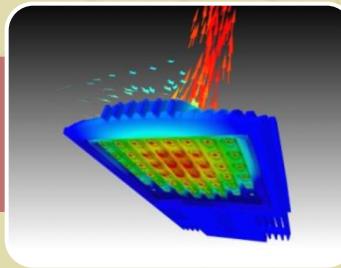
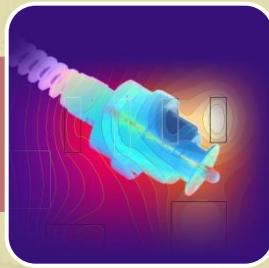
Budapest University of
Technology and
Economics

Department of Electron Devices



Microelectronics: Power LEDs

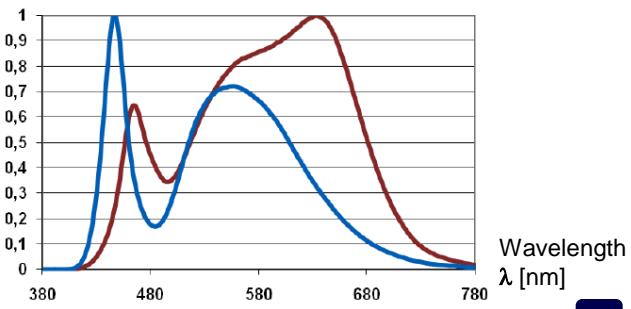
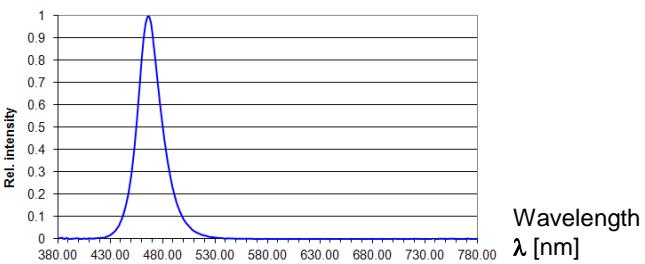
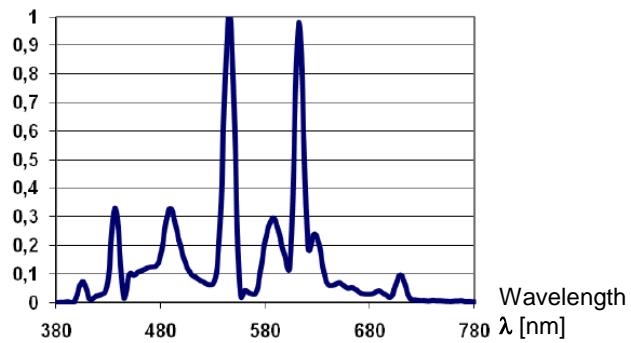
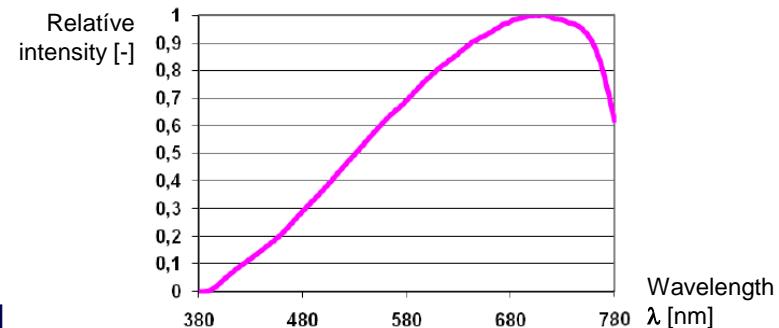
Prof. András Poppe, BME-EET



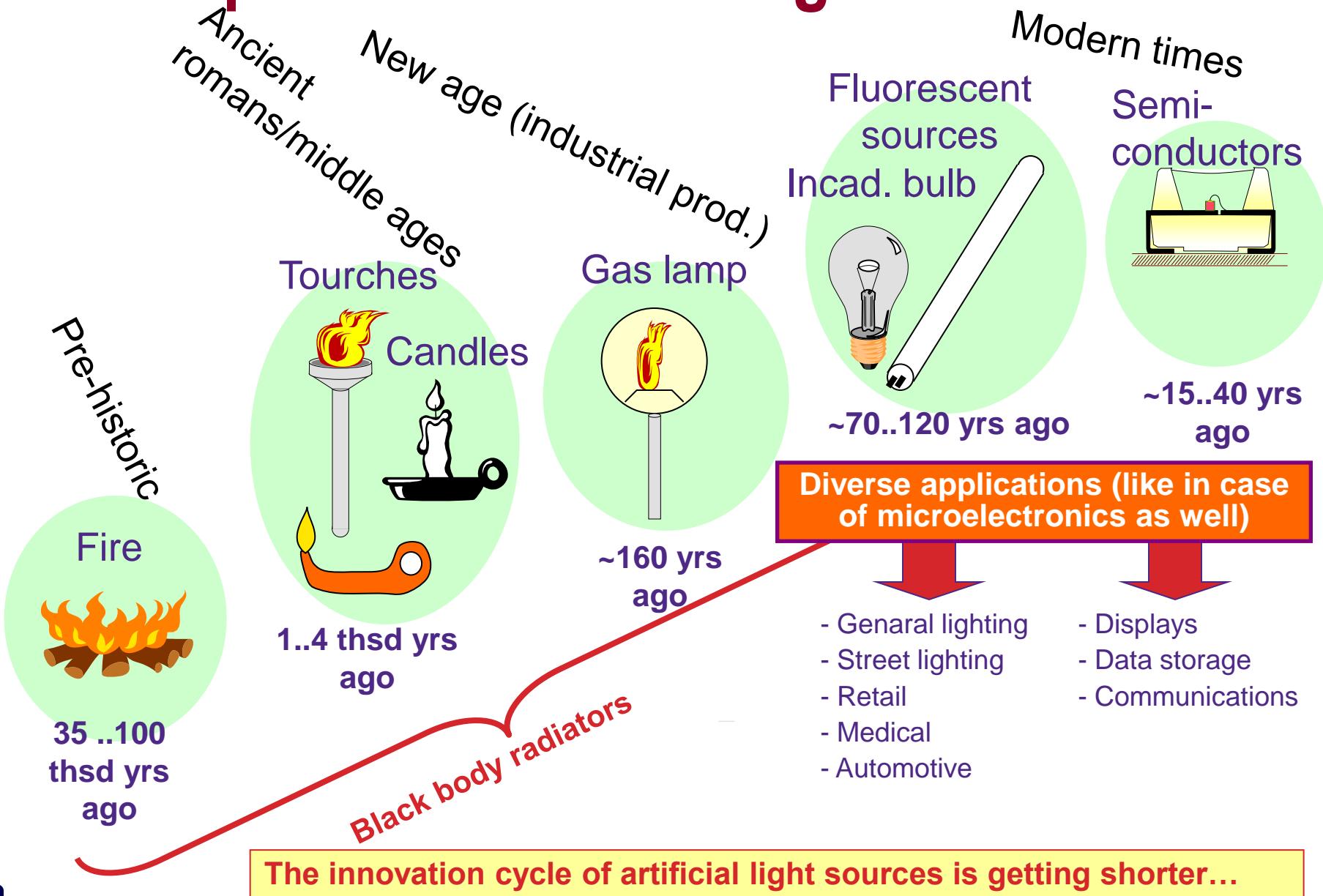
Types of light sources

- ▶ Incandescent bulb:
 - Black-body radiator
 - Continuous spectrum
- ▶ Gas discharge lamps:
 - discrete electron state transition in ionized gas or vapor
 - Discrete spectrum lines (even multiple ones)
 - Primary radiation
- ▶ Mercury lamp, CFL:
 - Like above but phosphor is also used to convert primary radiation to longer wavelengths as well
- ▶ Color LED:
 - Electron state transitions in a semiconductor crystal lattice
 - Almost monochromatic primary radiation
- ▶ White LED:
 - Electron state transitions in a semiconductor crystal lattice → primary emission
 - + wavelength conversion with phosphor

(relative) spectral power distributions



Development of artificial light sources



The first LEDs...

A Note on Carborundum.

To the Editors of Electrical World:

Strs:—During an investigation of the unsymmetrical passage of current through a contact of carborundum and other substances a curious phenomenon was noted. On applying a potential of 10 volts between two points on a crystal of carborundum, the crystal gave out a yellowish light. Only one or two specimens could be found which gave a bright glow on such a low voltage, but with 110 volts a large number could be found to glow. In some crystals only edges gave the light and others gave instead of a yellow light green, orange or blue. In all cases tested the glow appears to come from the negative pole, a bright blue-green spark appearing at the positive pole. In a single crystal, if contact is made near the center with the negative pole, and the positive pole is put in contact at any other place, only one section of the crystal will glow and that the same section wherever the positive pole is placed.

There seems to be some connection between the above effect and the e.m.f. produced by a junction of carborundum and another conductor when heated by a direct or alternating current; but the connection may be only secondary as an obvious explanation of the e.m.f. effect is the thermoelectric one. The writer would be glad of references to any published account of an investigation of this or any allied phenomena.

NEW YORK, N. Y.

H. J. ROUND.

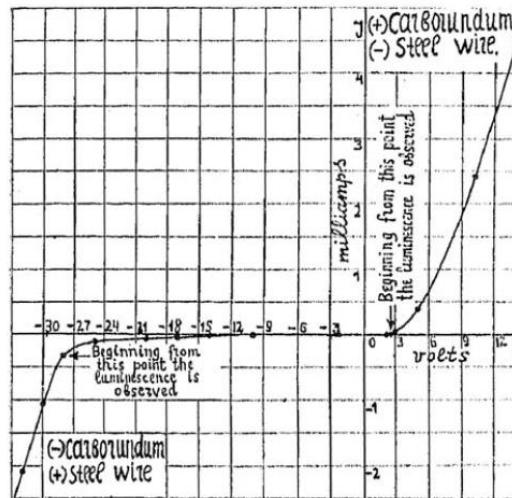


Henry Joseph Round
(1881 – 1966)

- ▶ 1907: first detection and description of electroluminescence
 - 1907: the first LED
 - Siliconcarbide (SiC) aka *carborundum*

The first LEDs...

- ▶ Oleg V. Losev (Олег В. Лосев) already had publications at the age of 20..
- ▶ first detailed description of electroluminescence of SiC
 - Major conclusion: emitted light is not from black-body radiation,
 - First LED (measurements between 1924 and 1928)
 - Light emission in both directions
- ▶ Soviet patent: 1929



Oleg Vladimirovich Losev
(1903 – 1942)



Source: PatentsFromRU.com

Предлагаемое изобретение использует общизвестное явление свечения в карборундовом детекторе и состоит в том, что в световом реле для быстропротягивающего телеграфного или телефонного приема, передачи изображений на расстояние и других целей, в качестве модулируемого электрическим током источника света, применяется свечение в точке контакта карборундового детектора, включенного непосредственно в цепь модулирующего тока.

На фиг. 1 и 2 изображает схему предпосылки светового реле и фиг. 2—схему устройства для фотографической записи сигналов с применением светового потока.

К зажимам А источника тока сигнала, подлежащих записи через потенциометр Р включается светодиодный детектор D₁ в цепь которого включена батарея В, дающая дополнительное постоянное напряжение для наложения его на напряжение тока сигнала и усиления действия реле; пучок света от эмиссии этого детектора состоит из направляющей установки выбора детектора D. Оптическая система L предназначена направлять световой поток, излучаемый карборундовым детектором, на движущуюся фотографическую пла-

тинку F, на которой производится запись изменений этого потока. Детектор D, оптическая система L и пластина F заключены в светонепроницаемую камеру. Примерное включение светового реле показано на чертеже 2, где Е—приемник-усилитель высокой частоты, Т—автотрансформатор высокой частоты, а оставшаяся часть схемы вполне аналогична только что описанной.

Предмет патента.

1. Световое реле для быстропротягивающего телеграфного или телефонного приема, передачи изображений на расстояние и для других целей, характеризующееся применением, в качестве модулируемого электрическим током источника света, свечения в точке контакта карборундового детектора общизвестного устройства, каковой детектор включен непосредственно в цепь модулирующего тока.

2. Выразименение охарактеризованного в патенте светового реле, отличающееся тем, что вспомогательно с указанным детектором D включен источник дополнительного напряжения постоянного тока В (фиг. 1 и 2) с целью усиления действия реле.

History of commercial LEDs

- 1962: TI – 1st commercial GaAs IR LED,
- 1962: GE – 1st commercial red LED (N. Holonyak)
- 1972: yellow / greenish LED-ek
- 1978: 1st high intensity LED
- 1989: GaN homo-junction LED
- 1993: Efficient blue LED
- 1997: White LED (blue+phosphor)
- 2001: White LED (UV LED + phosphor)
- Today: various high power LED-ek
 - 1 .. 10 .. 100 W – HPS lamps also replaced



Akasaki, Amano and Nakamura professors
Nobel prize in physics 2014

First LEDs were used as indicators only, modern high efficiency LEDs completely changed the lighting industry by now...

Characteristic features of LEDs

► Electrical parameters

- **Forward voltage:** 2.5 V .. 4 V, depending on the color;
 - Larger V_F : multiple PN junctions connected in series
 - CoB LEDs: cca. 50 V, AC LEDs: 120 V / 230 V
- **Forward current:**
 - Low power classical LEDs: ~10 mA
 - High power LEDs: 300 mA ... 800 mA ... 1500 mA
- Reverse direction: small breakdown voltage \Rightarrow protection diode (is also an LED – red)

► Characteristics of the package:

- **Thermal resistance:** 300 K/W .. 10 K/W .. 1 K/W .. 0.1 K/W
- Package style: exposed cooling surface/MCPCB, type of optics

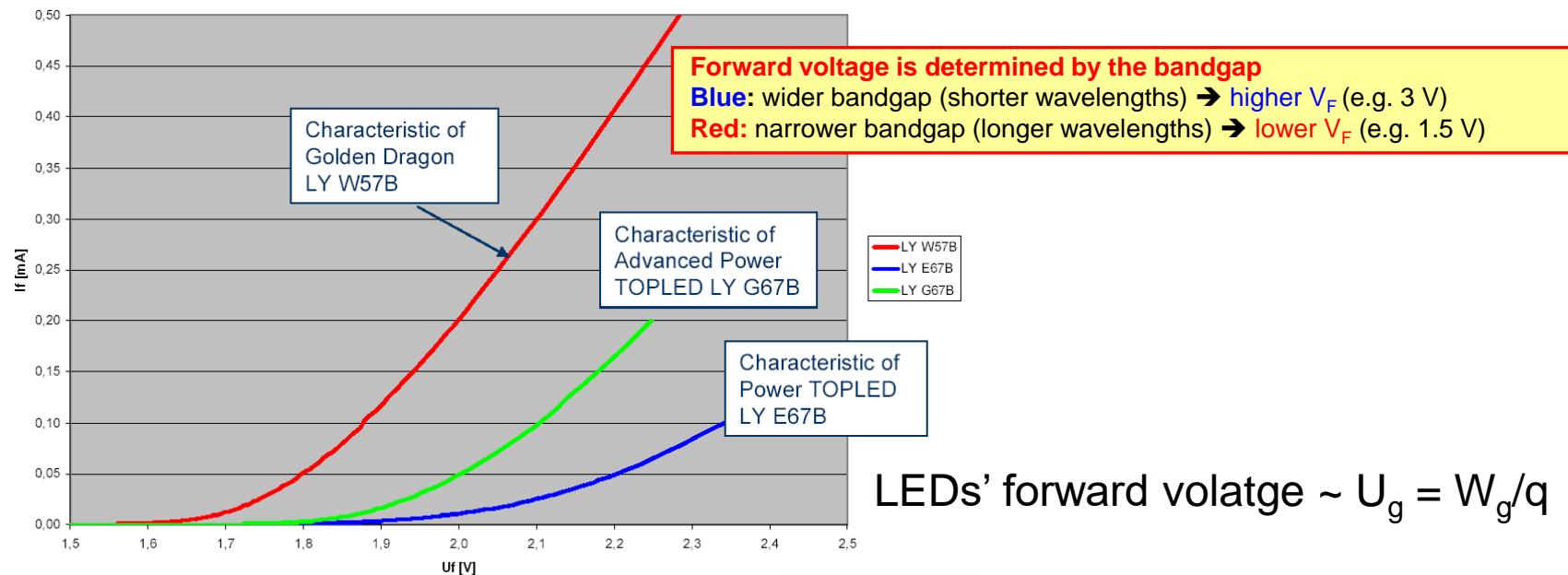
► Optical parameters

- **Luminous flux [lm]**, luminous efficacy [lm/W]
- Emitted optical power aka radiant flux [W]
- spectrum + dominant wavelength (colour LEDs) or correlated colour temperature + x,y coordinates (fwhite)
- Radiation pattern

► Efficiencies – many of them



Forward voltage and colour of light



Example

$$U_g = (c \cdot h) / (\lambda \cdot q)$$

Peak wavelength from the spectrum of a blue LED: **447nm**

$$c \cdot h = 3e8 \cdot 6.625e-34 = 1.988e-25$$

$$\lambda \cdot q = 4.47e-7 \cdot 1.602e-19 = 7.16e-26$$

$$U_g = 1.988 / 7.16 \cdot 10 = 2.777 \text{ V}$$

Different efficiencies of LEDs

► **Quantum efficiency** (kvantumhatásfok) [%]

- Number of emitted photons per number of injected electrons
- For PC white LEDs: **conversion efficiency of phosphor**

► **Extraction efficiency** [%]

- Number of photons leaving the LED to free space per photons generated in the junction

► **Power efficiency / wall-plug efficiency – WPE / η_e [%]** (energy conversion efficiency)

- Emitted optical power (*total radiant flux*) per supplied electric power ($P_{opt} / P_{el} = \Phi_e / P_{el}$)

► **Efficacy / luminous efficiency η_V [lm/W]**

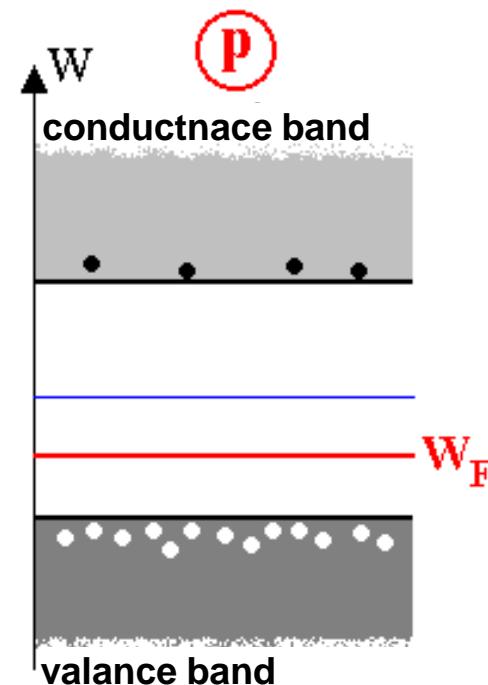
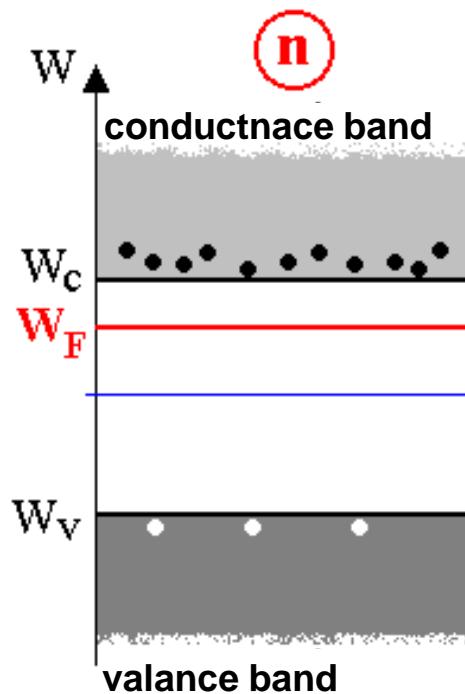
- Emitted total luminous flux per supplied electrical power

RECAP SOME PN-JUNCTION BASICS...



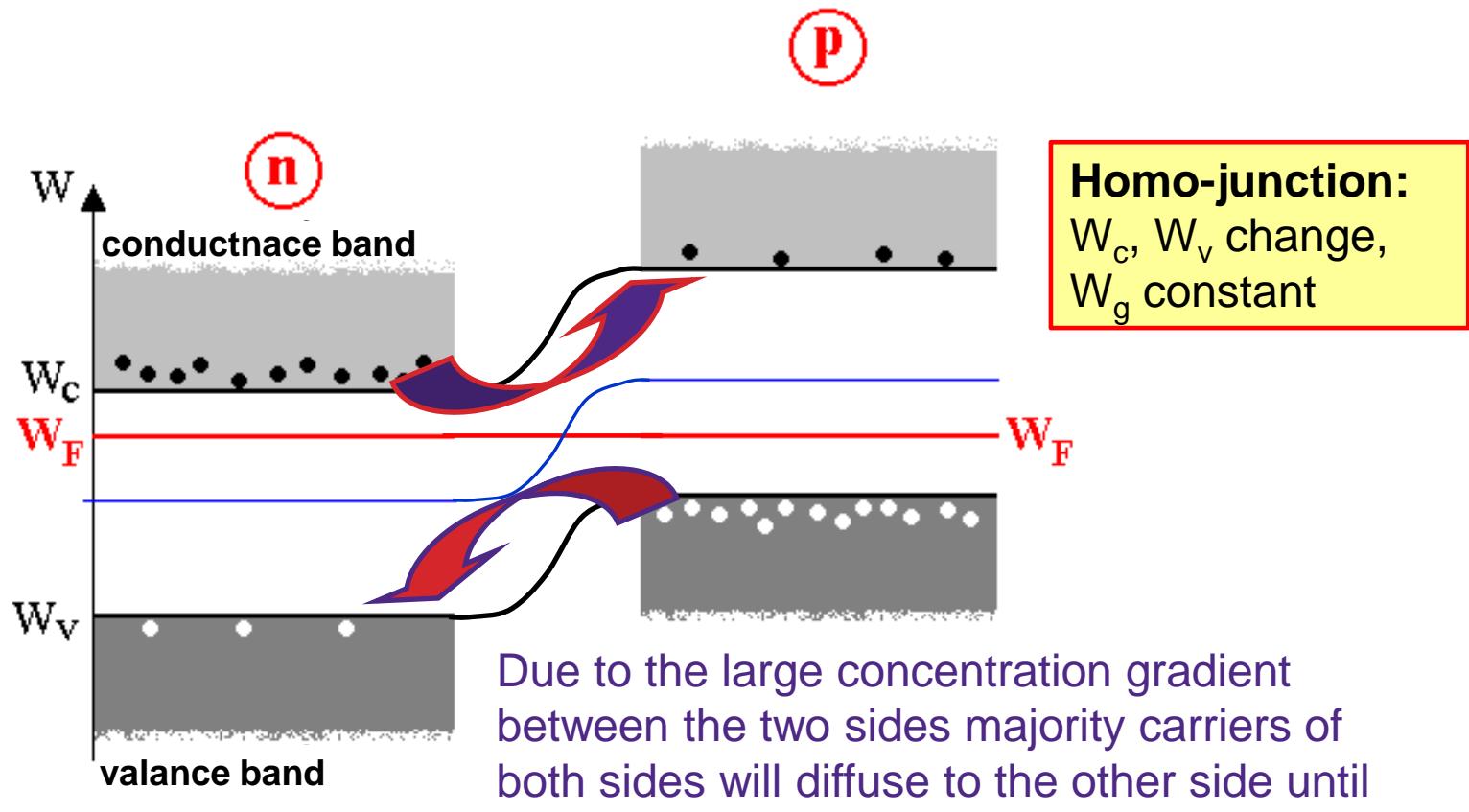
N and P type layers: separated

- Fermi levels shifted with respect to the intrinsic Fermi level according to doping:



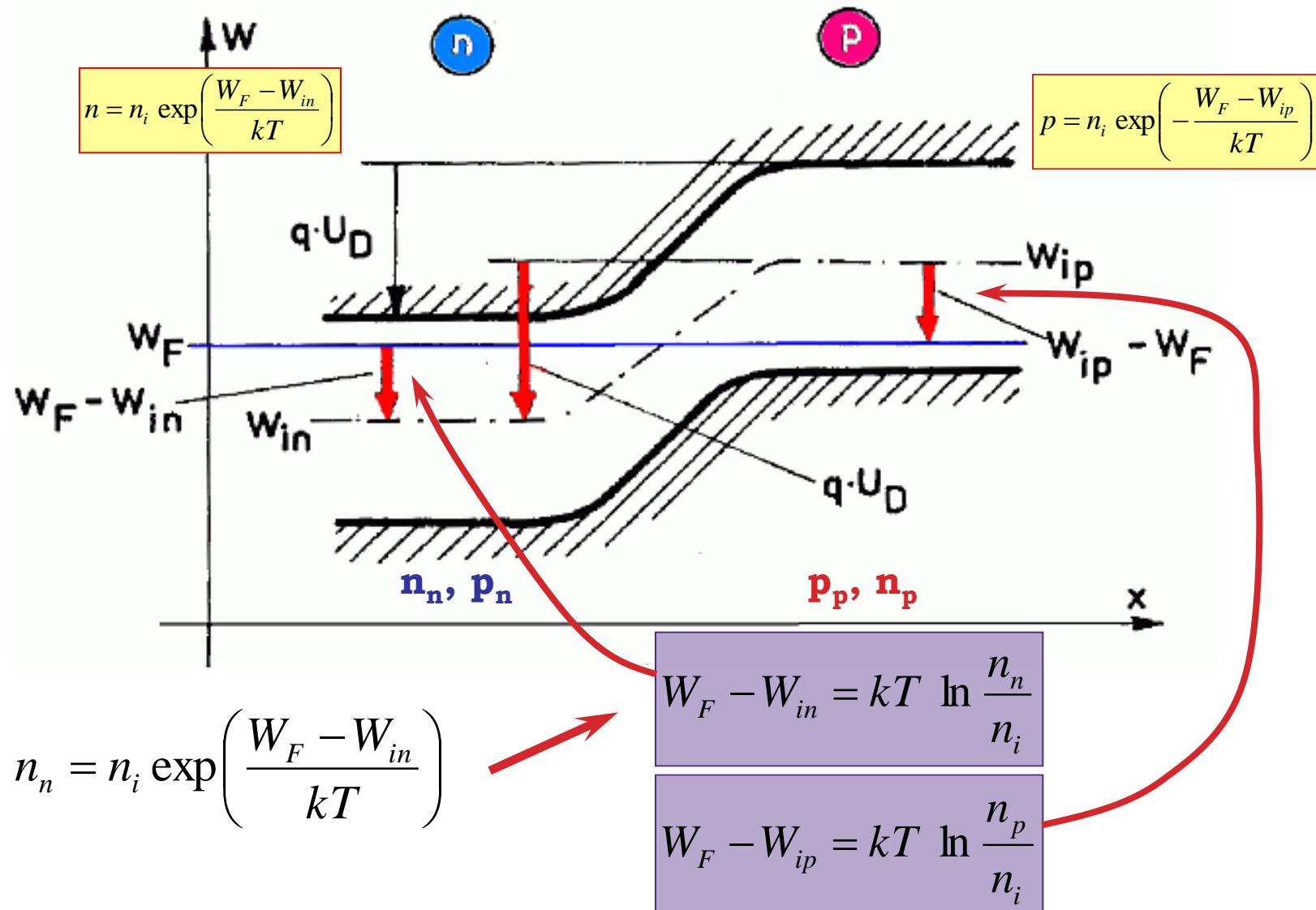
PN junction

- ▶ A potential step develops between the p and n sides. This will be so high that the Fermi-levels of both sides will be equal:

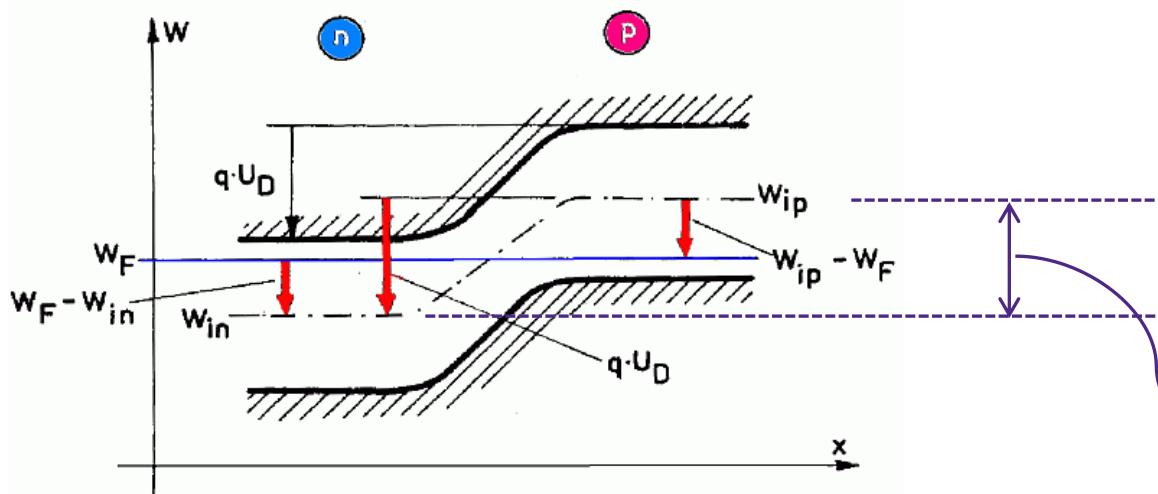


- ▶ Carrier gradient between the two sides of the junction → diffusion current → depletion layer / spacecharge

Diffusion potential



Diffusion potential



$$U_D = \frac{W_{in} - W_{ip}}{-q} = \frac{kT}{q} \ln \frac{n_n}{n_p} = \frac{kT}{q} \ln \frac{n_n p_p}{n_i^2}$$

$$U_D = U_T \ln \frac{N_d N_a}{n_i^2}$$

„built-in” voltage

$$W_F - W_{in} = kT \ln \frac{n_n}{n_i}$$

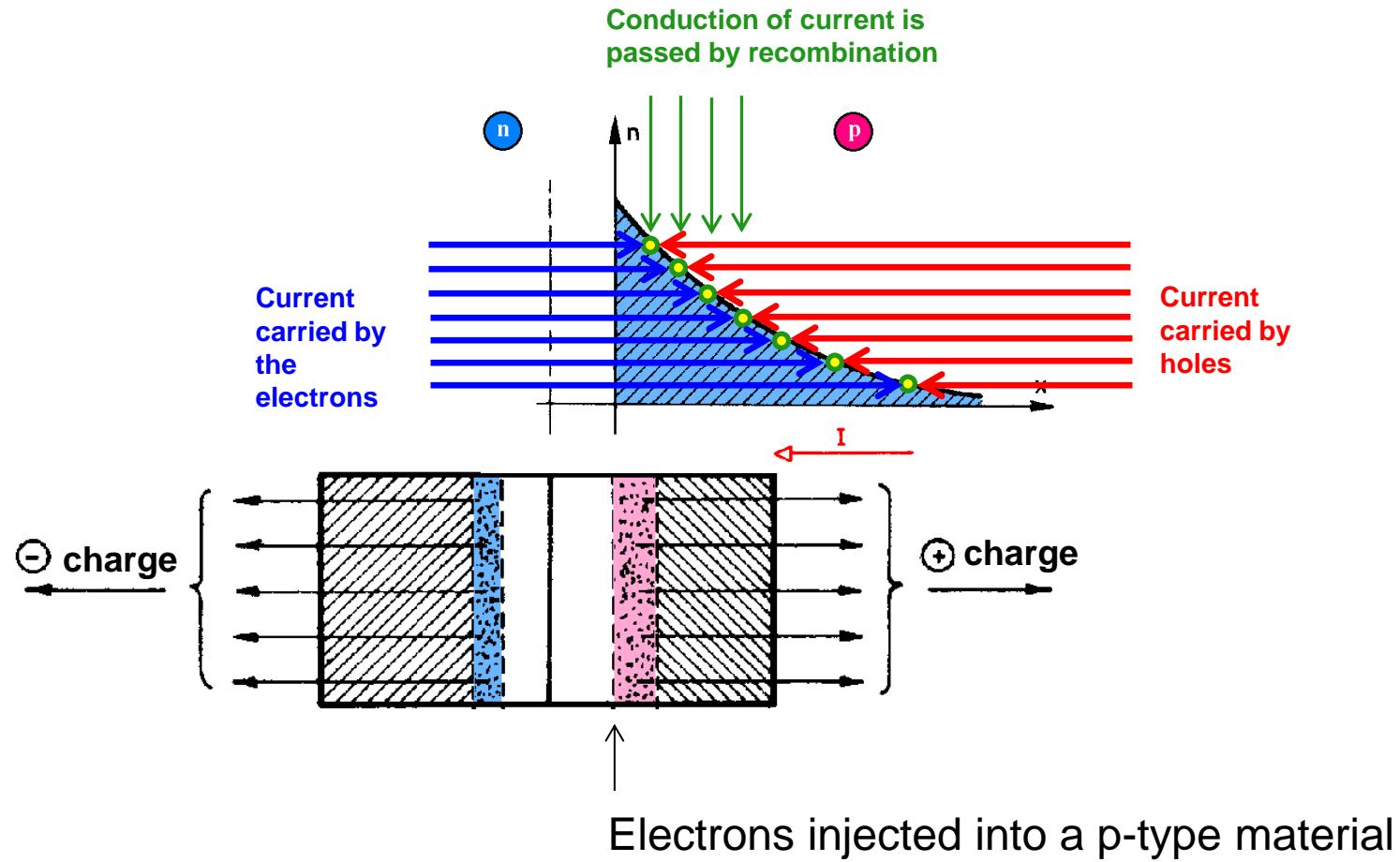
$$W_F - W_{ip} = kT \ln \frac{n_p}{n_i}$$

$$W_{ip} - W_{in} = kT \ln \frac{n_n}{n_p}$$

$$n_p = n_i^2 / p_p$$

mass effect law

Recombination at the PN junction



The type of recombination matters (direct: light, indirect: heat)

TODAY'S LED STRUCTURES: DOUBLE HETEROJUNCTION MQW

Double heterojunction

► Nowadays it is typical for all LEDs

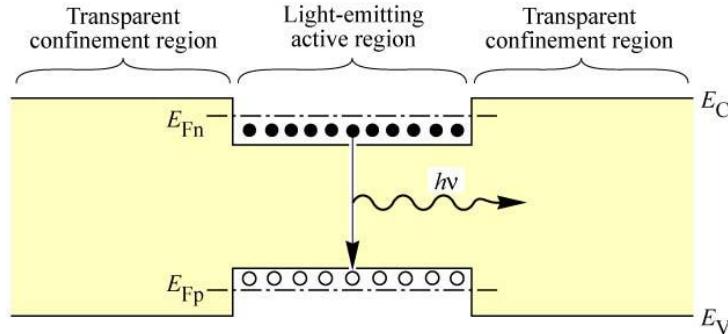


Fig. 9.2. Double hetero-structure with optically transparent confinement regions. Re-absorption in the active region is unlikely due to the high carrier concentration in the active region and the resulting Burstein–Moss shift of the absorption edge.

E. F. Schubert
Light-Emitting Diodes (Cambridge Univ. Press)
www.LightEmittingDiodes.org

► Explanation of the benefit:

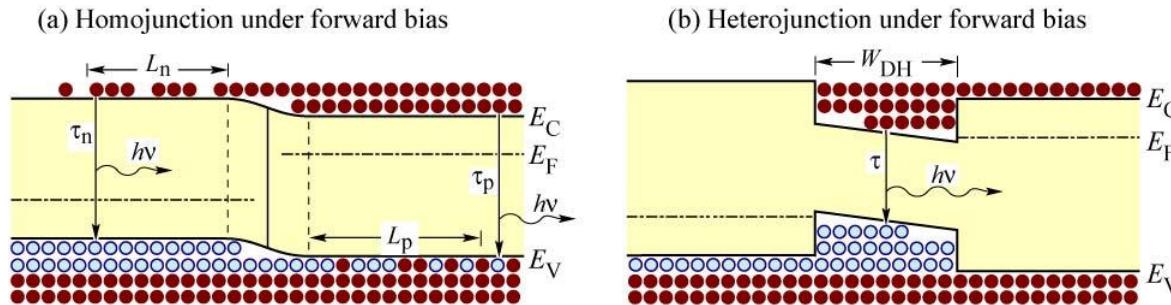


Fig. 7.2. Free carrier distribution in (a) a homojunction and (b) a heterojunction under forward bias conditions. In homojunctions, carriers are distributed over the diffusion length. In heterojunctions, carriers are confined to the well region.

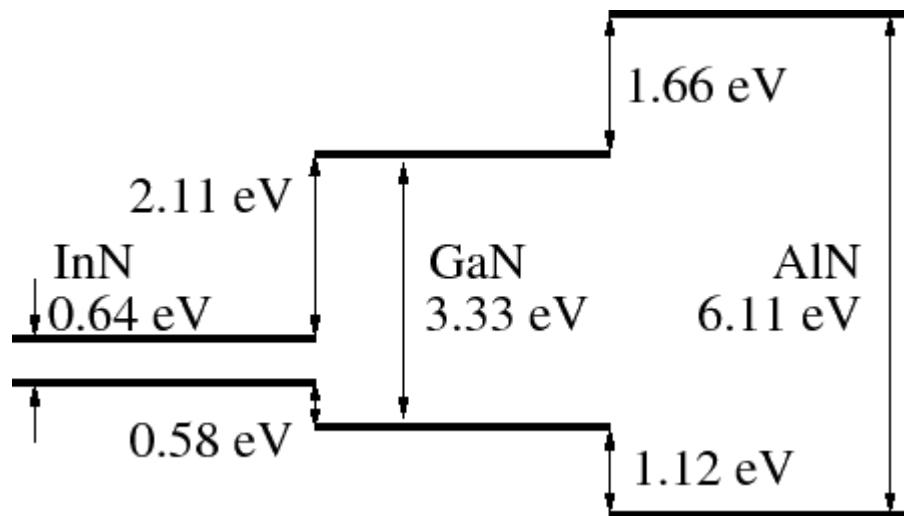
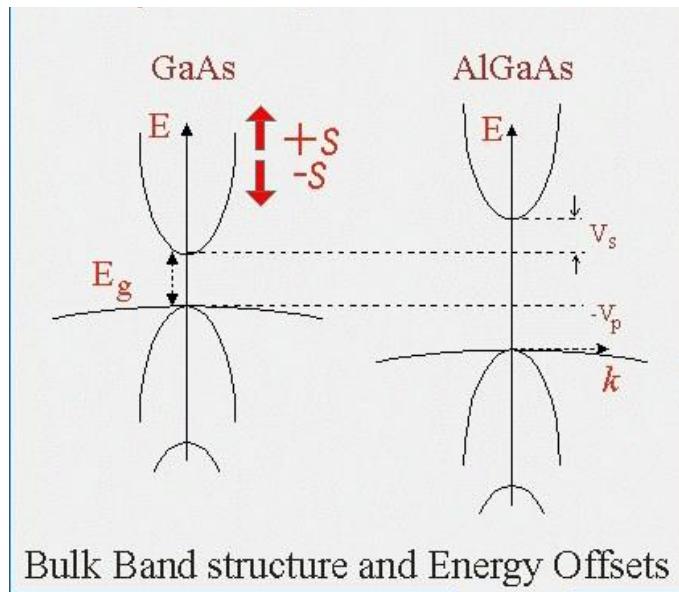
E. F. Schubert
Light-Emitting Diodes (Cambridge Univ. Press)
www.LightEmittingDiodes.org

<http://www.ecse.rpi.edu/~schubert/Light-Emitting-Diodes-dot-org/chap07/chap07.htm>

The heterojunction

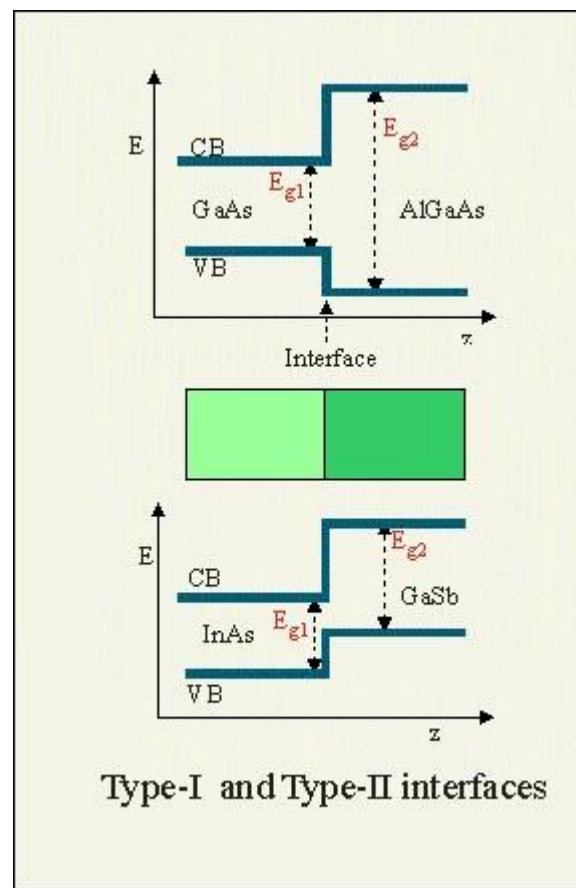
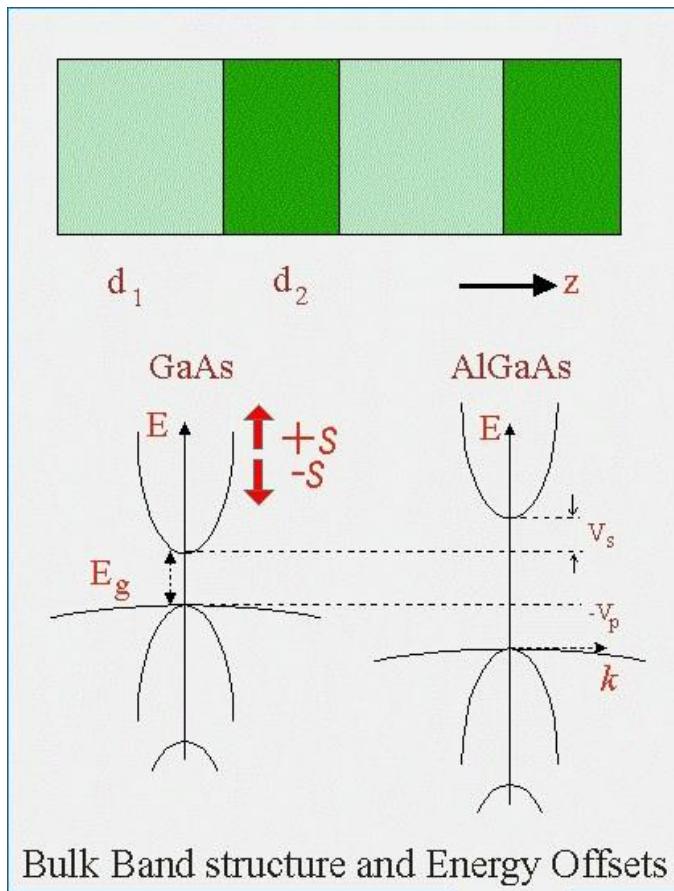
A heterojunction is formed when two materials with different bandgap mate...

Compound semiconductors: the valence and conduction bands are shifted, sometimes asymmetrically...

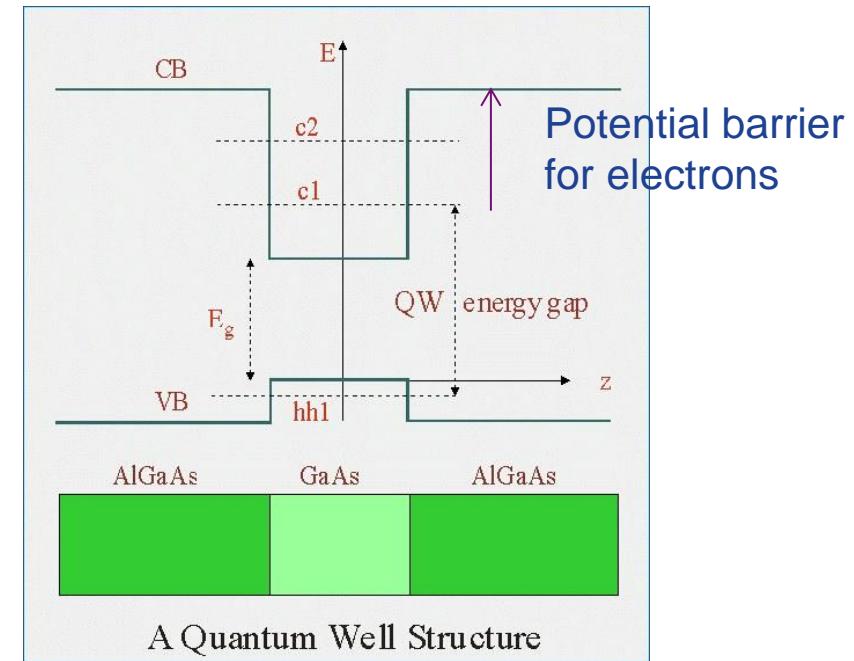
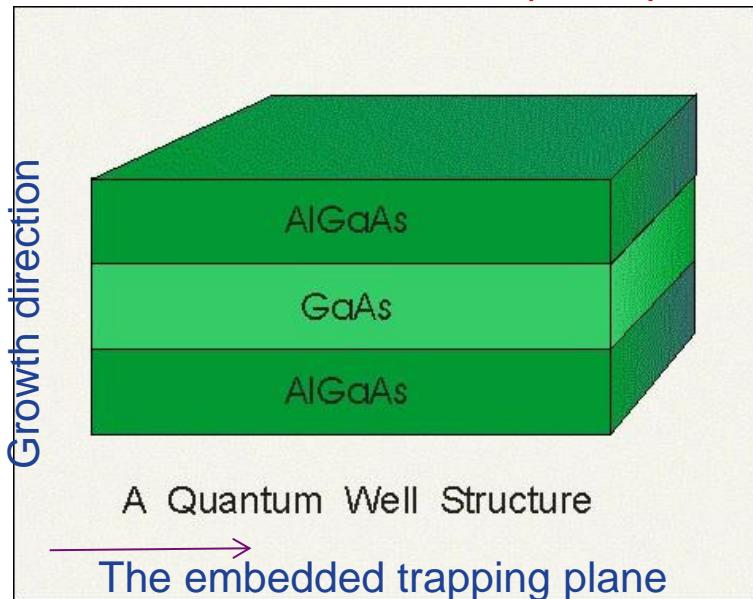


$$\mathcal{E}_{\text{off}}^{\text{ABC}} = \frac{\mathcal{E}_{\text{off}}^{\text{AC}}(\mathcal{E}_g^{\text{ABC}} - \mathcal{E}_g^{\text{BC}}) - \mathcal{E}_{\text{off}}^{\text{BC}}(\mathcal{E}_g^{\text{ABC}} - \mathcal{E}_g^{\text{AC}})}{\mathcal{E}_g^{\text{AC}} - \mathcal{E}_g^{\text{BC}}},$$

Heterojunction: A change in the semiconductor material when not only the positions of the valence and conductance bands change (homojunction) but the bandgap also changes:



The Quantum Well (QW) structure



- The charge carriers are trapped in the GaAs layer
 - They cannot move in the direction of layer growth
 - They are free to move in the trapping plane
- **Charge carriers confined into a small region → increased probability of recombination**
- The gap of the QW is wider than the gap of the bulk material.
- The QW gap energy can be controlled by the width of the QW.

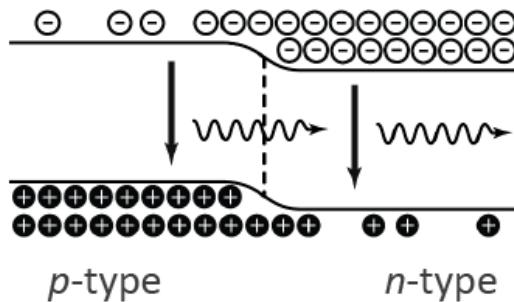
GaN QW: p-GaN – InGaN – n-GaN

- **MQW: Multiple Quantum Well**

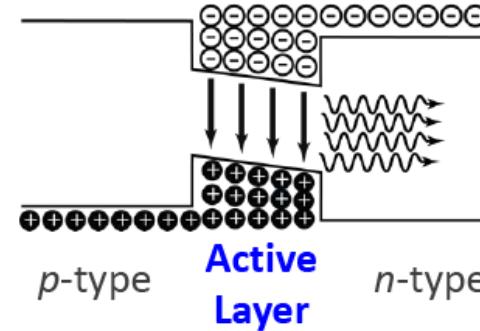
Double heterojunction

Energy Band Diagrams

Homojunction LED



Double Heterostructure LED



Internal Quantum Efficiency

$$\eta_{IQE} = \frac{\text{Light generated}}{\text{Electrons injected}} = \frac{R_{\text{radiative}}}{R_{\text{radiative}} + R_{\text{non-radiative}}} = \frac{Bn^2}{An + Bn^2 + Cn^3}$$

Shockley-Read-Hall (SRH) *Spontaneous Emission* *Auger*

Double heterostructures **increase carrier concentrations (n)** in the active layer and **enhance radiative recombination** rates (more light generated).

Source: http://www.nobelprize.org/nobel_prizes/physics/laureates/2014/nakamura-lecture-slides.pdf

MODELLING LED OPERATION

Ideal diode characteristic

$$J_n \Big|_{x=0} = \frac{qD_n n_p}{L_n} (\exp(U/U_T) - 1)$$

$$J_p = \frac{qD_p p_n}{L_p} (\exp(U/U_T) - 1)$$

**I_0 is proportional with
the minority carrier
concentration!**

$$I = A(J_n + J_p)$$

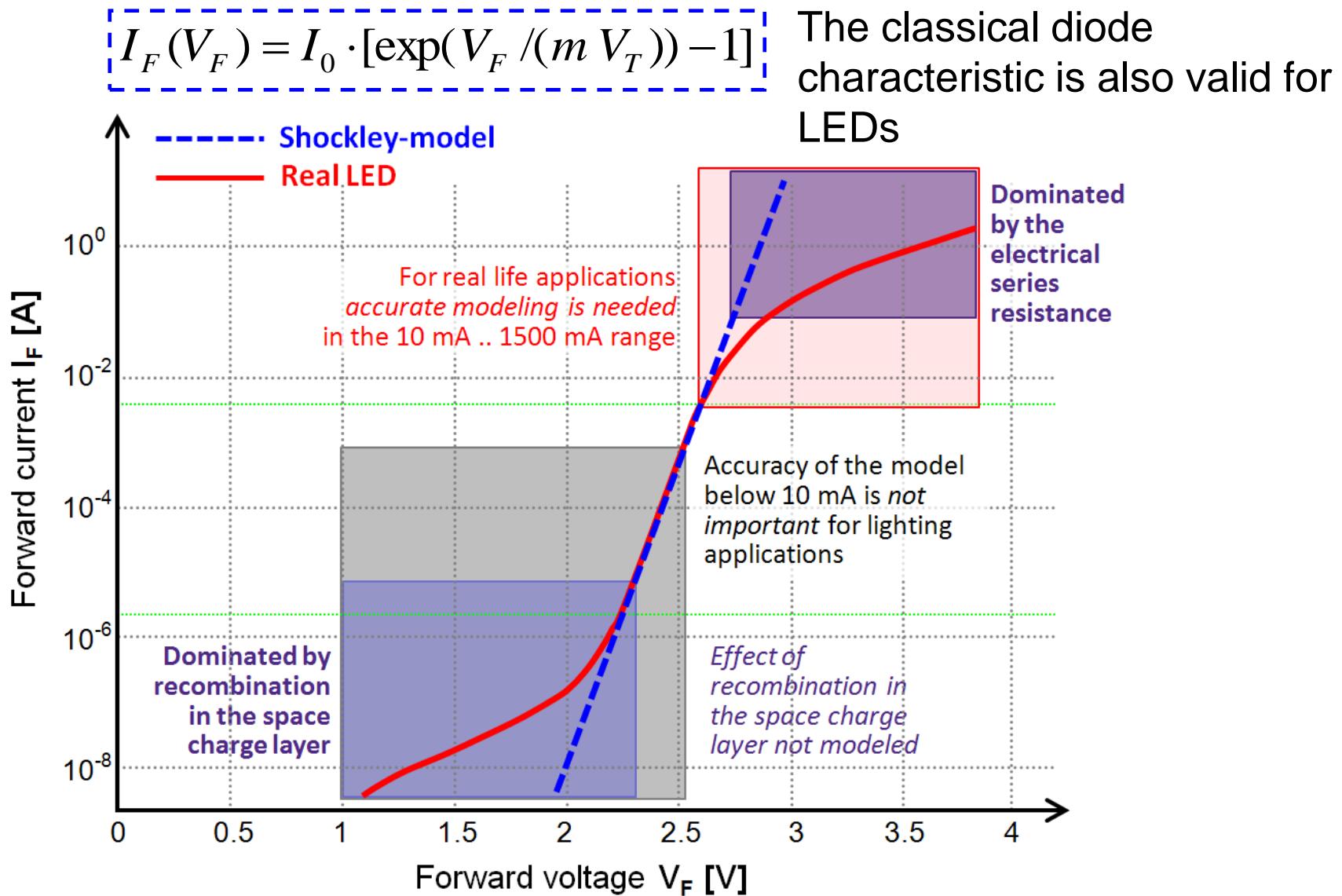
$$I = Aq \left(\frac{D_n n_p}{L_n} + \frac{D_p p_n}{L_p} \right) (\exp(U/U_T) - 1)$$

$$I = I_0 (\exp(U/U_T) - 1)$$

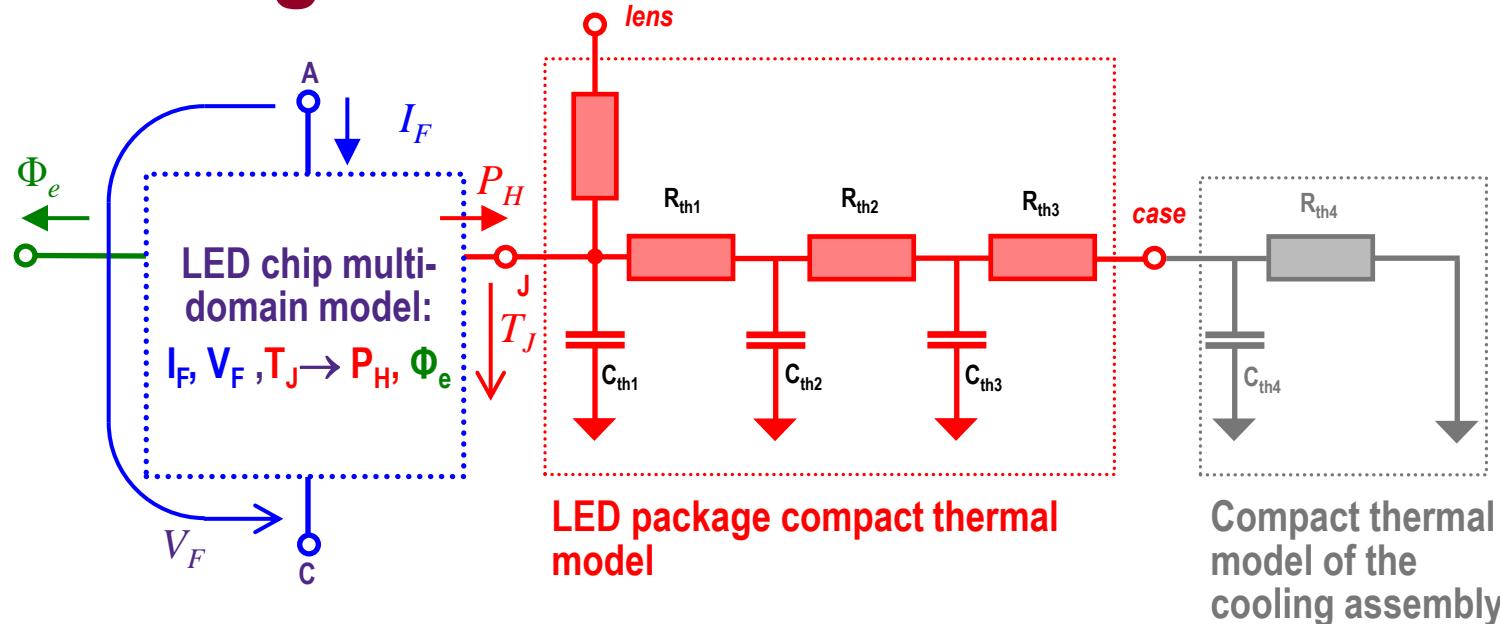


$$U = U_T \ln(I/I_0 + 1)$$

A real LED I-V characteristic



Modelling LEDs and LED luminaires



- ▶ **Thermal model:** represents the junction to ambient heat-flow path by means of an RC network model
 - Using within a CFD simulator: only the **package** model
 - Using an electro-thermal circuit simulator: the **package** model and the model of the cooling assembly
- ▶ **LED chip modell:** electro-**thermal** and also calculates the **emitted optical power** → **multi-domain**
 - Further properties: Φ_V , spectrum → **hot lumens**
 - **Black box:** no deep physics of the device → **this is the compact model**

Chip level model for an ideal LED

► I_F – 2 components:

- I_{dis} – heat generation
(non-radiative recombination)
- I_{rad} – light emission
(radiative recombination)

$$I_{rad}(V_F) = \Phi_e / V_F$$

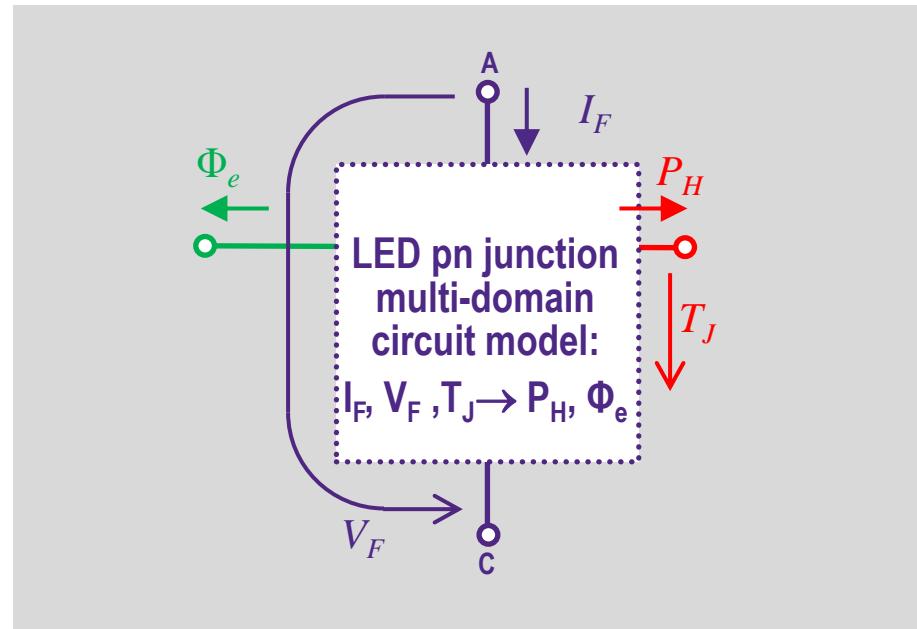
► Measured $I_F = P_H + \Phi_e$

$$I_F = \frac{P_H}{V_F} + \frac{\Phi_e}{V_F}$$

$$I_F(V_F) = I_{dis}(V_F) + I_{rad}(V_F) \quad \text{ahol}$$

$$I_{dis}(V_F) = I_F - \Phi_e / V_F$$

$$I_{rad}(V_F) = \Phi_e / V_F$$



$$I_{rad}(V_F) = I_{0_rad} \cdot [\exp(V_F / (n_{rad} V_T)) - 1]$$

$$I_{dis}(V_F) = I_{0_dis} \cdot [\exp(V_F / (n_{dis} V_T)) - 1]$$

Chip level model for an ideal LED

► I_F – 2 components:

- I_{dis} – heat generation
(non-radiative recombination)
- I_{rad} – light emission
(radiative recombination)

$$I_{rad}(V_F) = \Phi_e / V_F$$

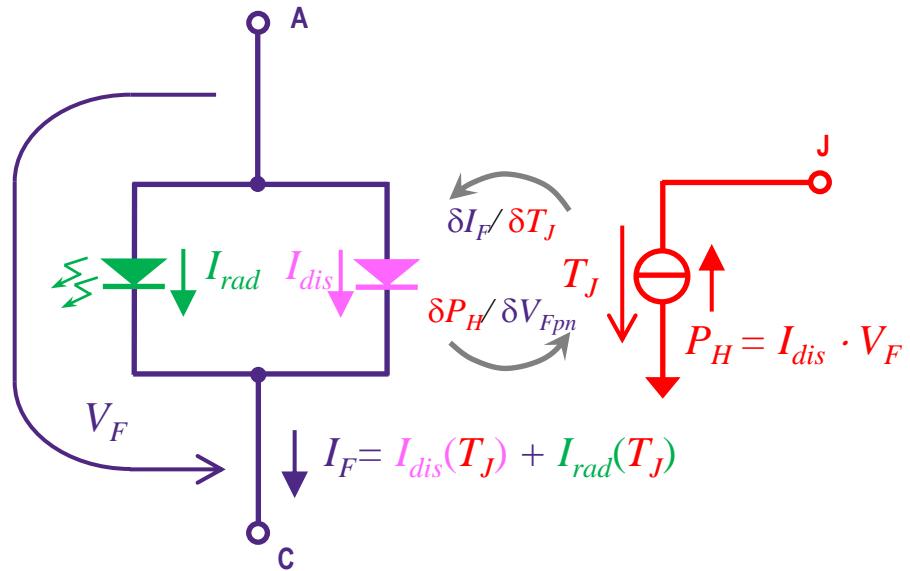
► Measured $I_F = P_H + \Phi_e$

$$I_F = \frac{P_H}{V_F} + \frac{\Phi_e}{V_F}$$

$$I_F(V_F) = I_{dis}(V_F) + I_{rad}(V_F) \quad \text{ahol}$$

$$I_{dis}(V_F) = I_F - \Phi_e / V_F$$

$$I_{rad}(V_F) = \Phi_e / V_F$$



$$I_{rad}(V_F) = I_{0_rad} \cdot [\exp(V_F / (n_{rad} V_T)) - 1]$$

$$I_{dis}(V_F) = I_{0_dis} \cdot [\exp(V_F / (n_{dis} V_T)) - 1]$$

Light emission and dissipation

► The total supplied electric power: $P_{el} = V_F \cdot I_F$

$$I_F(V_F) = I_{dis} \cdot [\exp(V_F / nV_T) - 1] + I_{rad} \cdot [\exp(V_F / mV_T) - 1]$$

$$P_{el} = I_{dis} \cdot [\exp(V_F / nV_T) - 1] \cdot V_F + I_{rad} \cdot [\exp(V_F / mV_T) - 1] \cdot V_F$$

$$P_{el} = I_{dis} \cdot [\exp(V_F / mV_T) - 1] \cdot V_F + I_{rad} \cdot [\exp(V_F / nV_T) - 1] \cdot V_F$$

$P_{diss} = P_{el} - \Phi_e$

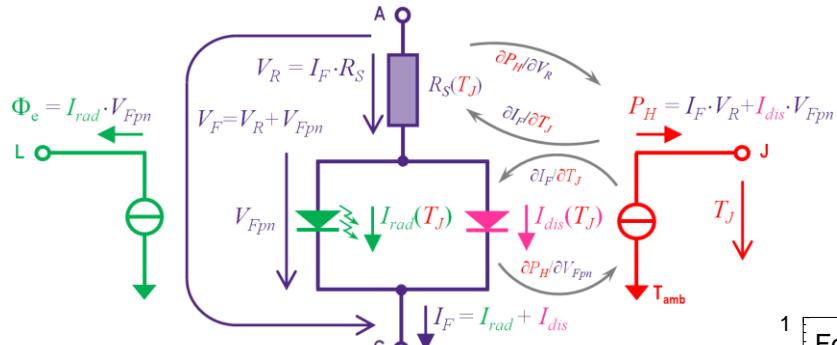
This is the **radiant flux** Φ_e [mW]
or optical power P_{opt} [mW]

heat

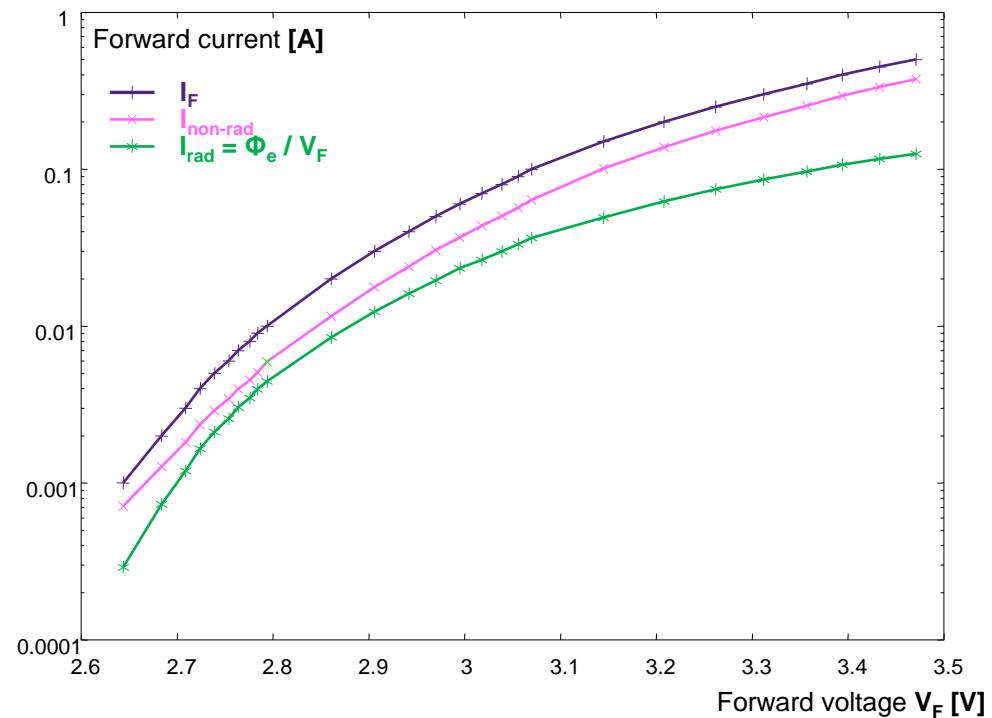
light

Ideal LED + series resistance

- Modelling the series resistance is important since LEDs for lighting are used at high current operating point

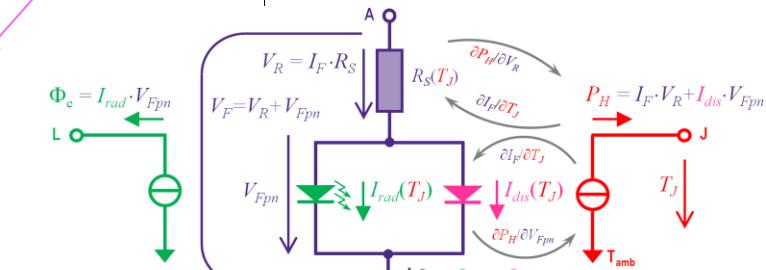
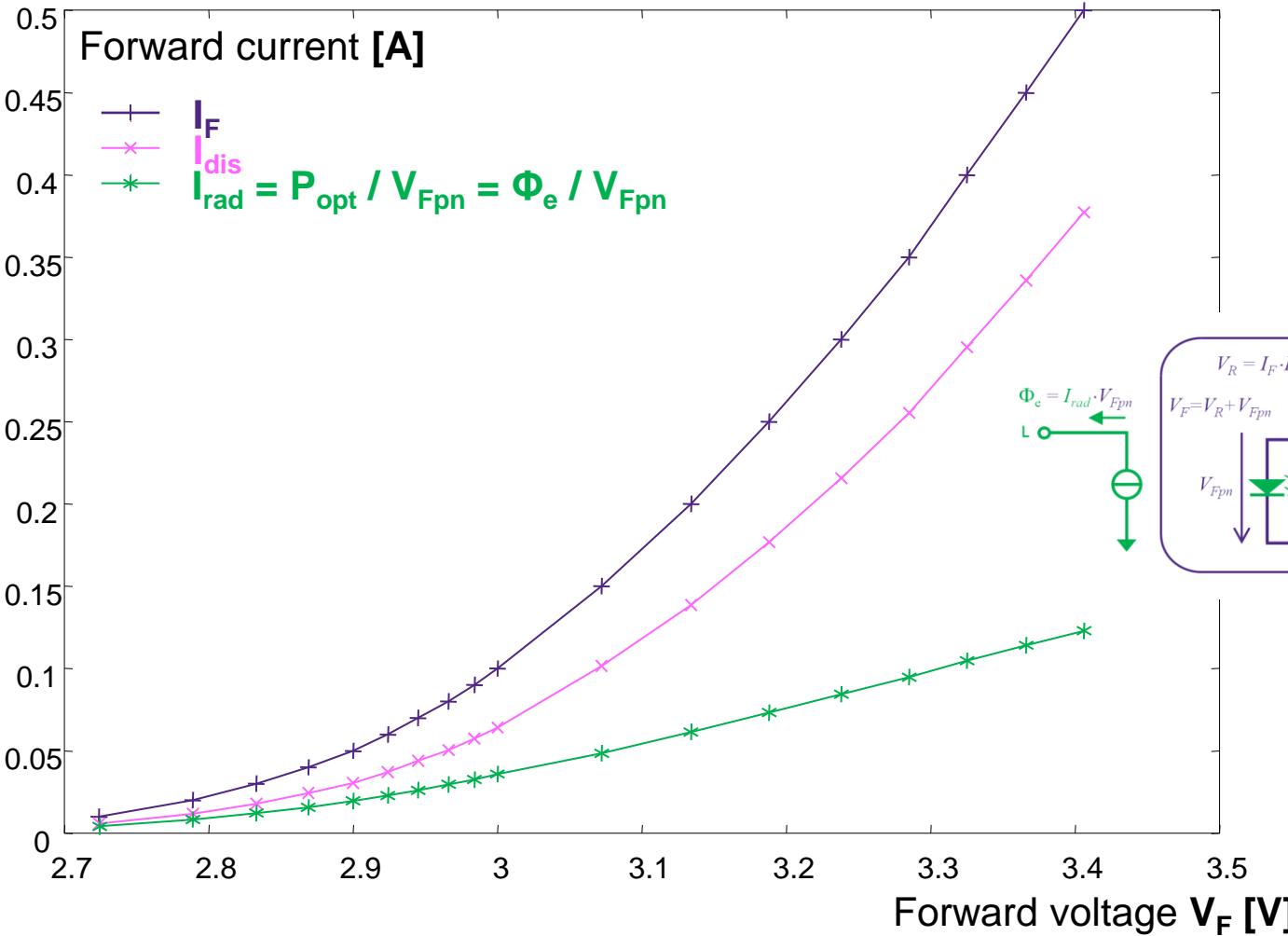


Measured LED forward current components at $T_J = 30^\circ\text{C}$



Some measurement results:

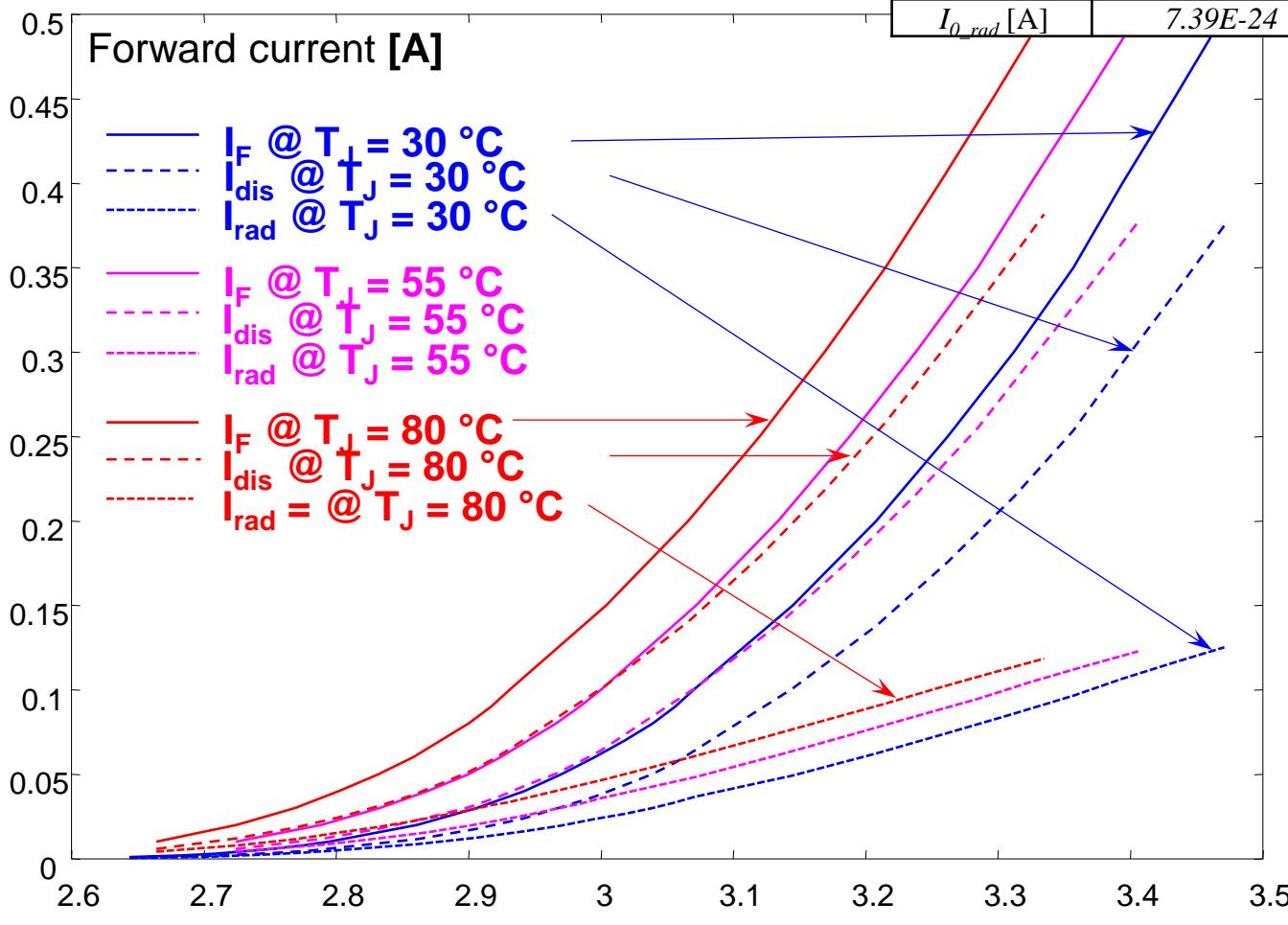
Measured LED forward current components at $T_J = 55^\circ\text{C}$



Some measurement results:

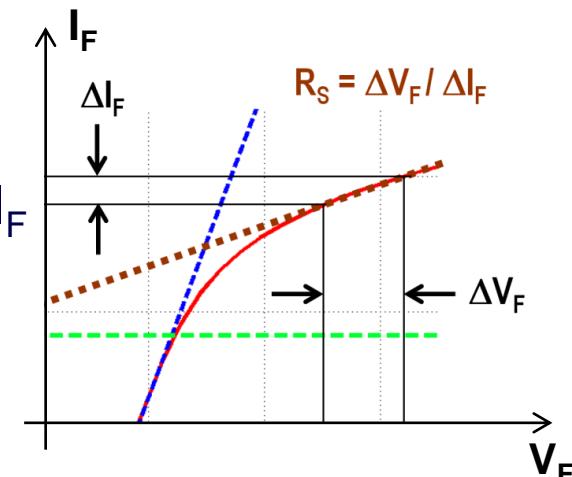
Measured LED forward current components at $T_J = 30^\circ\text{C}, 55^\circ\text{C}, 80^\circ\text{C}$

	$T_J = 30^\circ\text{C}$	$T_J = 55^\circ\text{C}$	$T_J = 80^\circ\text{C}$
$R_s [\Omega]$	0.77	0.81	0.85
$m [-]$	2.63	2.46	2.29
$I_{0_dis} [\text{A}]$	$3.94E-20$	$1.70E-19$	$3.01E-19$
$n [-]$	2.17	1.94	1.71
$I_{0_rad} [\text{A}]$	$7.39E-24$	$5.85E-24$	$4.31E-24$



Parameter extraction (overview):

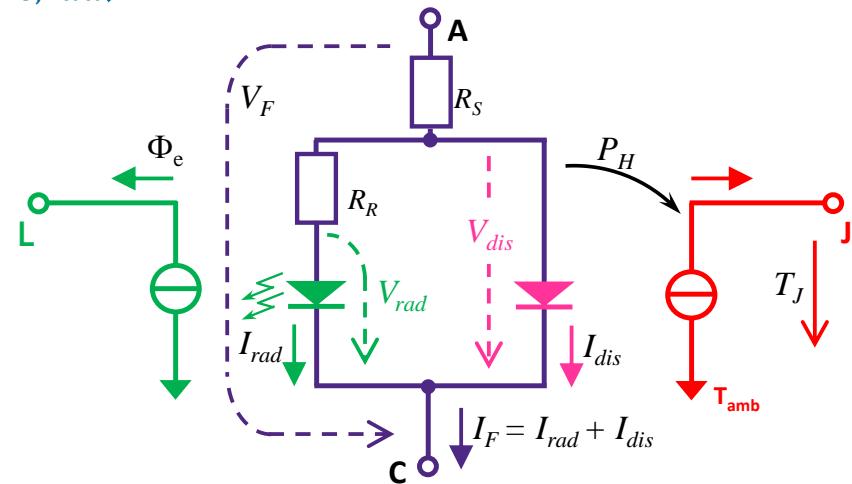
1. For all set T_J
 - a) Find R_S (see diagram)
 - b) Internal junction voltage: $V_{Fpn} = V_F - R_S(T_J) \cdot I_F$
 $I_F(V_{Fpn})$ is the characteristic of the inner junction (free of effect of R_s)
 - c) Find components of I_F :
 - $I_{rad}(V_{Fpn}) = P_{opt}(I_F)/V_{Fpn}$
 - $I_{dis}(V_{Fpn}) = I_F(V_{Fpn}) - I_{rad}(V_{Fpn})$
 - d) Curve fitting
 - $I_{rad}(V_{Fpn})$ points to the $I_{rad}(V_{Fpn}) = I_{0_rad} \cdot [\exp(V_{Fpn}/(n_{rad} V_T)) - 1]$ equation
 - $I_{dis}(V_{Fpn})$ points to the $I_{dis}(V_{Fpn}) = I_{0_dis} \cdot [\exp(V_{Fpn}/(n_{dis} V_T)) - 1]$ equation
2. Continue at step 1 for the next T_J
3. Fit the data series of the parameters $I_{0_rad}(T_J)$, $n_{rad}(T_J)$, $I_{0_dis}(T_J)$, $n_{dis}(T_J)$ és $R_S(T_J)$ approximate formulae (e.g. linear temperature dependence of R_S)



Improved LED model

- The previous model (2014) overestimated light emission at high currents, disregarding some optical losses...
- R_R is added to account for these losses

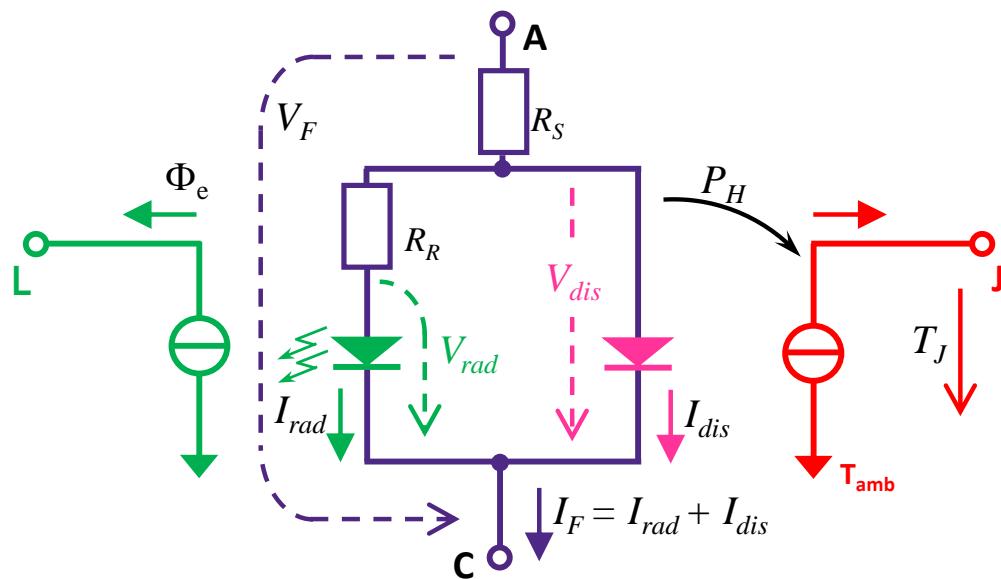
$$V_{dis} = I_{rad} \cdot R_R + m_{rad} \cdot U_T \cdot \ln \left(\frac{I_{rad}}{I_{0,rad}} \right)$$



- Developed and implemented in the Delphi4LED project of the EU (Visual Basic macro for Excel, generic SPICE circuit macro for LT-Spice)

Improved LED chip level multi-domain model

The two diode branches represent the dissipative and radiative recombination processes



Implementations:

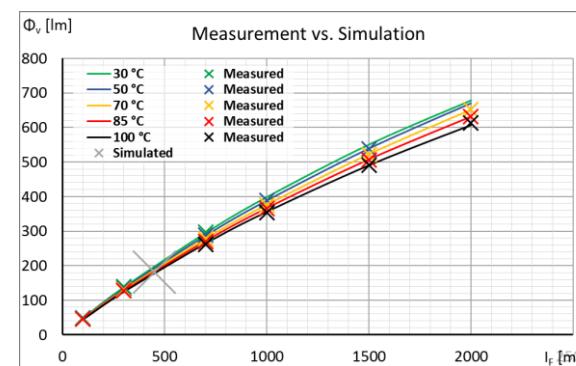
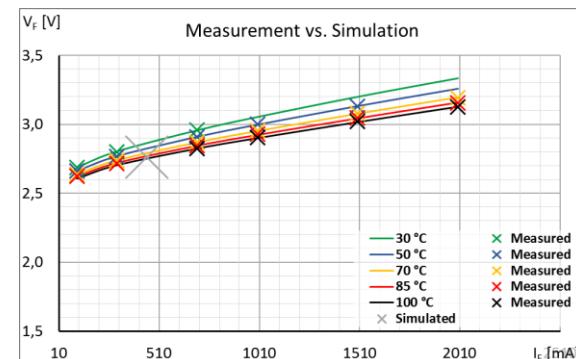
- VB macro
- LT Spice (electrical only Spice)

In about 2 decades of the total forward current I-V-L characteristics fits the measured data within $\sim 1\text{-}2\%$

Parameter identification

- New model topology, reformulated equations and set of parameters assure ~1% fit to measured characteristics of individual LED samples

A	B	C	D	E	F	G	H
15			4				
Sample:	XPG3_01	XPG3_02	XPG3_03	XPG3_04	XPG3_05		
Max Vf error:	0%	1%	0%	0%	0%		
Max Fi_e error:	1%	1%	1%	1%	1%		
Max Fi_v error:	1%	1%	1%	1%	1%		
UT	-	0,0296	0,0296	0,0296	0,0296	0,0296	
I0	=	7,6395E-24	6,9812E-24	8,1736E-24	7,2375E-24	7,6335E-24	
m	=	1,7354	1,7349	1,7359	1,7353	1,7358	
R	=	0,1929	0,2141	0,197	0,2138	0,1973	
I0_rad	=	4,0317E-23	3,4889E-23	3,4027E-23	3,3826E-23	3,1585E-23	
m_rad	=	1,8150	1,8131	1,8075	1,8107	1,8089	
R_rad	=	0,0190	0,021001	0,020001	0,021001	0,019001	
a_el	=	-8,079E-06	-2,501E-06	-6,348E-06	1,973E-07	-3,155E-06	
b_el	=	2,153E-05	1,209E-05	1,762E-05	7,854E-06	1,475E-05	
c_el	=	-1,050E-06	2,362E-06	-5,003E-08	2,998E-06	-4,546E-07	
d_el	=	1,326E-03	5,207E-04	1,085E-03	1,150E-04	4,845E-04	
e_el	=	-4,104E-03	-2,919E-03	-3,586E-03	-2,243E-03	-2,982E-03	
f_el	=	-8,353E-04	-1,348E-03	-9,861E-04	-1,462E-03	-9,515E-04	
a_rad	=	-8,304E-06	-2,668E-06	-6,589E-06	2,394E-07	-3,053E-06	
b_rad	=	2,209E-05	1,261E-05	1,824E-05	8,137E-06	1,492E-05	
c_rad	=	-8,946E-07	2,563E-06	8,893E-08	3,154E-06	-1,960E-07	
d_rad	=	1,364E-03	5,481E-04	1,127E-03	1,034E-04	4,618E-04	
e_rad	=	-4,173E-03	-2,981E-03	-3,670E-03	-2,259E-03	-2,985E-03	
f_rad	=	-8,187E-04	-1,338E-03	-9,643E-04	-1,449E-03	-9,417E-04	
a_Kap	=	0,000	-0,002	-0,001	-0,002	-0,003	
b_Kap	=	-0,057	0,320	0,123	0,326	0,463	
c_Kap	=	2,306	-12,216	-5,028	-12,613	-17,489	
d_Kap	=	0,000	0,002	0,000	0,002	0,002	
e_Kap	=	0,081	-0,181	-0,028	-0,221	-0,324	
f_Kap	=	-6,896	3,296	-0,971	5,542	8,955	
g_Kap	=	0,000	0,000	0,000	0,000	0,000	
h_Kap	=	-0,066	-0,082	-0,088	-0,055	-0,057	
i_Kap	=	334,911	333,904	333,653	333,397	333,066	

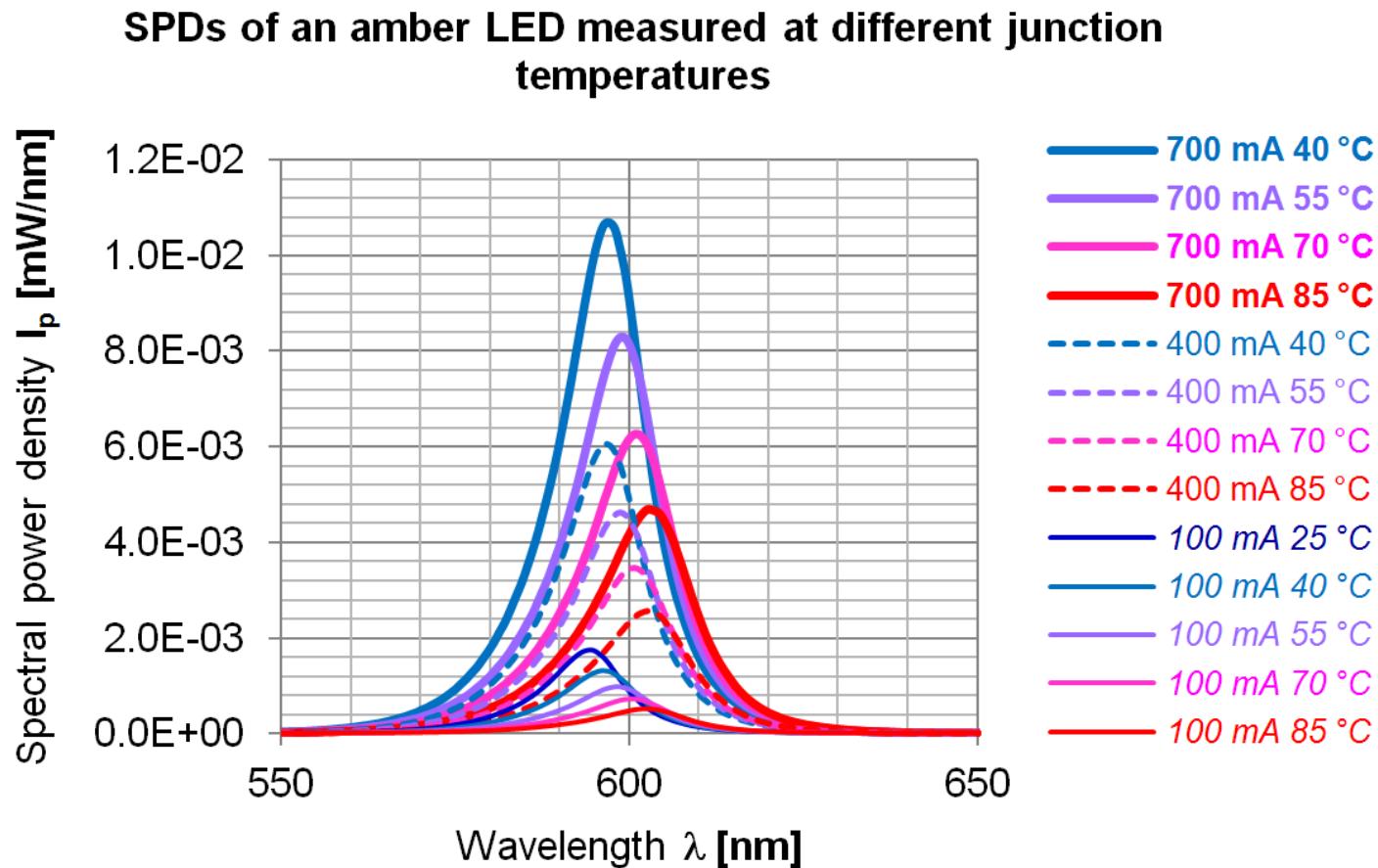


TEMPERATURE DEPENDENCE



Peak wavelength, intensity vs. T_J

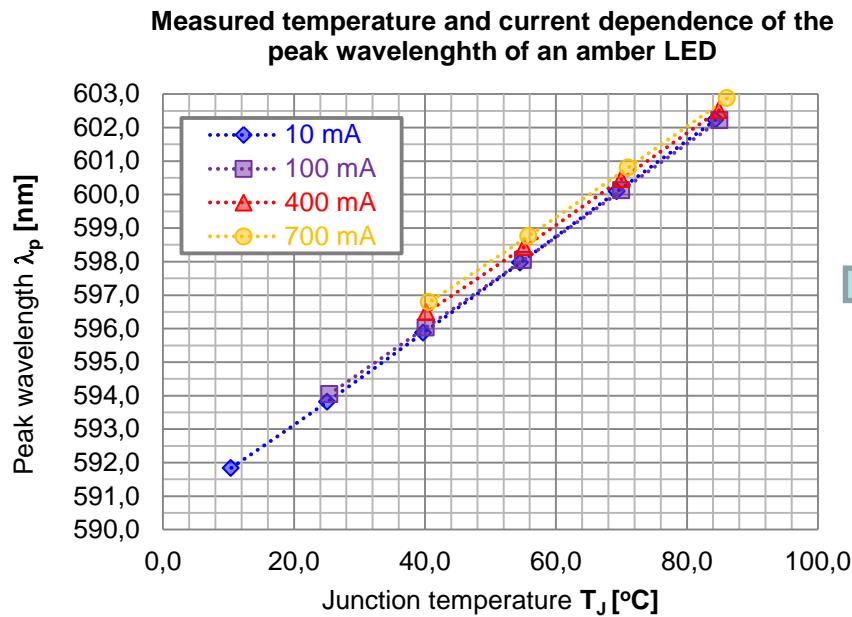
- ▶ Peak wavelength shifts
- ▶ Intensity diminishes



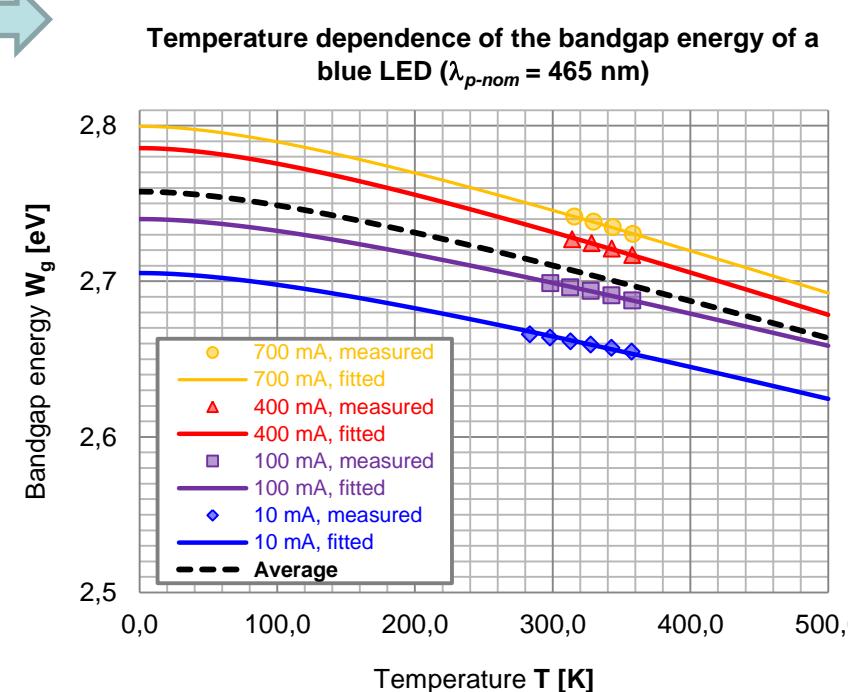
Bandgap vs. T_J

► Peak wavelength shifts

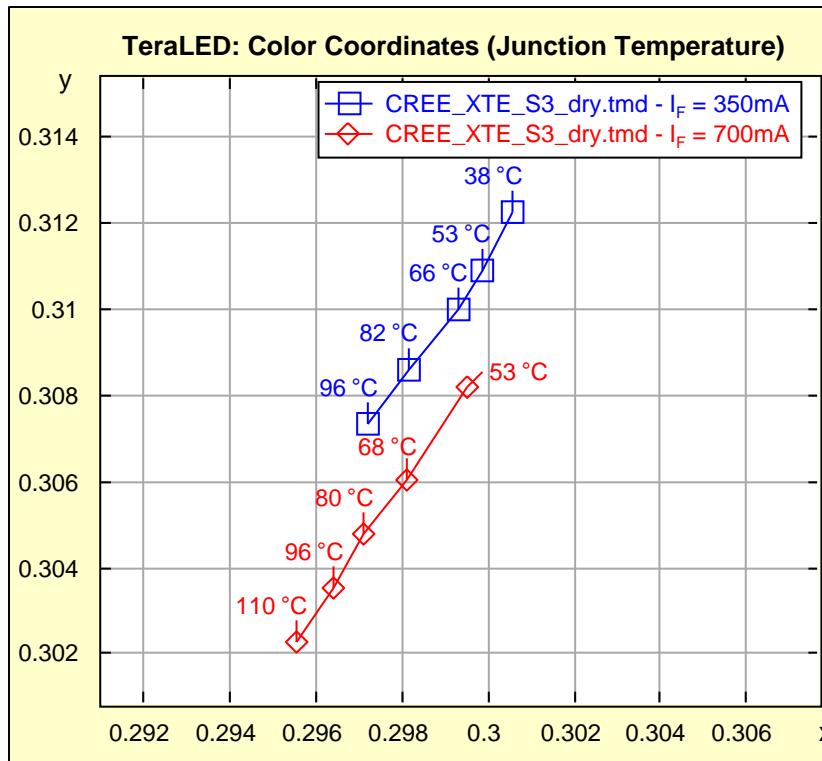
- Due to the temperature dependence of W_g



$$W_g(T) = W_{g0} - \frac{\alpha \cdot T^2}{\beta + T}$$

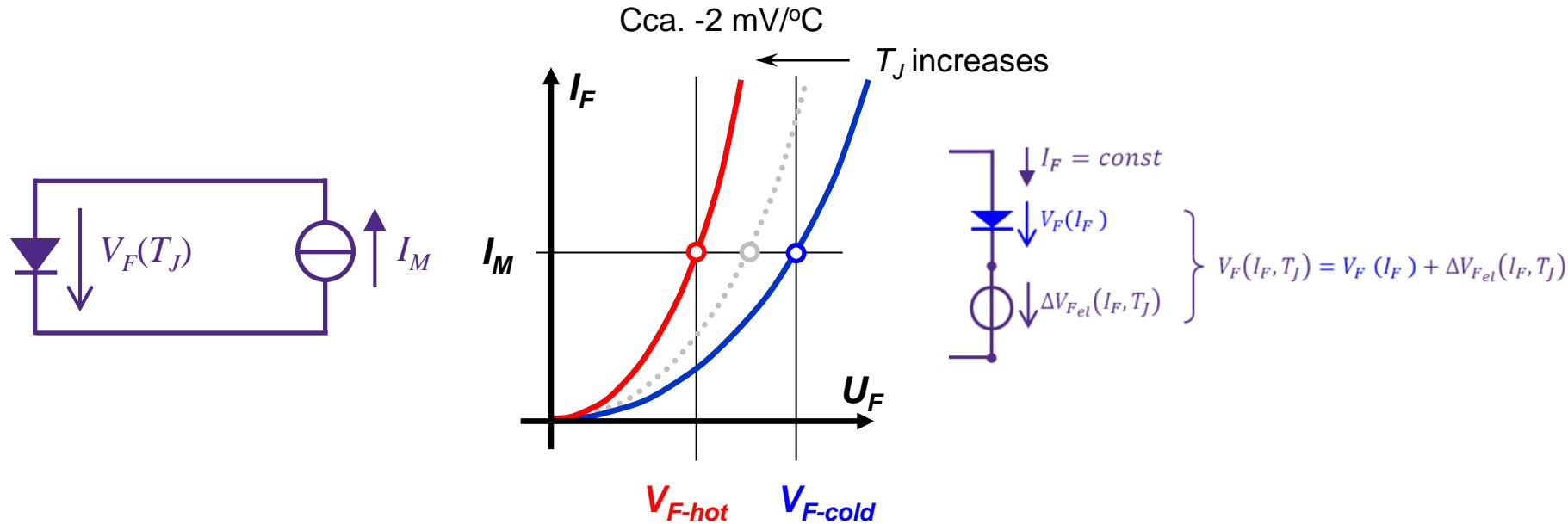


Color coordinates (CIE 1931 2° xy)



Basic powering: current driven scheme

- The forward voltage depends on T_J (almost linearly)

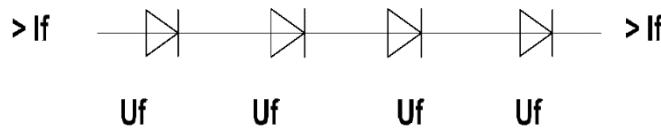


- ... all other properties (efficiency, efficacy) are also T_J dependent
- These are more stable in current driven operation than in voltage driven mode
 - *Recap from diodes:*

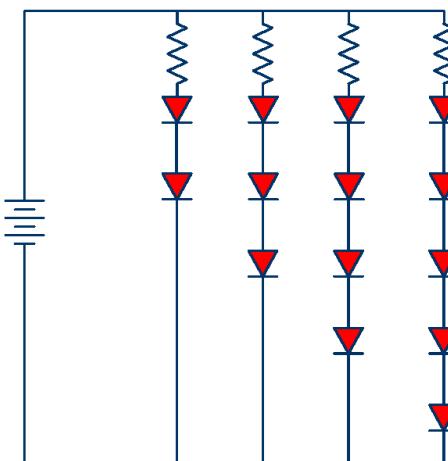
$$\Delta V_F = 60 \text{ mV} \rightarrow 1 \text{ order of magnitude change of } I_F$$

Consequence: *design of the electrical environment*

- ▶ Multiple LEDs must always be connected in series!
 - The forward current is the same, resulting in the
 - same brightness and color



- ▶ The simplest approximation of a current generator: large **load resistor**
 - The current can be well set by the load resistor
Such LED strings can already be connected in parallel:



Source— LED Light For You course, 2006 (Electrical basics for LED)

Consequence: design of the electrical environment

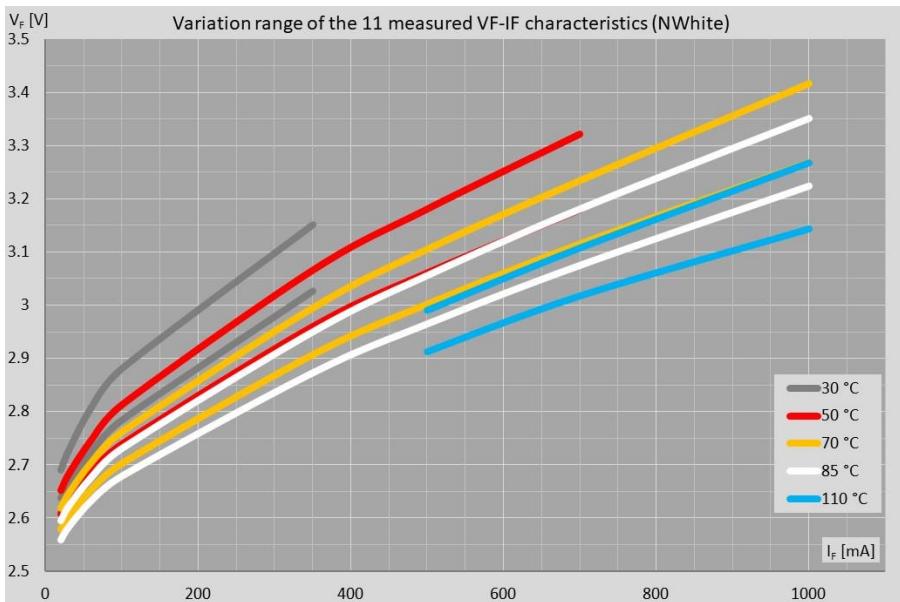
- ▶ LEDs connected in parallel



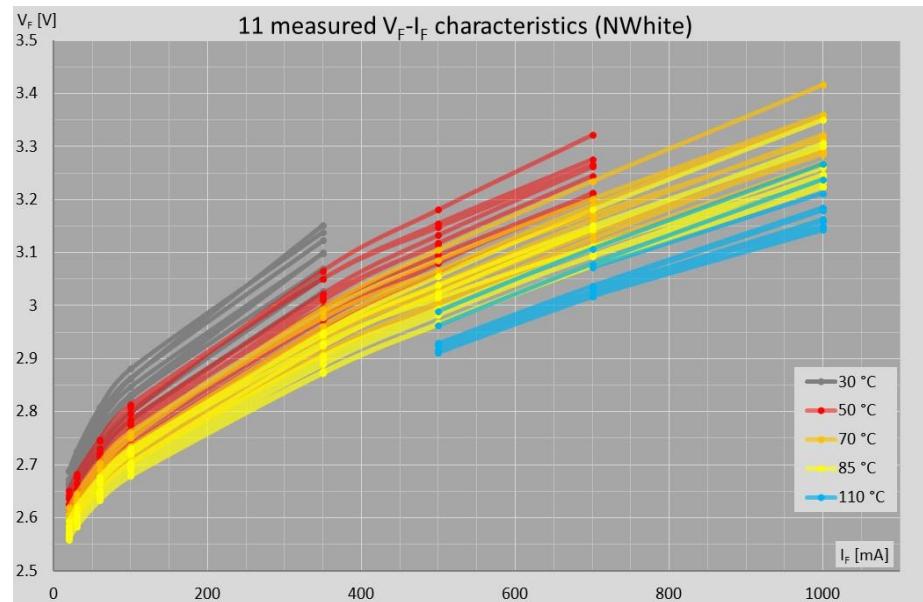
- The overall voltage is determined by the LED with the smallest V_F
- Huge differences of the forward currents of the different LEDs will develop
- **Due to these differences some LEDs will be overheated and will die quickly**
- Some LEDs may not emit light at all (e.g. if they are from a different bin)
- Binning at in-line test is never perfect, therefore there is always a scatter of V_F . The differences are more pronounced if the applied I_F is smaller than the binning I_F .

Source—LED Light For You course, 2006 (Electrical basics for LED)

XPE2 PCW IVL from BME and extraction



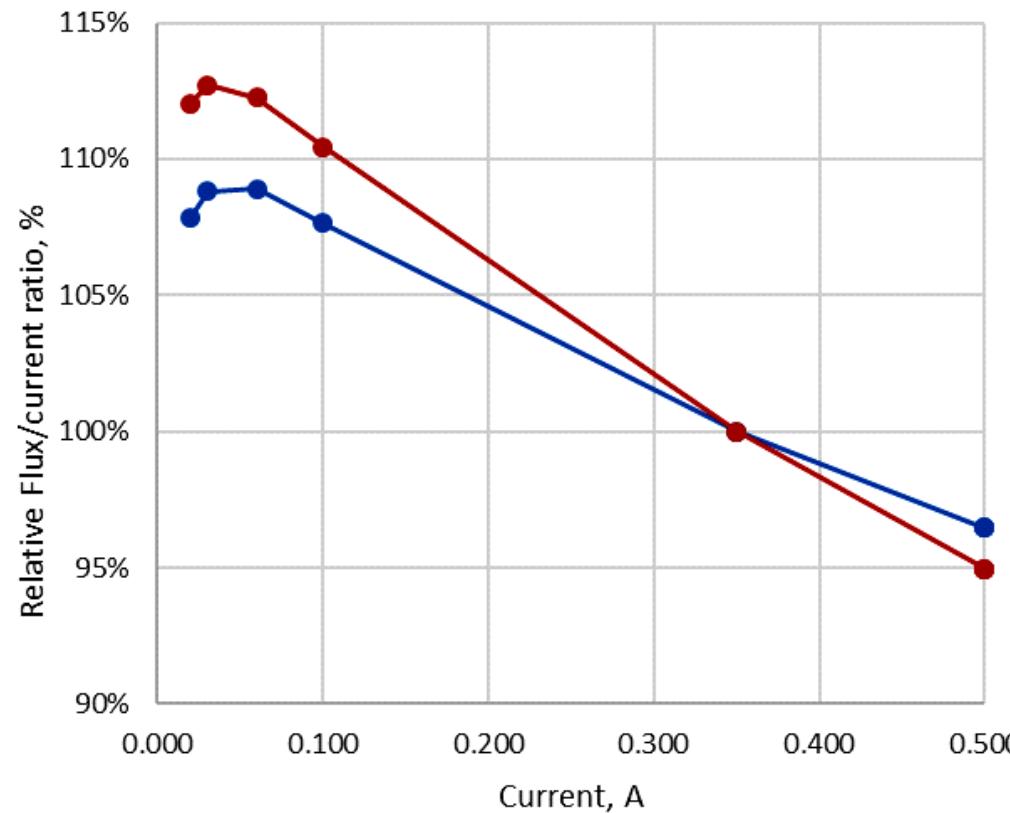
2 extreme LEDs at each temperature



all LEDs at all temperatures

- Overlap between the sets of characteristics belonging to adjacent temperatures. This blurs the picture significantly

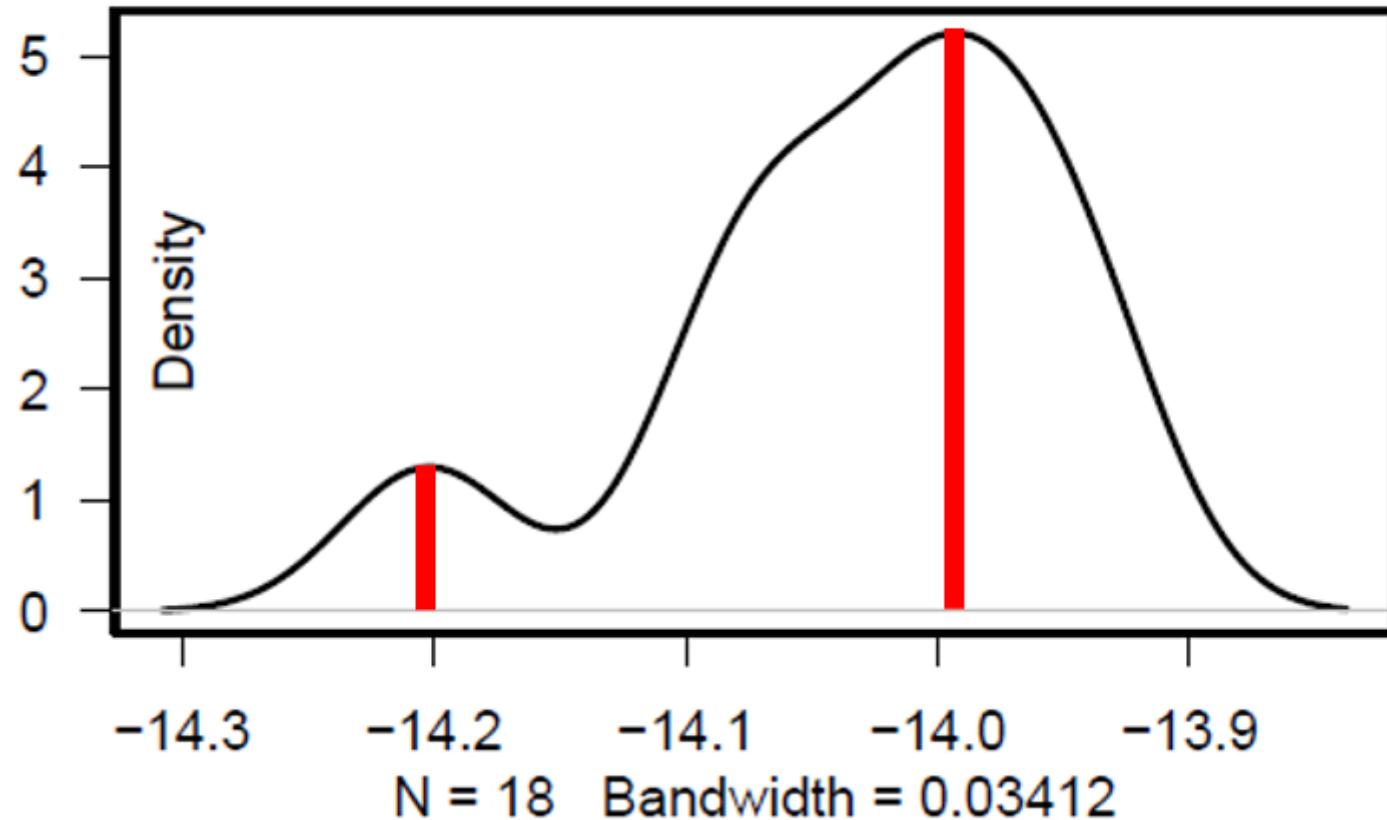
Multiple “LED types” in one bin



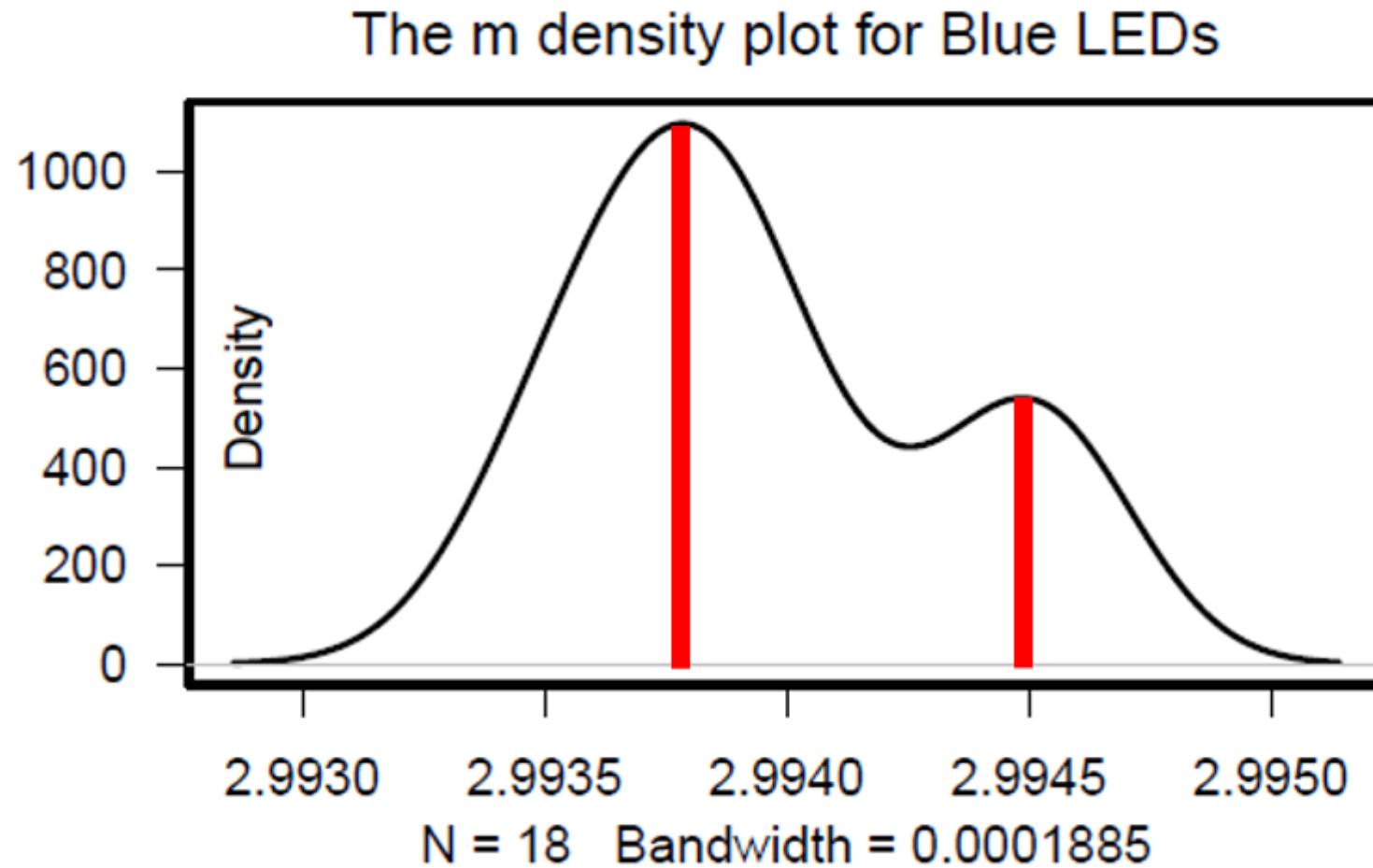
The $[\Phi_e(I_F, T_J)/I_F](I_F)$ and $[\Phi_v(I_F, T_J)/I_F](I_F)$ curves are like fingerprints of LED manufacturing processes

Multiple “LED types” in one bin

Log₁₀ density plot for Blue LEDs

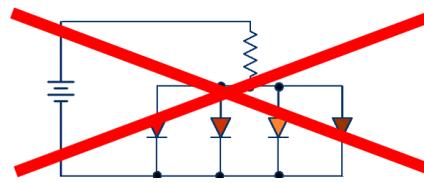


Multiple “LED types” in one bin



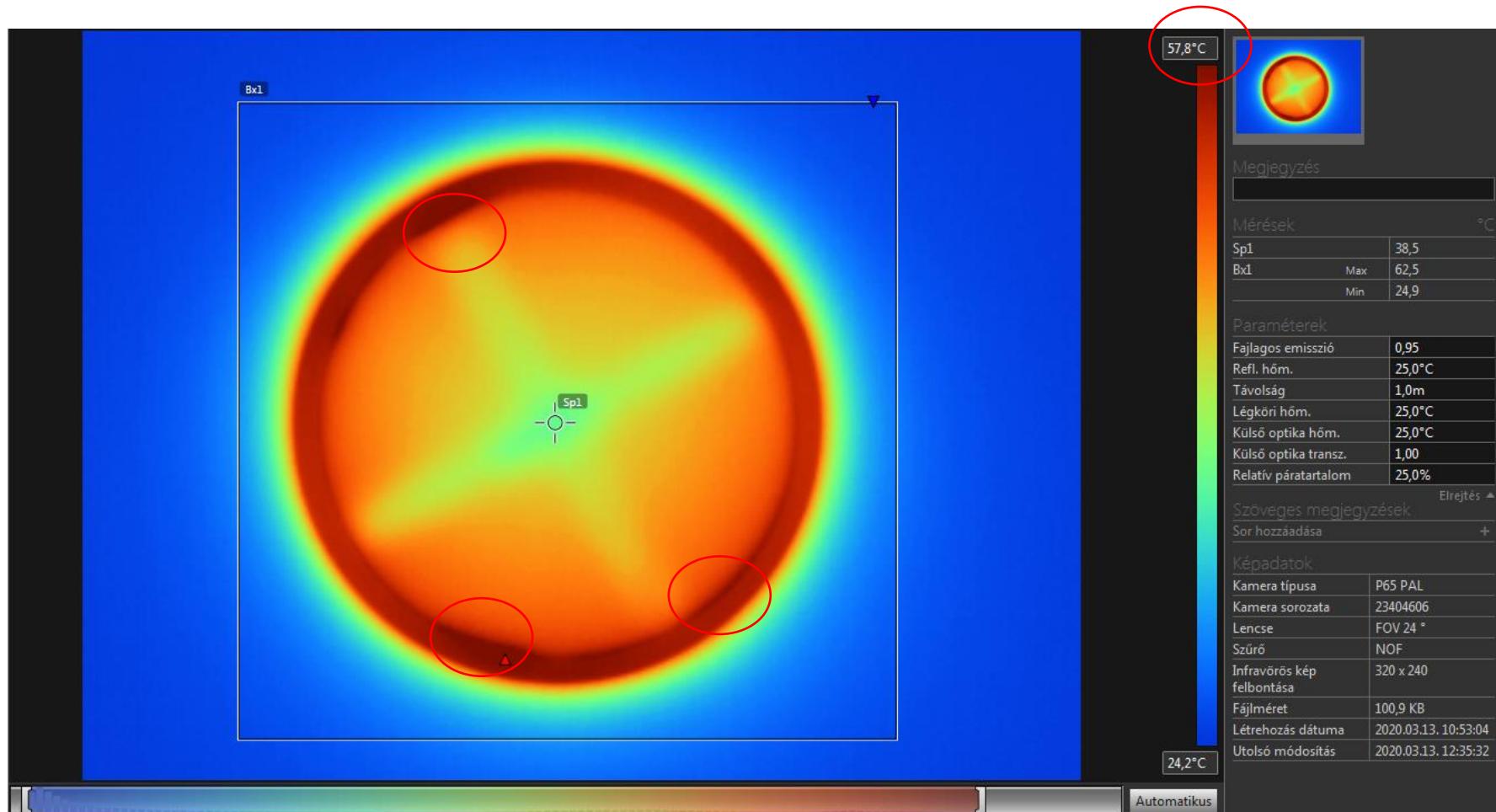
Photos of a luminaire with wrong circuitry

- Some regions got burnt due to local overheating:



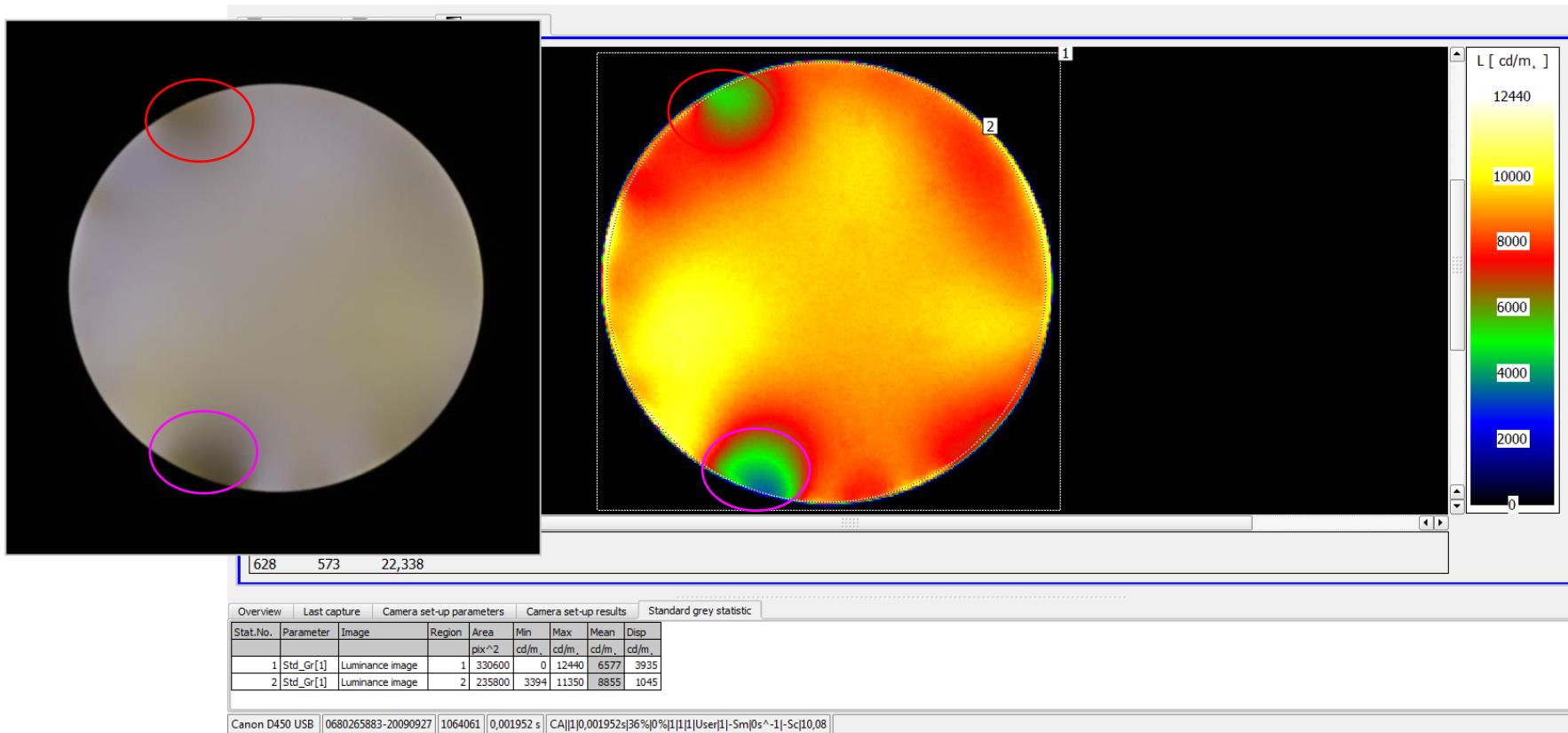
IR image of the same LED lamp

- Highest temperature measured by IR camera on the metallic frame of the luminaire 57.8 °C.



Normal photo and measured luminance maps

- At the hot spots ~50% drop in luminance was measured:



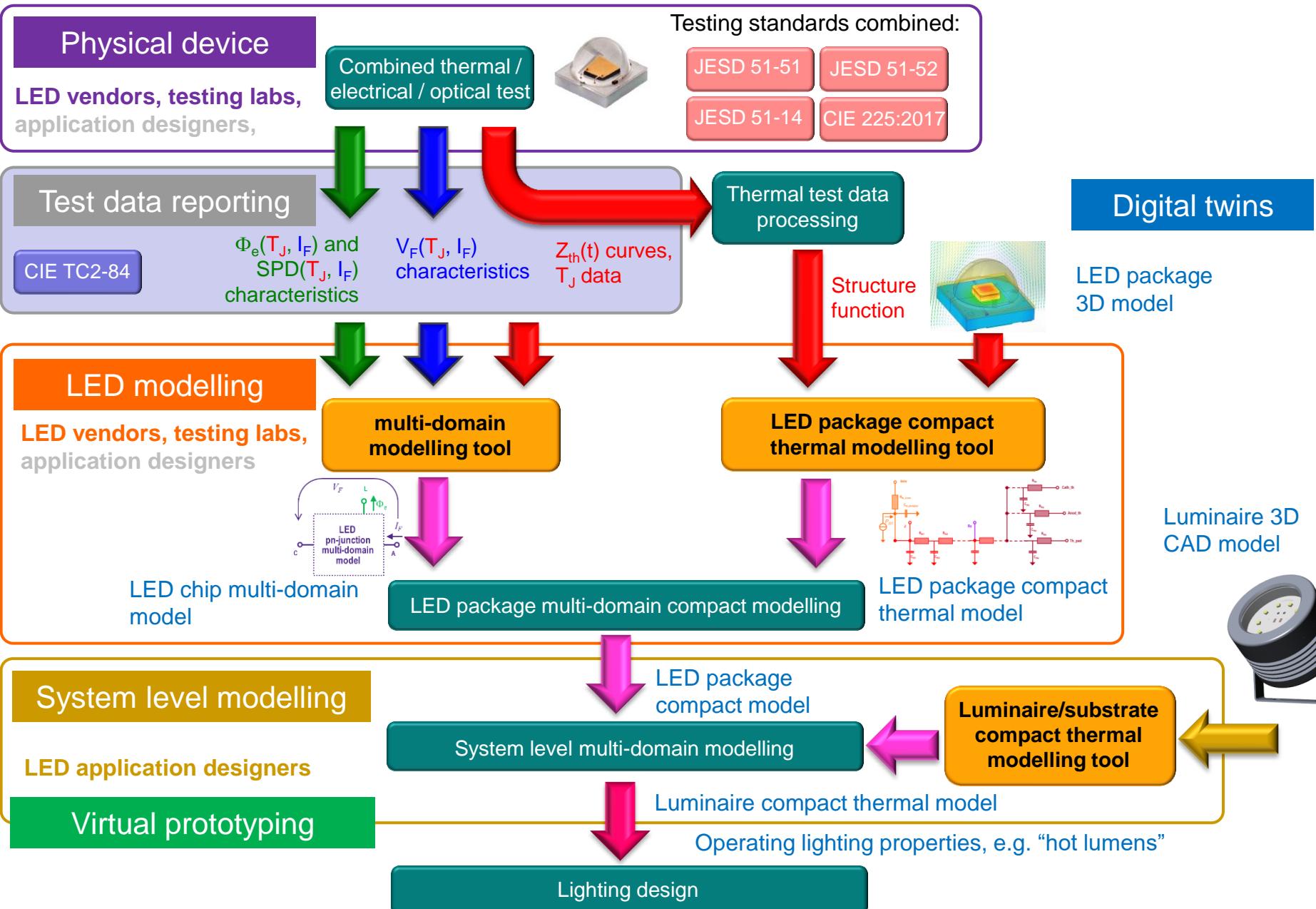
Fatal failures

- ▶ Burnt diffusor plane
- ▶ LED lenses burnt
- ▶ LED carrier tape delaminated and burnt



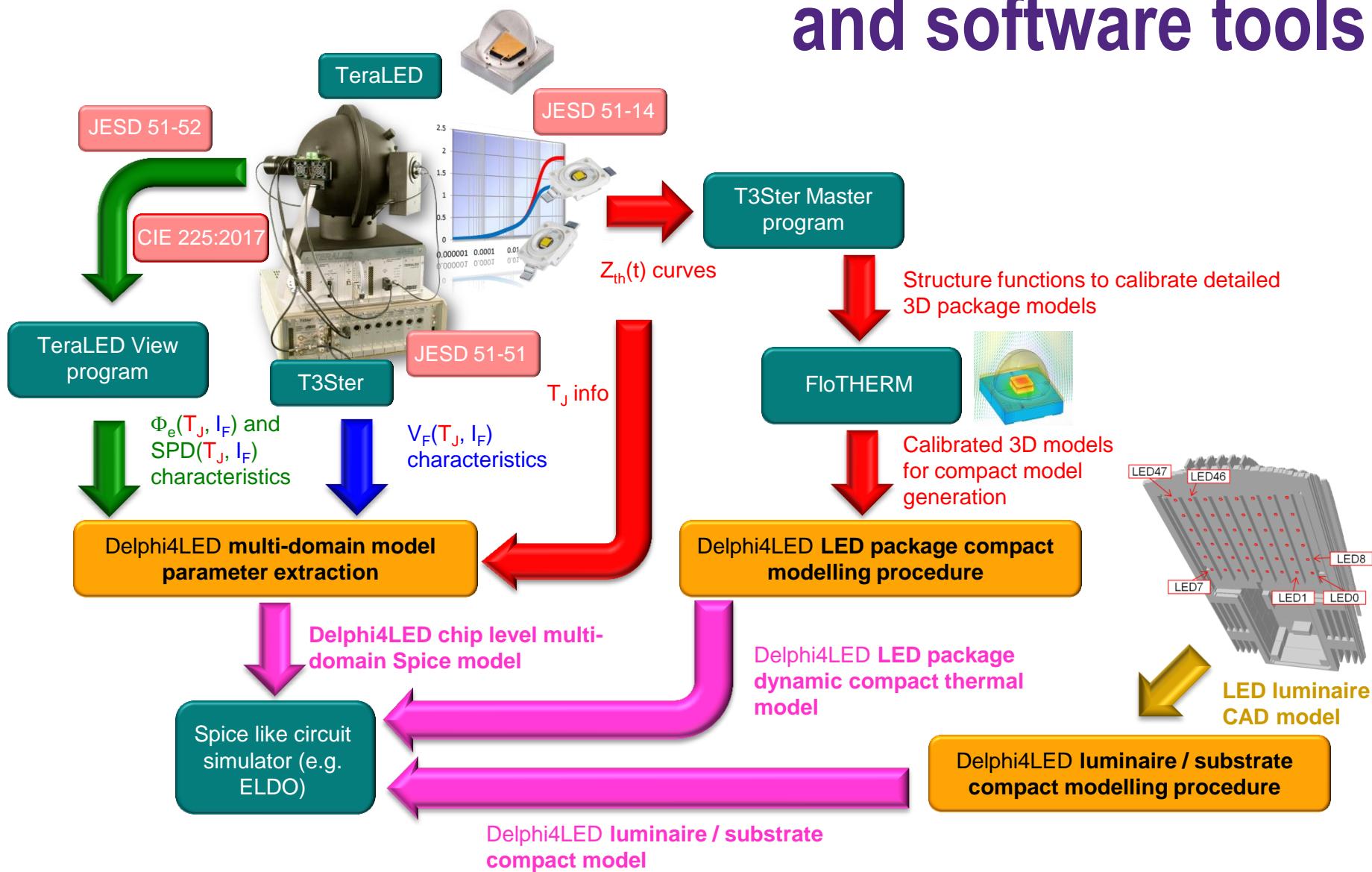
MEASURING AND MODELLING OF LED-S IN AN INDUSTRY 4.0 APPROACH

The Delphi4LED Industry 4.0 workflow



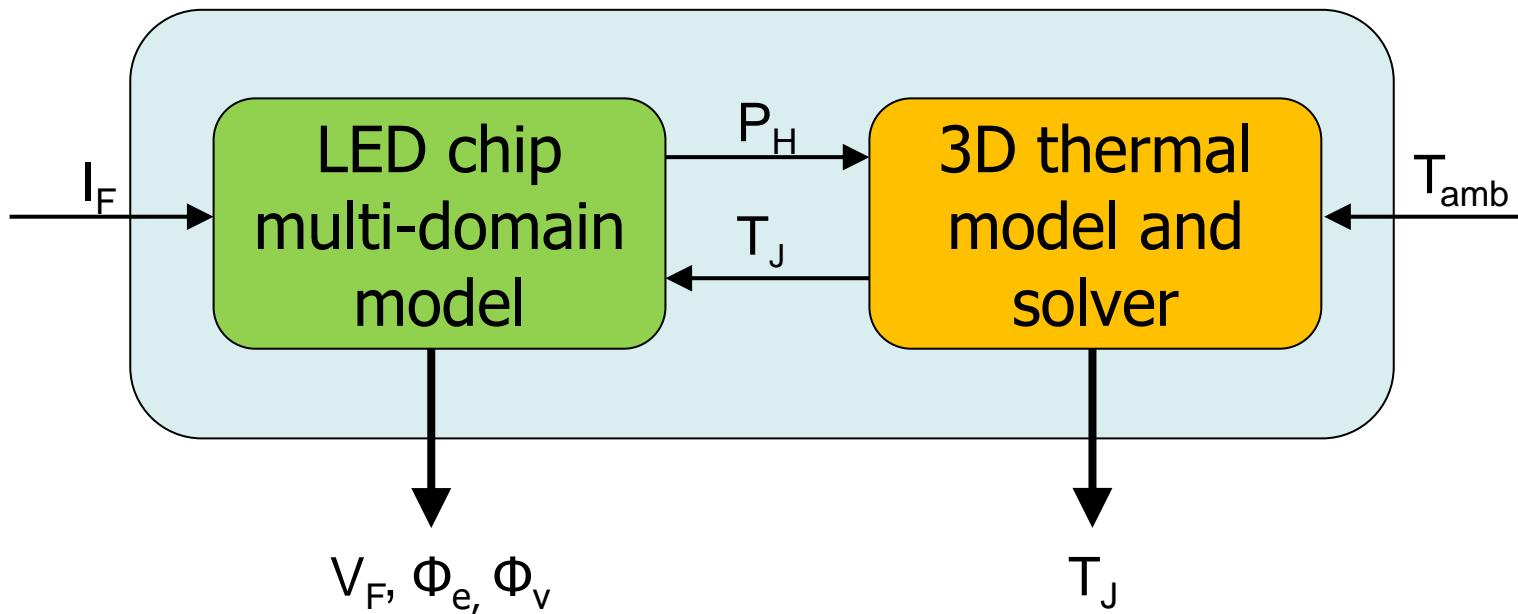
Realized by

SIEMENS hardware and software tools



All models integrated: *luminaire design tool*

- ▶ Co-simulation between LED chip multi-domain model and the thermal model of the 3D environment (substrate, luminaire)
- ▶ Used for different project demonstrators
 - Support of both chip level LED models
 - Support of different parameter sets for these models



Demo tool: *Luminaire Design Calculator*

Microsoft Excel application:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	GOALS													
2	Luminous Flux (lm)	1200												
3														
4	CONSTRAINTS													
5	T _j max (degC)	150												
6	T _s max (degC)	85												
7	T _a (degC)	45												
8	Total Power Consumption (W)	15												
9														
10	DESIGN													
11	Number of LEDs	6												
12	Substrate Type	SMI 3W												
13	Heatsink R _{th} (K/W)	2.5												
14	Forward Current (A)	1.5												
15	Optics Efficiency	0.8												
16	Driver Efficiency	0.9												
17														
18														

SIMULATE

RESULTS	
Highest T _j (degC)	85.38
Highest T _s (degC)	79.27
Total System Power Consumption (W)	29.96
Total System Luminous Flux (lm)	2465
Total Luminous Flux from LEDs (lm)	3082
Total System Optical Power (mW)	7582
Total Optical Power from LEDs (mW)	9478
Total System Lumens/Watt (lm/W)	82.28
Total Lumens/Watt from LEDs (lm/W)	113.1

PER LED RESULTS	V _f (V)	Luminous Flux (lm)	T _j (DegC)	T _s (DegC)	P (W)	P _{dis} (W)
1	3.03	513.47	85.38	79.27	4.5	2.96
2	3.03	513.69	85.17	78.84	4.5	2.96
3	3.03	513.61	85.25	79.00	4.5	2.96
4	3.03	513.63	85.23	78.96	4.5	2.96
5	3.03	513.62	85.24	78.98	4.5	2.96
6	3.03	513.65	85.21	78.92	4.5	2.96

Test results vs. simulation

System electrical power and luminaire total luminous flux comparison (new process / Major)

GOALS

Luminous Flux (lm) 1200

CONSTRAINTS

T _j max (degC)	150
T _s max (degC)	85
T _a (degC)	25
Total Power Consumption (W)	15

DESIGN

Number of LEDs	5
Substrate Type	SMI 8W
Heatsink R _{th} (K/W)	2.5
Forward Current (A)	0.8
Optics Efficiency	0.8
Driver Efficiency	1

SIMULATE



RESULTS

Highest T _j (degC)	47.55
Highest T _s (degC)	43.33
Total System Power Consumption (W)	11.71
Total System Luminous Flux (lm)	1302
Total Luminous Flux from LEDs (lm)	1628
Total System Optical Power (mW)	3971
Total Optical Power from LEDs (mW)	4964
Total System Lumens/Watt (lm/W)	111.2
Total Lumens/Watt from LEDs (lm/W)	139

10,7 W
measured

1339 lm
measured

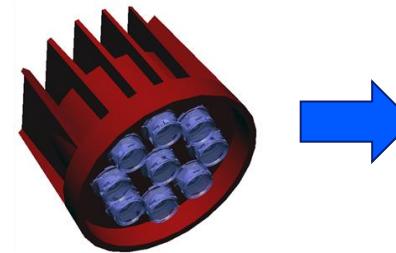
Cost benefits of using compact models

- Considerable savings in terms of development time and development costs

Main design costs	"SME" old process	"SME" new proces	Gain
Personal costs	0.896	0.633	29%
Material costs	0.049	0.028	43%
Testing	0.056	0.056	0%
Total	1.000	0.717	28%

Main design costs	"Major" old process	"Major" new proces	Gain
Personal costs	0.819	0.502	39%
Material costs	0.055	0.028	48%
Testing	0.126	0.045	65%
Total	1.000	0.575	42%

Demo experiments in Delphi4LED



- Luminaire designs also realized and physically tested

Open access summaries

<https://doi.org/10.3390/en12101909>

<https://doi.org/10.3390/en12122389>

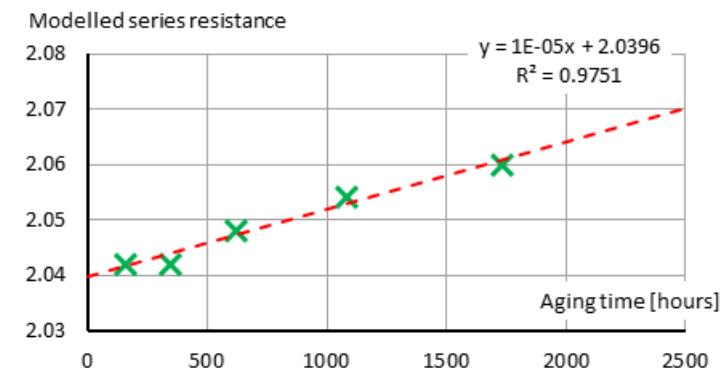
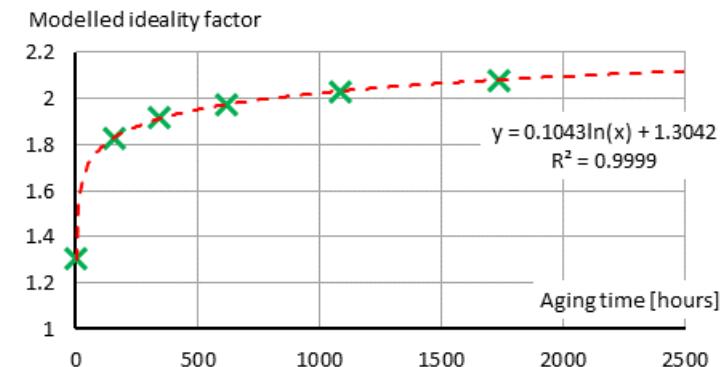
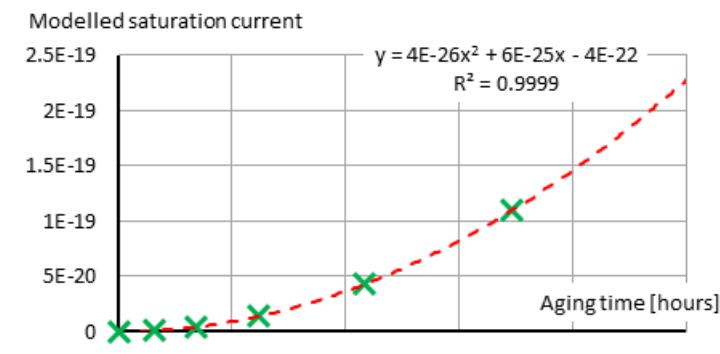
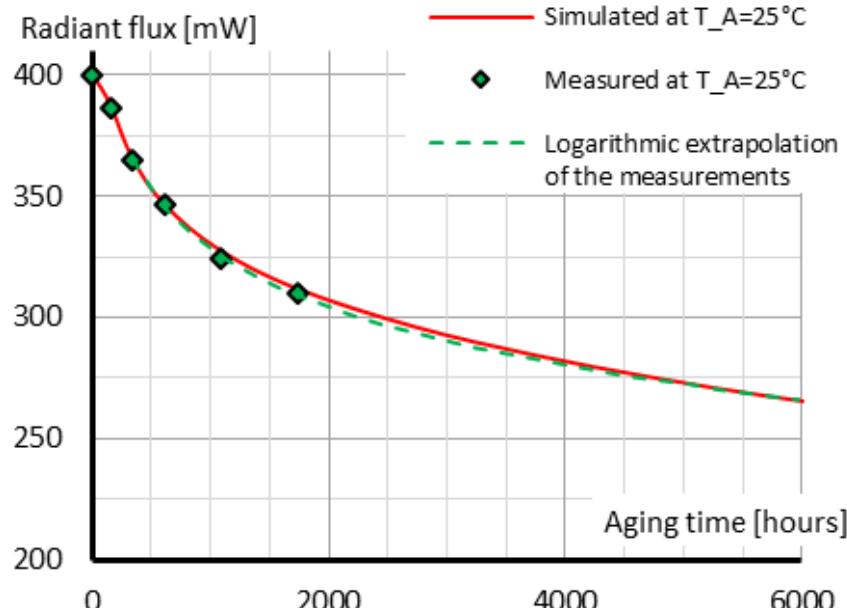
The screenshot shows a web browser window displaying an article from the MDPI Energies journal. The URL in the address bar is <https://doi.org/10.3390/en12122389>. The page title is "Luminaire Digital Design Flow with Multi-Domain Digital Twins of LEDs†". The article is marked as "Open Access" and "Article". The authors listed are Genevieve Martin, Christophe Marty, Robin Bornoff, Andras Poppe, Grigory Onushkin, Marta Renicz, and Joan Yu. The article is associated with Signify, Ingelux, Mentor, and Department of Electron Devices. It was received on May 9, 2019, revised on June 15, 2019, accepted on June 18, 2019, and published on June 21, 2019. The abstract discusses the challenges of designing LED luminaires due to limited information in LED datasheets and the need for reliable products.

The future beyond Delphi4LED

- **Standardize LED test data reporting**
 - Technical report of CIE TC2-84 is in an advanced state thanks to Delphi4LED, finalize the report and go for a standard
- **Include LED reliability / lifetime issues in LED digital twins**
 - First results based on LM-80 test data
 - Connect standard LED ageing tests of LEDs to real current, temperature and power cycles
- **Extend LED modelling** from total fluxes to metrics of light quality, such as “alphaopic” fluxes
- **On the overall benefits:**
- How precise LED modeling helps better design ‘artistic effects’ for end-users

Modelling beyond 0h of operation

- ▶ Elapsed lifetime in the multi-domain LED model (fixed current and temperature)
 - 100% ... 78% aging range (in terms Φ_V)
- ▶ Successful implementation of this very first approach – fitting LM80 test data
 - 0.5% absolute and
 - 1.2% maximum simulation inaccuracy



Circuit Simulation using LTspice

In this lab, we will use LTspice XVII, a widely used industrial circuit design tool. Since it is freeware, it is free to download and use, and there is no upper limit of nodes, components, or even sub-circuits.

Main parts:

- a) Schematic design editor
- b) SPICE simulation engine
- c) waveform display

Run simulations:

- a) time-domain (transient)
- b) small-signal (AC)
- c) large-signal (DC)
- d) large-signal transfer (DC Transfer)
- e) operating point calculation
- f) noise

The software is also equipped with the features needed to design switching power supplies, which is the main application of this tool today. However, it is not suitable to design printed wiring boards (in a discrete case), and neither to create the physical layout of integrated circuits (layout), nor for logical simulations.

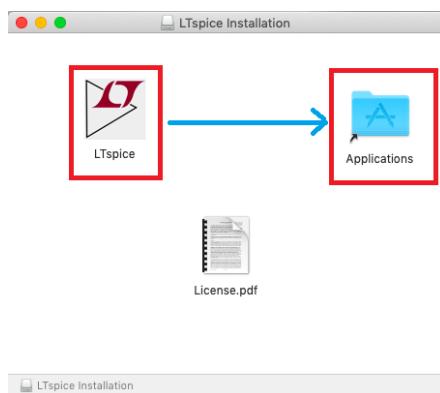
You can download the latest version of the software from the following link:

<https://www.analog.com/en/design-center/design-tools-and-calculators/ltspice-simulator.html>

Around the middle of the page, click on *Download for Mac 10.9+*. A 116.9 MB installer will start to download. Once it is done, install it on your computer.

Installing LTspice on Mac:

After downloading the installation file to your computer, Launch the .dmg installer, a window will appear, drag & drop *LTspice* into *Applications*.



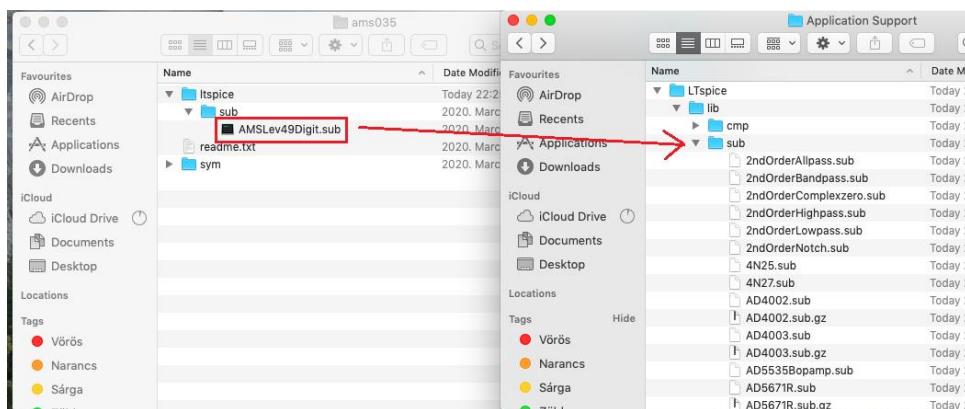
A window will appear showing the the copying progress of 'Ltpice.app' to 'Applications'.

In order to design integrated circuits provided by Austria Microsystems (which are N and P channel MOS transistors), we need to add additional symbols and models to the contents of the lib folder. To do this, you need to download the AMS components from the EDU system. You will find two folders. Inside the sym folder you can find an *AMScells* subfolder containing two .asy files, which are no less than hierarchical symbols (NMOS, PMOS) edited with a graphical editor. You can see that the symbol drawings are described in a unique format. The subfolder contains an AMSLev49Digit.sub file which is no less than the description of the transistor model in SPICE language. Here, we are talking about the BSIM3 level 49 transistor description, which is a quite complex model with many parameters.

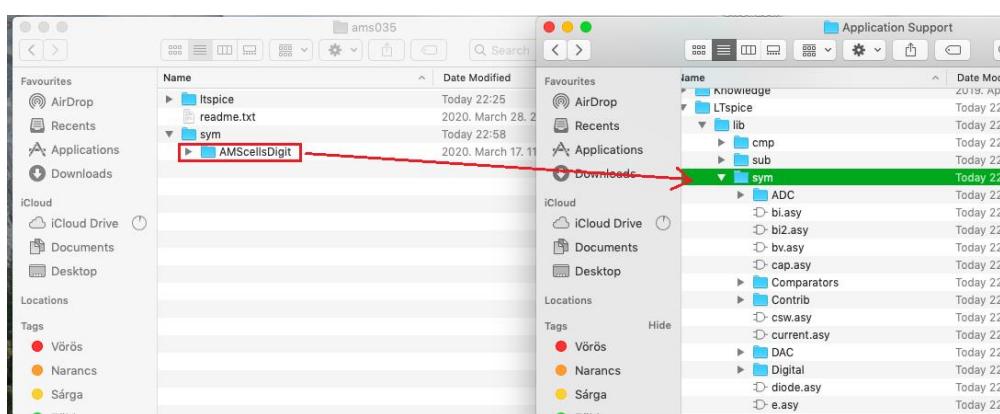
Once the installation is finished, open a new *Finder* window ($\text{⌘}+\text{N}$) and go to your *home* folder. Press $\text{⌘}+\text{Shift}+\text{.}$ [dot] to show hidden files.

Navigate to *Library > Application Support > LTspice > lib*

Drag & drop the file from *ams035/Ltspice/sub* to the 'sub' folder of your Macintosh HD.

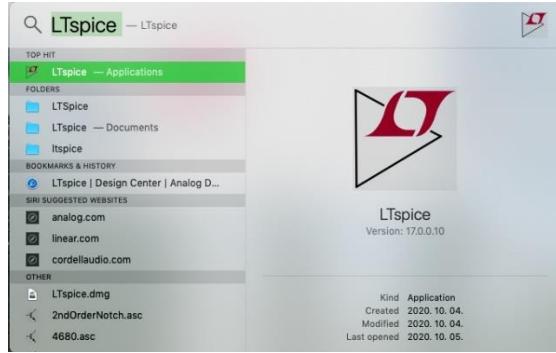


Repeat the step for the 'sym' folder.

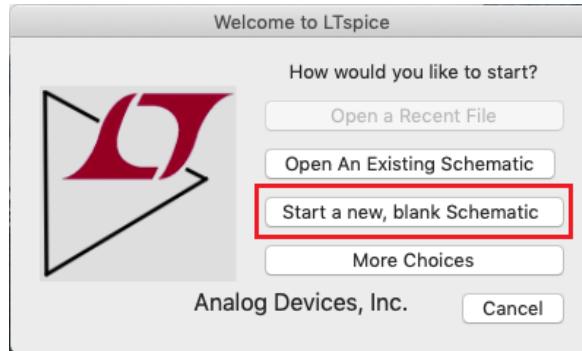


Press $\text{⌘}+\text{Shift}+\text{.}$ [dot] again to hide system folders.

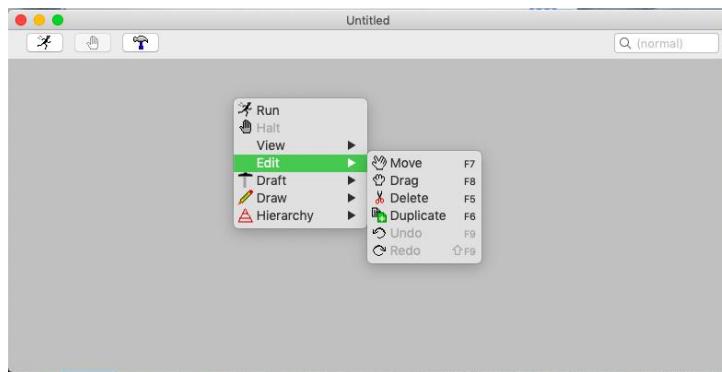
Now use the *spotlight* search ($\text{⌘}+\text{Space}$) to launch Ltspice, type in „ltspice”, hit **Return** and the software will launch.



A window called '**Welcome to LTspice**' appears everytime you launch the software. In here you can choose between opening an already existing schematics or to start a new blank one from the scratch, other options also exist. Choose '*Start a new, blank Schematic*'.

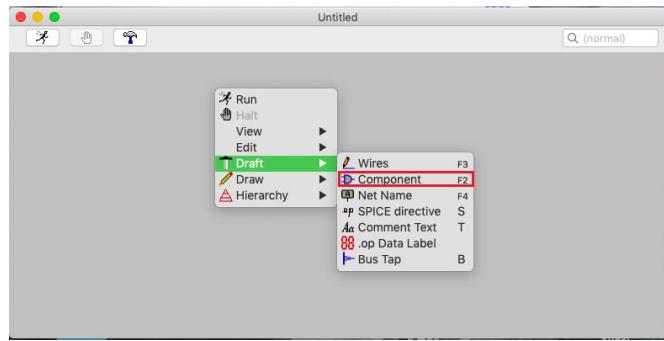


LTspice on Mac environment has different style than on Windows. For instance, on Mac, there is no menu bar on the top of the schematic workplace. By right click anywhere in the empty area of your schematic workplace, you can browse the options and tools that LTspice offers for the user (adding components, simulation,...,etc.)

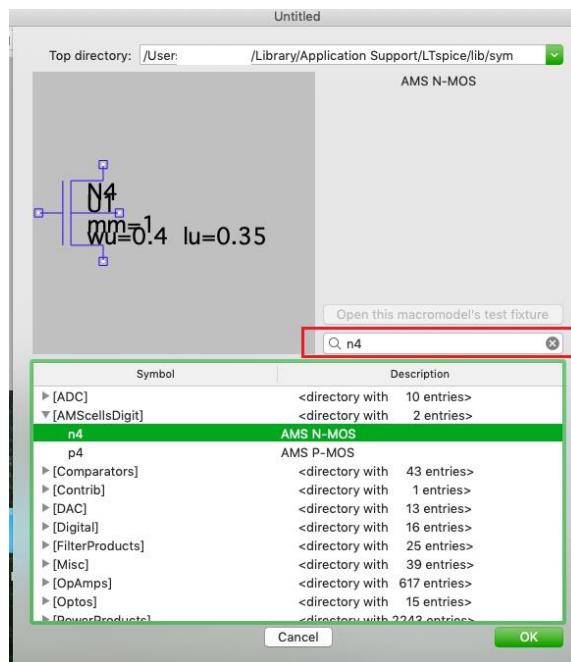


Drawing CMOS Inverter Schematic:

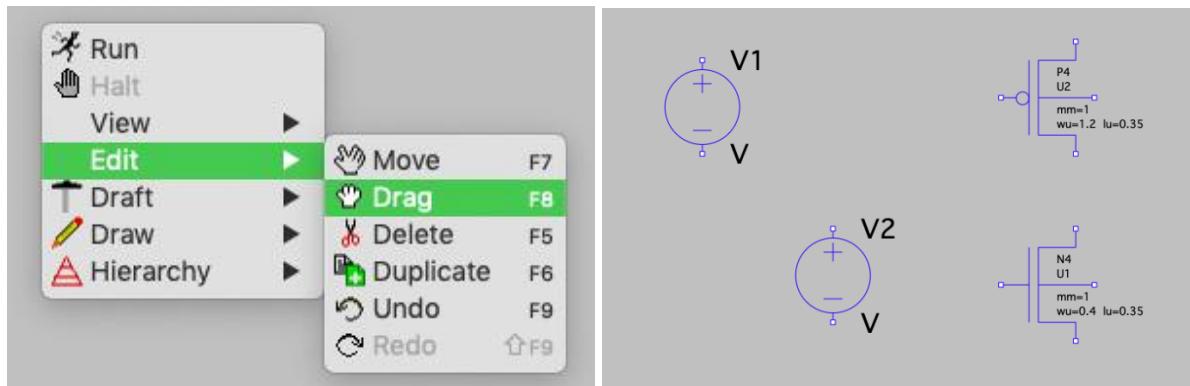
Right click on the empty workplace, and navigate to *Draft > Component* (for simplicity you can directly press **F2** on your keyboard).



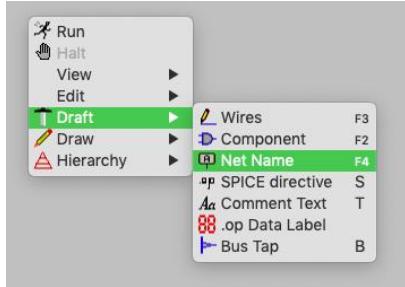
From the components library, we need to insert the NMOS Transistor, you can navigate to it from the folder *AMScellsDigit > n4*, or a faster way is to type **n4** in the search field. Choose **n4** and press *OK*. Now you can place your component anywhere in the workplace. Repeat the procedure to add the **p4** component (the PMOS transisitor).



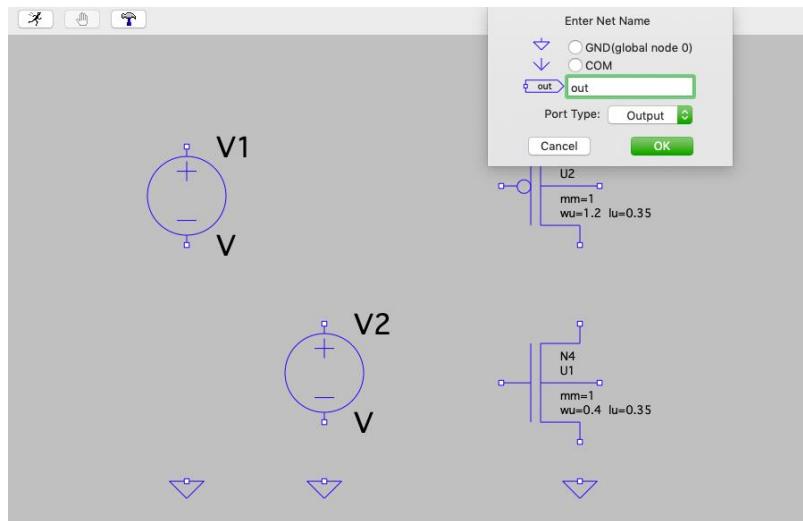
Now we need to add two voltage sources to our schematic. Navigate to *Draft > Component* and type in the serach field: **voltage** and click *OK*. Place two volage sources to the scematic. You can rearrange your components using the option *Drag* from the *Edit* menu.



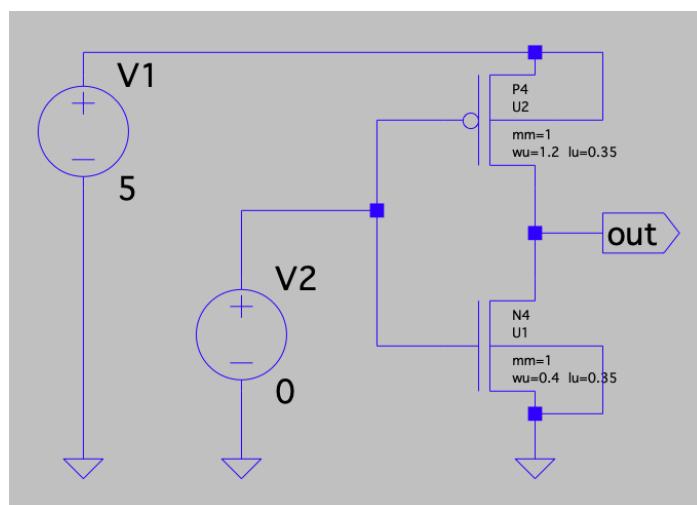
For the Ground (GND) terminal, choose the option *Net Name* from the *Draft* menu (or press **F4** on your keyboard). Choose the option  GND(global node 0) and hit *OK*. (Place 3 GND terminals).



Next is to add an output terminal, again from *Draft > Net Name*, call the port 'out' and choose the type to be *Output*.



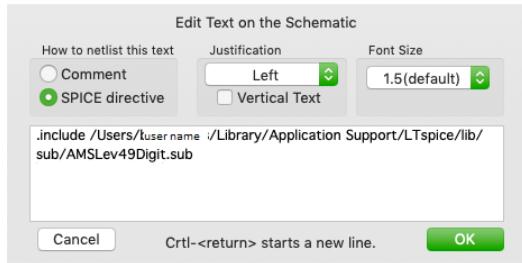
Right click on the voltage sources to set their DC value. Set **V1** to be **5V** and **V2** to **0**, use the option *Wires* in the *Draft* menu (or press **F3**) to connect the components as following:



Now we have to click on **SPICE directive** from the *Draft* menu (or press **S**). Here we can define the path of the model file. Insert this line below:

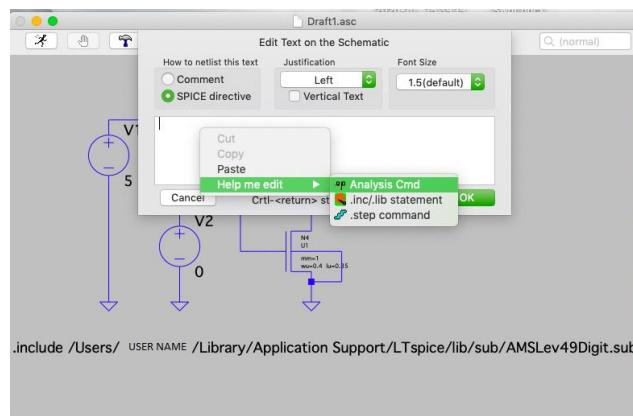
```
.include /Users/[YOUR USERNAME]/Library/Application Support/LTspice/lib/sub/AMSLev49Digit.sub
```

And place it somewhere on the schematic.

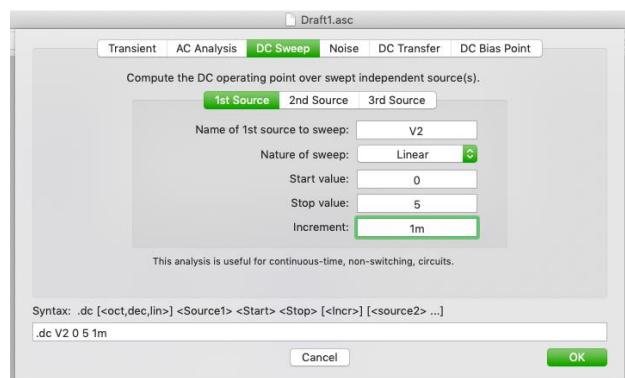


DC Simulation

click on **SPICE directive** from the *Draft* menu (or press **S**), right click on the text field and choose *Analysis Cmd*. Now the simulation settings menu will appear.



Choose the simulation type **DC Sweep**, fill the simulation settings as in the figure below. Click on **OK**.



Simulation tasks for the thermal laboratory

1. A $2 \times 2 \times 0.35\text{mm}$ Si substrate of an integrated circuit is soldered to a 1mm thick Cu plate. The bottom of the Cu is at constant 25°C . Place a 5W dissipating transistor on the chip.
¹Try three different cases: put the dissipator in the center of the substrate, put it in the center of at one edge and place it in the corner. Compare the results, summarize the experiences!

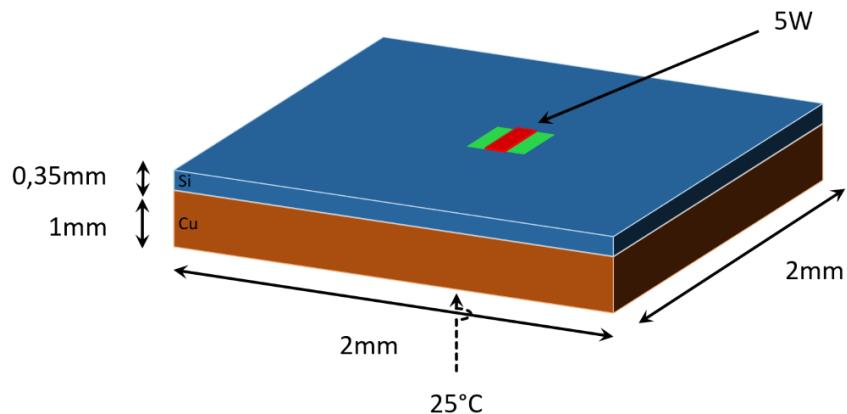


Figure 1.

2. The dimensions of the substrate of an operational amplifier chip are $1 \times 1 \times 0.3\text{ mm}$. The bottom side of the chip is cooled to some extent that can be accounted by setting the heat transfer of the lower side of the substrate to convection with a heat transfer coefficient $h=2000\text{W/m}^2\text{K}$. The transistor of the output stage ($P=0.1\text{W}$) is $400 \times 50\text{ }\mu\text{m}$, the transistors of the input differential amplifier are $25 \times 25\mu\text{m}$ ($P=0.001\text{W}$)². Position the three transistors so that the dissipation of the output transistors does not cause a temperature difference between the transistors of the input differential amplifier. Provide the adequate layout topology and explain it.

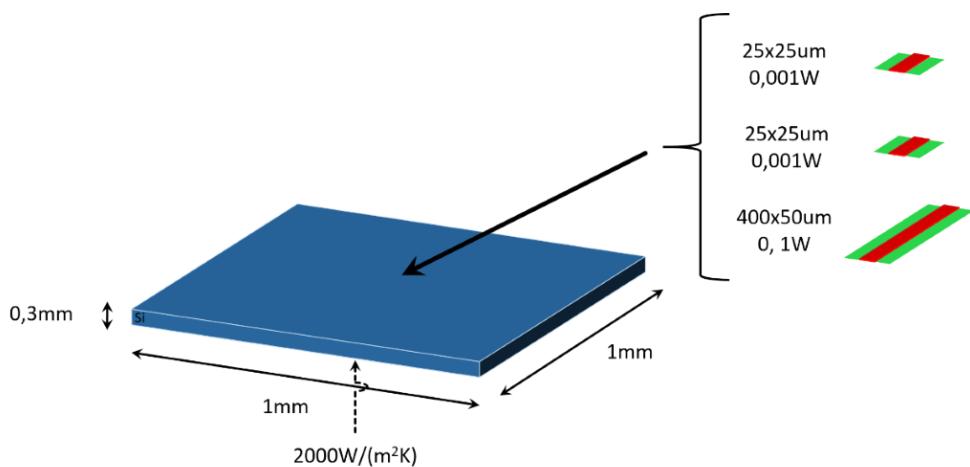


Figure 2.

¹ Transistors can be modeled with a rectangular dissipator.

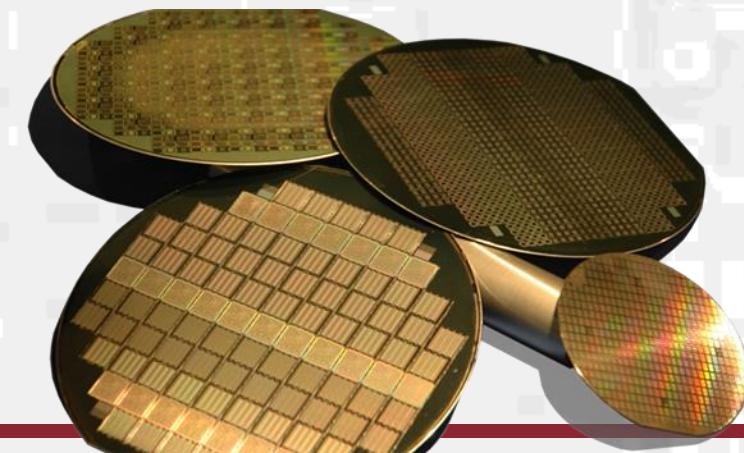
² Please refer to the document titled Laboratory Practice: Thermal Issues of Integrated Circuits, where phenomenon is discussed in detail



Budapest University of Technology and Economics
Department of Electron Devices

Solar cell working principles

Balázs Plesz



<http://www.eet.bme.hu>

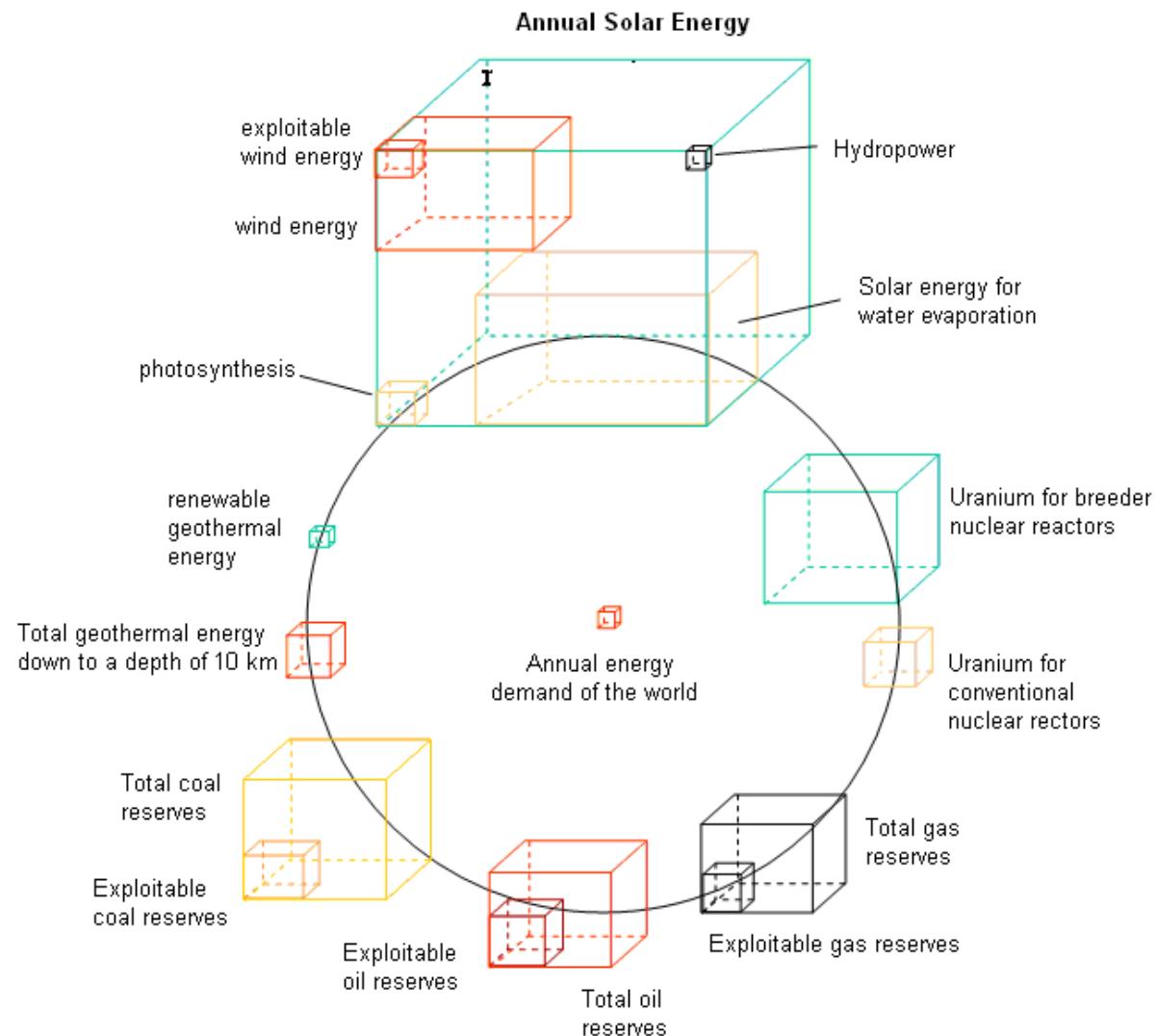
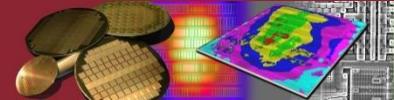


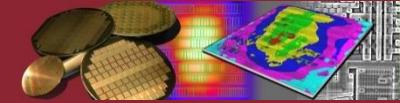
Motto

"One day, a person would no more think about buying a house without solar than they would a house without plumbing."

– Bob Clearman, Dow Chemical







Photovoltaic systems

- ▶ Stand alone or grid connected systems?
 - Reliability
 - Reconsideration of the tasks of the grid
 - A single plant for every user?
- ▶ Tracking or fixed Installation?
 - Higher energy gain with tracking
 - Higher costs
 - Shading
 - Higher maintenance





Dilemma of solar cell usage

High efficiency
solar cells with
concentrators

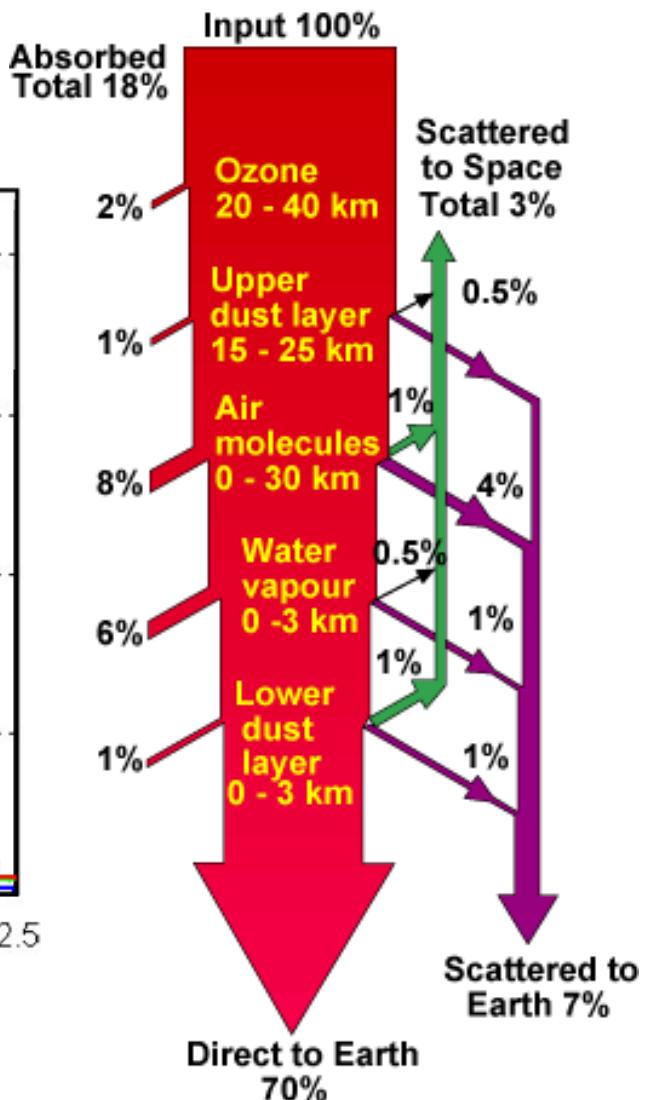
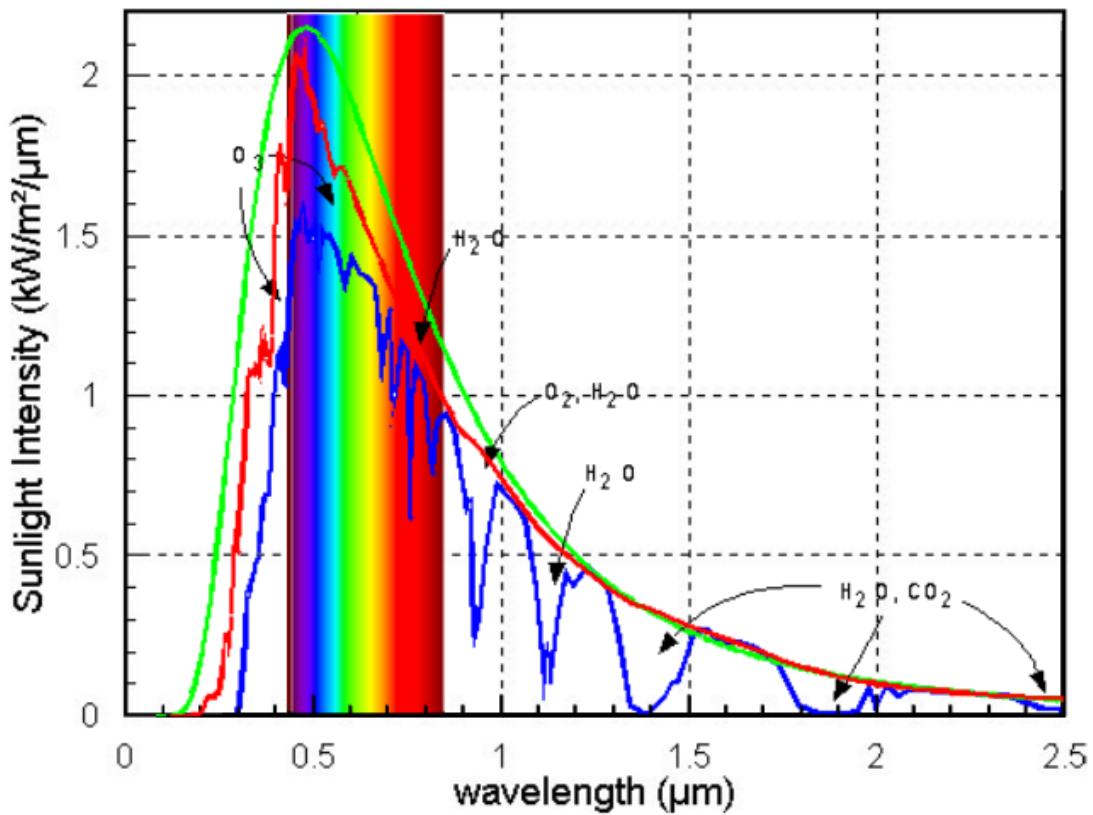


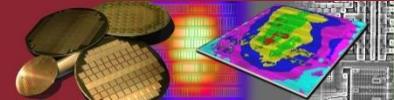
Large arrays
of low cost
solar cells



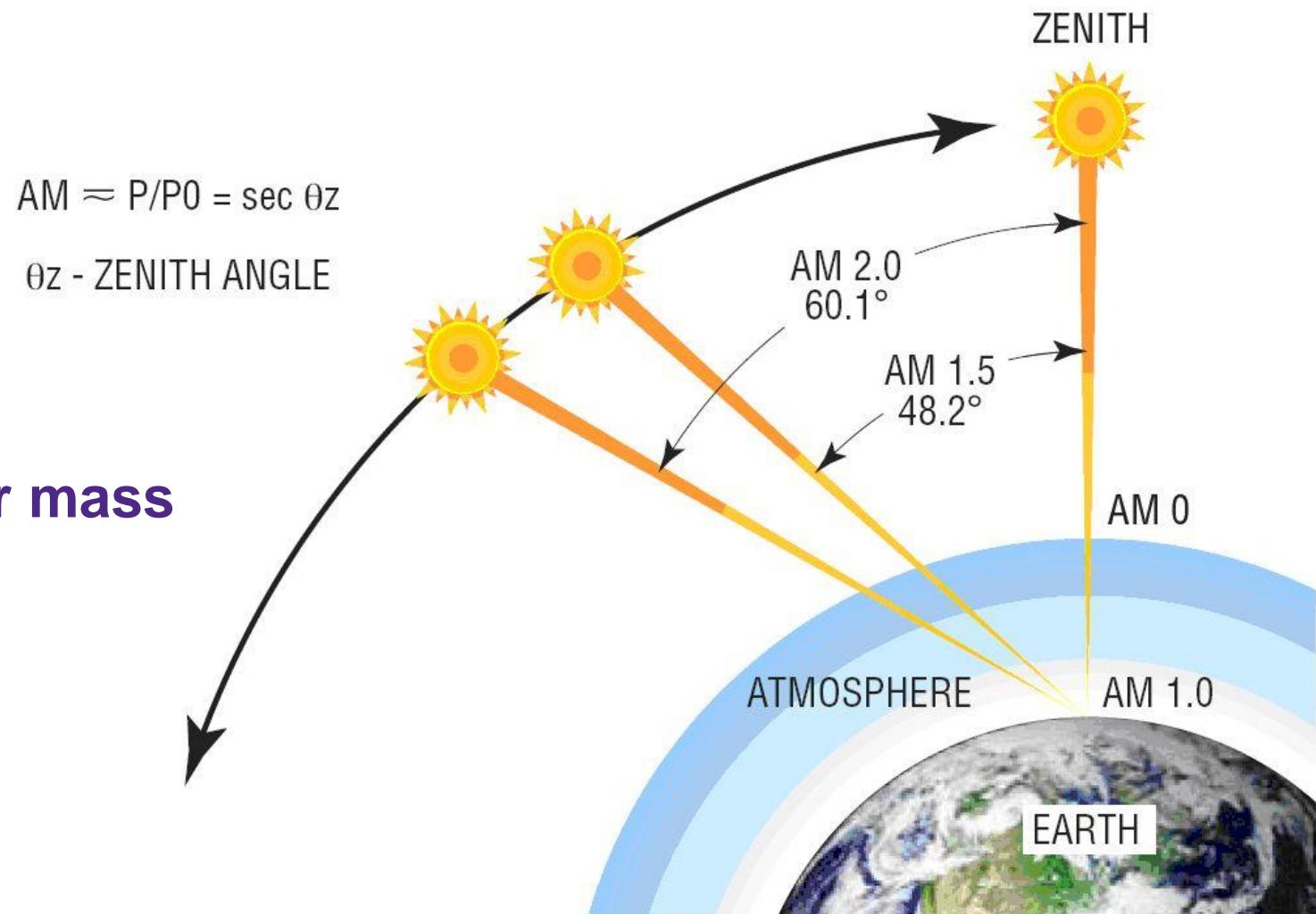


Solar energy properties





Solar radiation on the surface of Earth





HOW DO SOLAR CELLS WORK?





What is a solar cell?

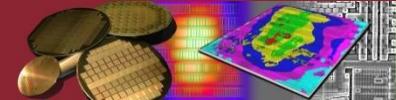
► Definition:

- Solar cells are devices that use the photoelectric effect to convert solar irradiation directly to electrical energy.

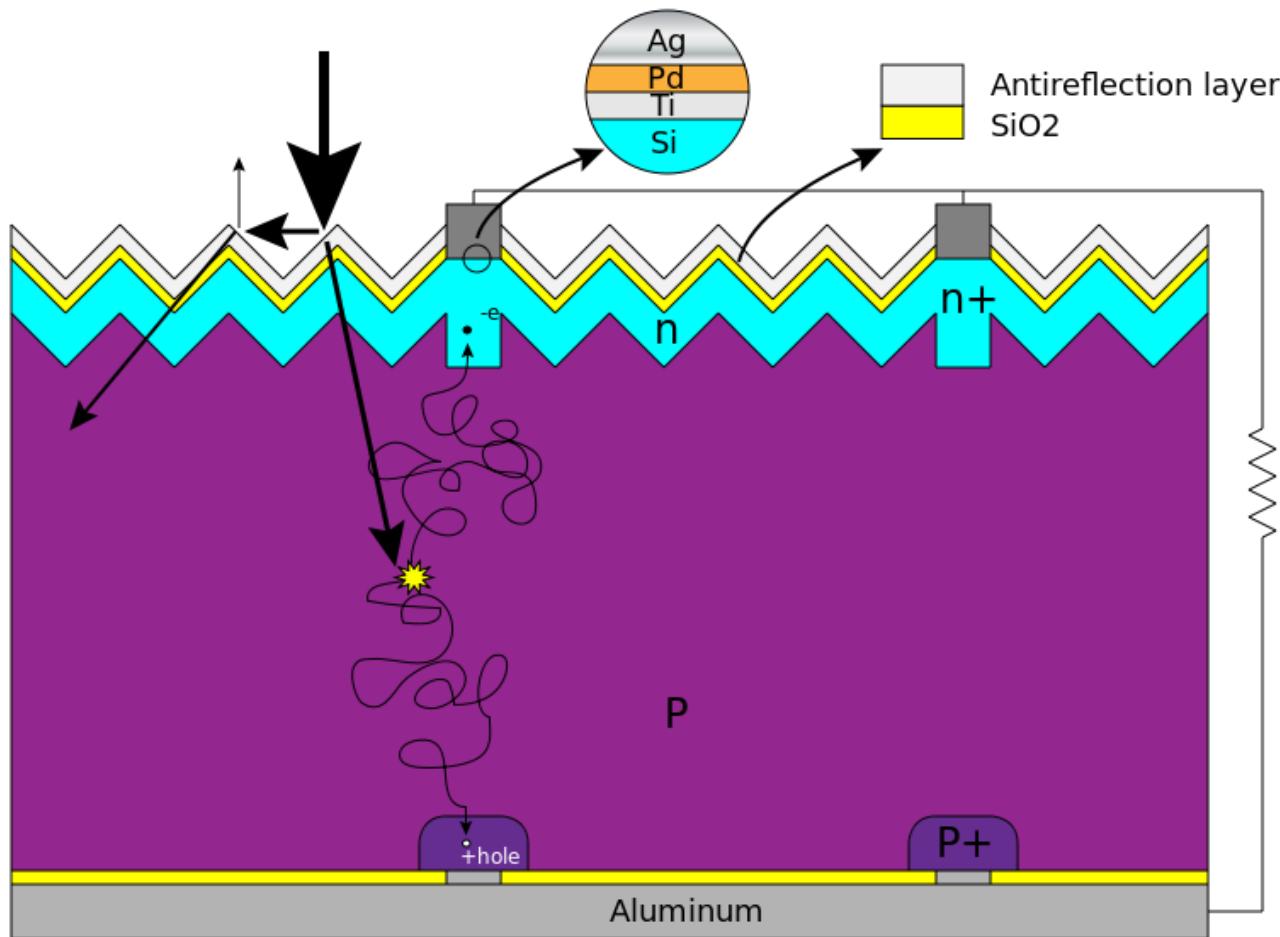
► Practical implementation:

- An illuminated semiconductor diode (p-n junction).

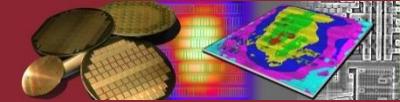




Solar cell basic structure



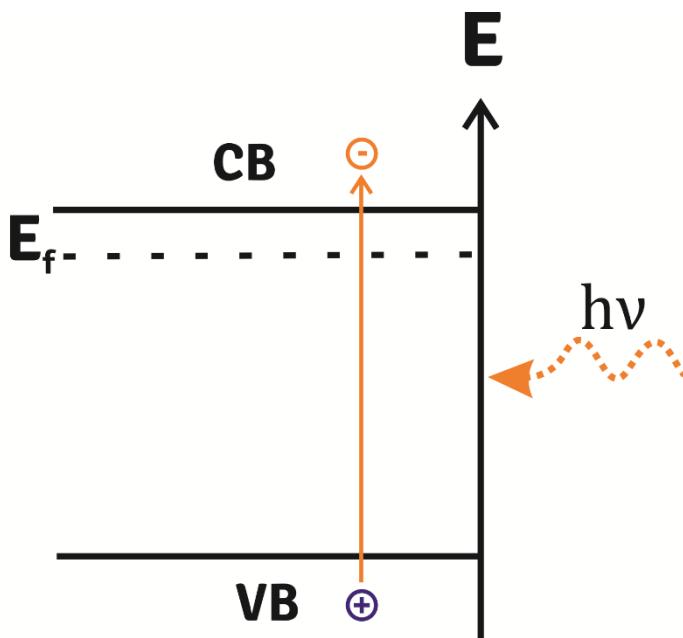
$$V_{pn} = V_T \ln \frac{N_d N_a}{n_i^2}$$

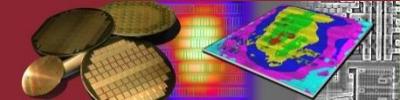


How do solar cells work?

► Photoelectric effect

- Incoming photons excites an electron from the valence to the conduction band (photogeneration)
This happens only if the energy of the photon is higher than the bandgap of the semiconductor ($h\nu > W_g$)

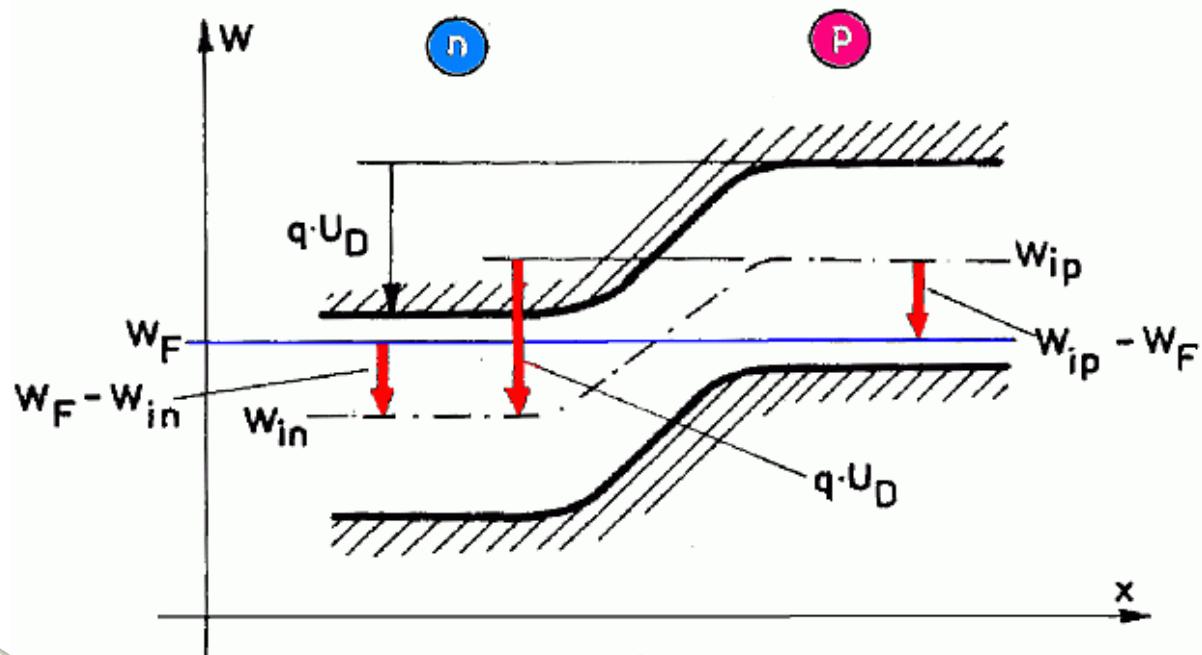




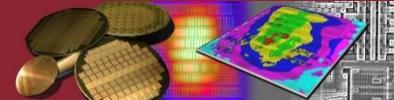
How do solar cells work?

► Built-in electric field

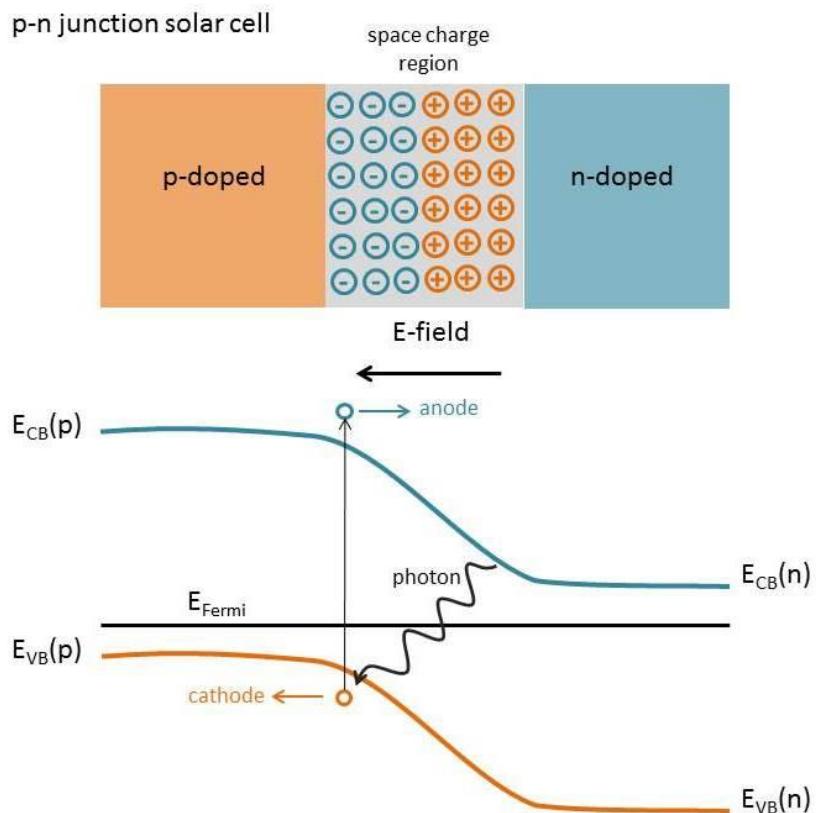
- The different dopings on the two sides of the p-n junction result in an electric field in the depleted region .



$$U_D = U_T \ln \frac{N_d N_a}{n_i^2}$$



How do solar cells work?



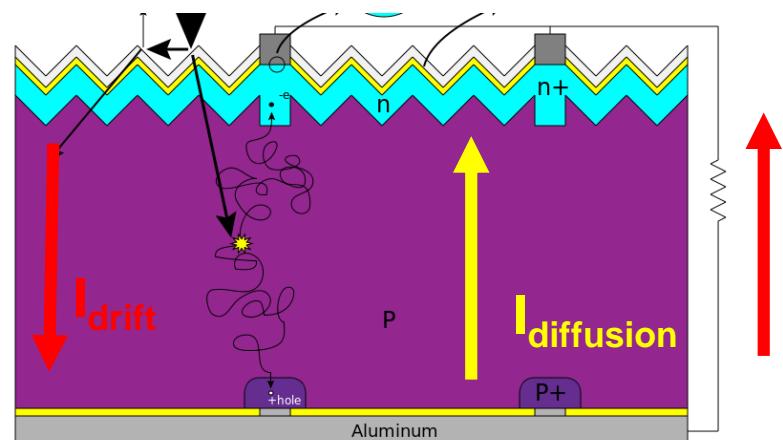
Photon with higher energy than the bandgap excite electron-hole pairs. The built-in potential of the p-n junction separates the charge carriers, and drives the electrons to the n-side and the holes to the p-side (drift current). Thus negative charge accumulates on the n-side and a positive on the p-side.





How do solar cells work?

- ▶ If we put an external resistor between the n-side and the p-side a current will start to flow on this external resistor, and we can extract electrical power.
- ▶ If there is no external resistor, the voltage will rise until the so called open circuit voltage, and due to the diffusion potential of the charge carriers a diffusion current will start to flow in the opposite direction to the drift current.





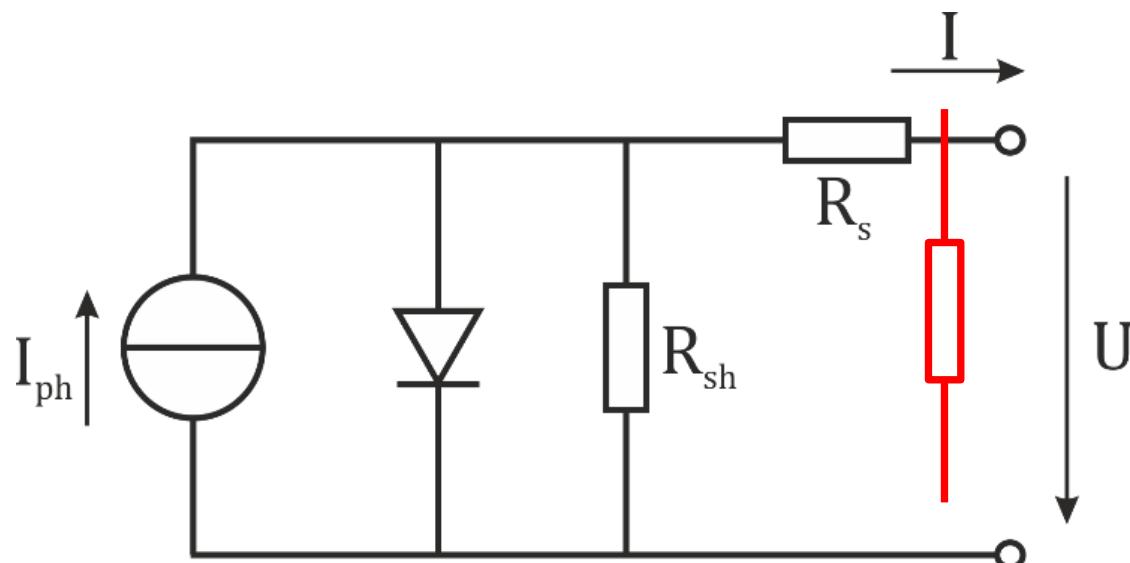
EQUIVALENT CIRCUIT AND I-V CURVE

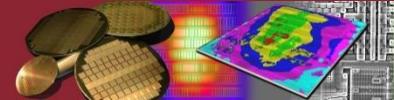




Equivalent circuit of a solar cell

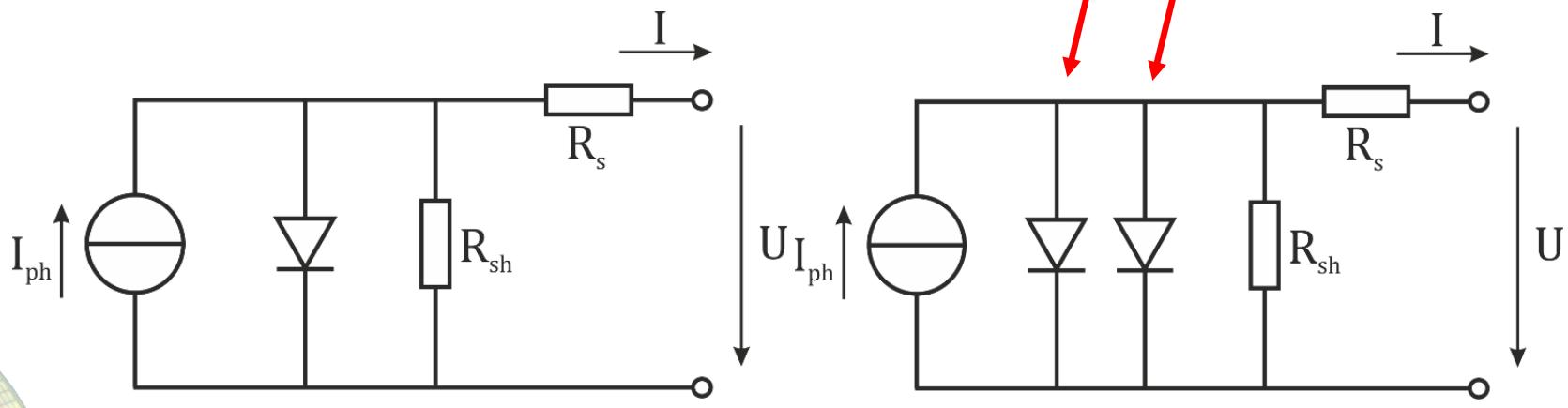
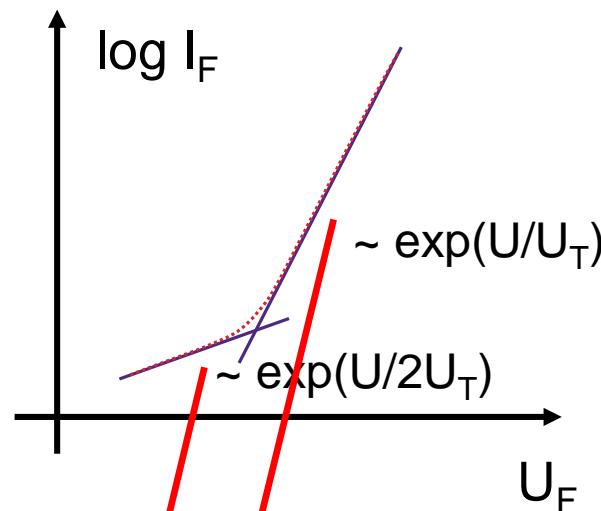
- ▶ Current source: drift current from the photogenerated charge carriers
- ▶ Diode: p-n junction, its current depends on the diffusion potential (forward current = diffusion current)
- ▶ The diffusion potential of the diode is determined by the load resistance and the parasitic resistances

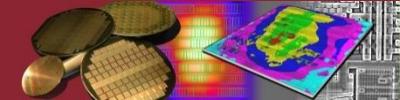




Single diode or two diode model

- ▶ Recombination current can be taken into account with a two diode model
- ▶ It is negligible compared to the photo current





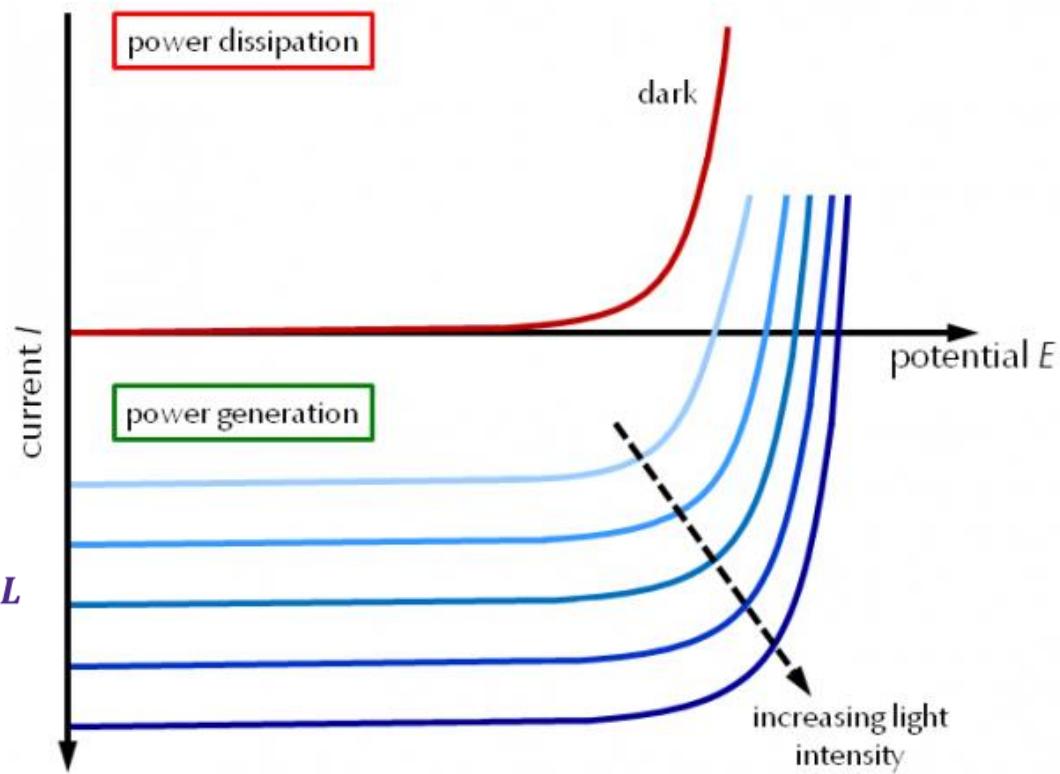
I-V curve of a solar cell

► Ideal solar cell

$$\blacktriangleright I = I_0 \left(e^{\frac{U}{nU_T}} - 1 \right)$$

► Non-ideal solar cell

$$\blacktriangleright I = I_{ph} - I_0 \left(e^{\frac{U+I \cdot R_s}{nU_T}} - 1 \right) - \frac{U+I \cdot R_s}{R_{sh}}$$



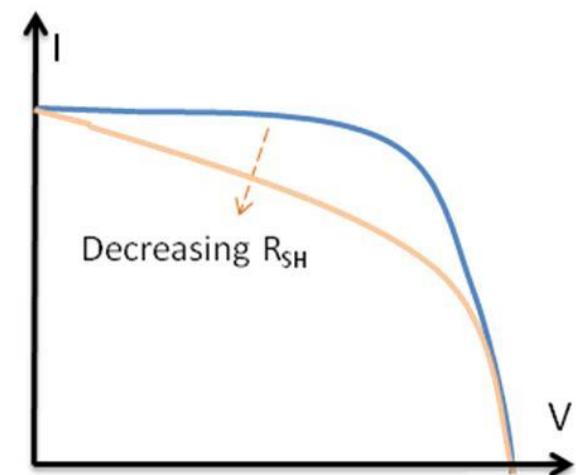
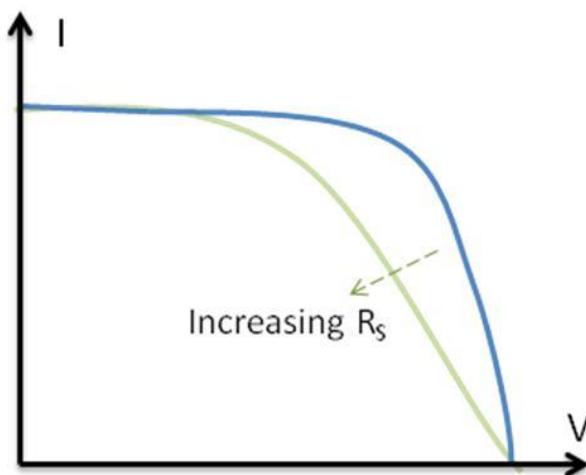
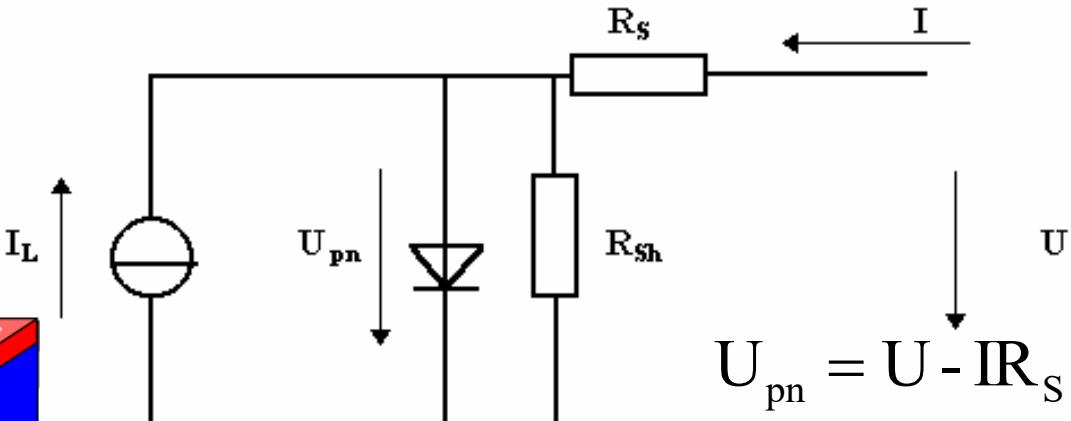
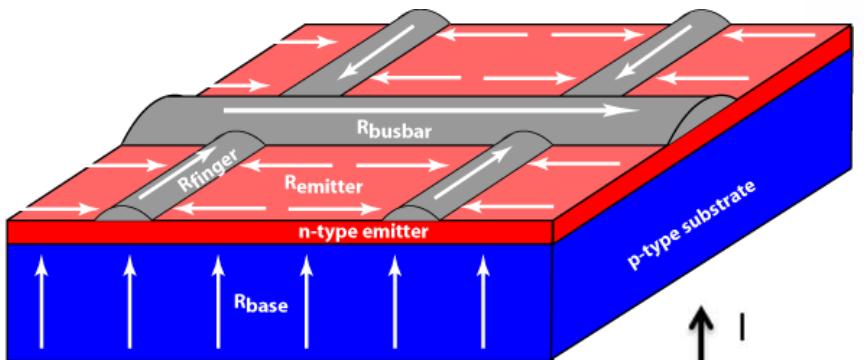
$$I_{ph} = k \cdot G_L$$

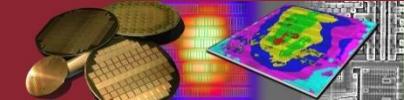




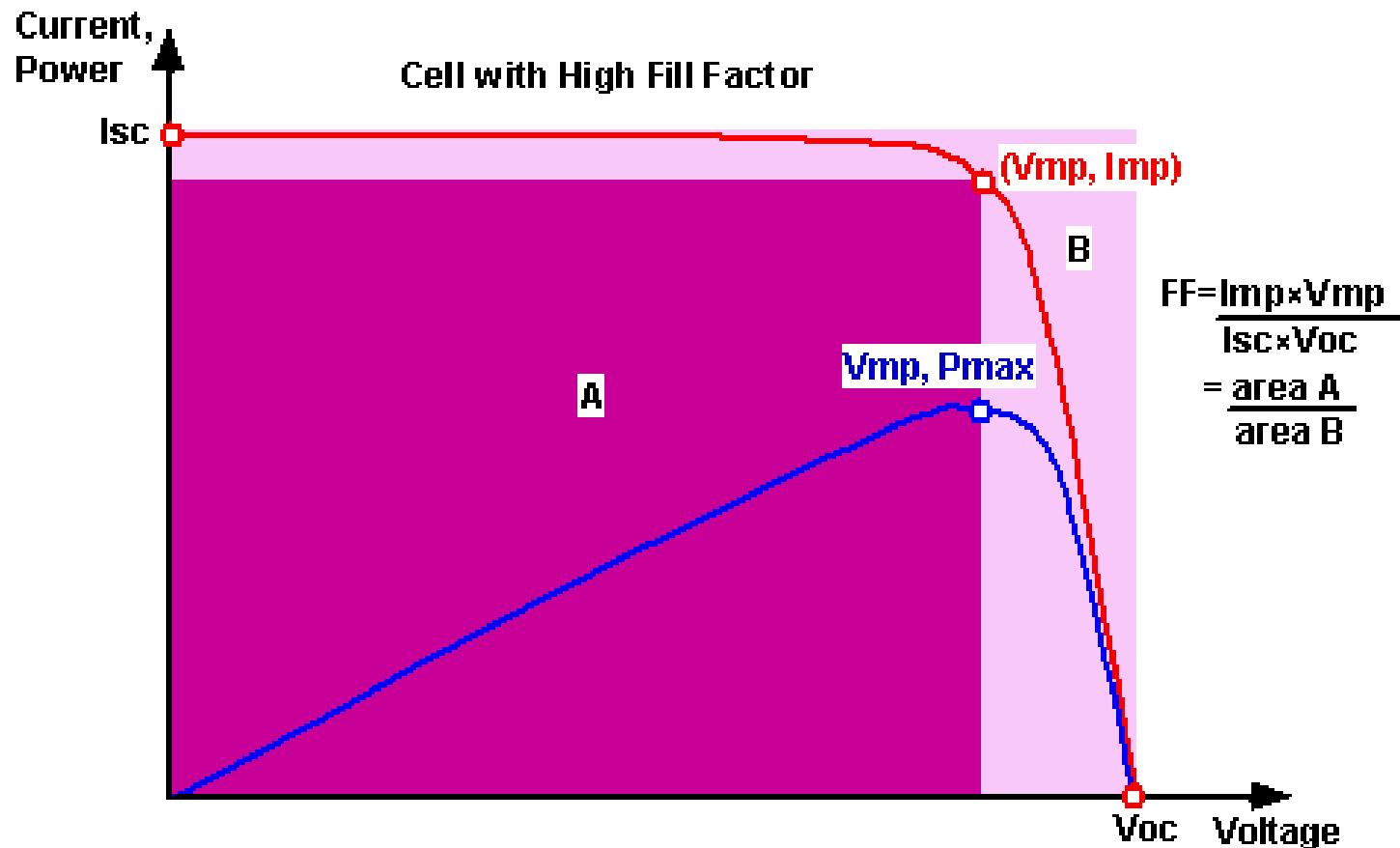
Effect of parasitic resistances

- ▶ R_s – series resistance
- ▶ R_{sh} – shunt resistance
- ▶ I_L – photocurrent





Ideal solar cell characteristics





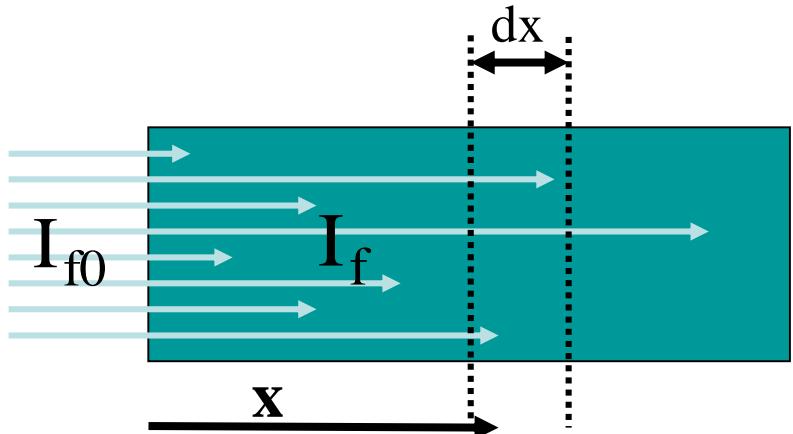
SPECTRAL RESPONSE



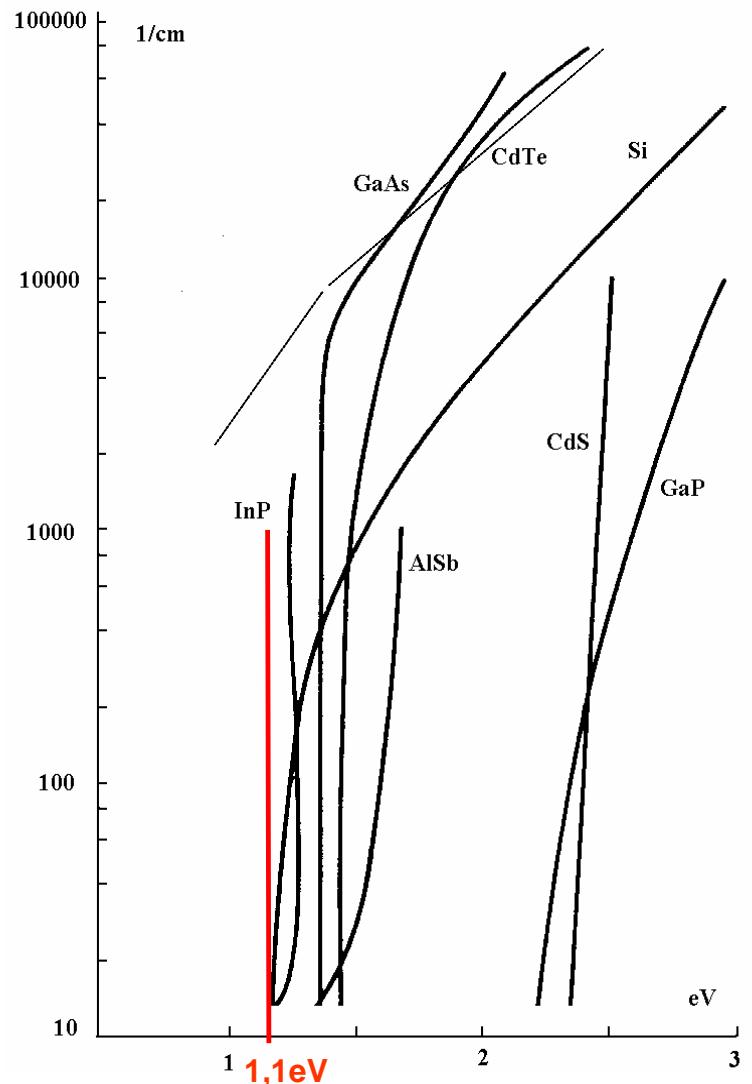


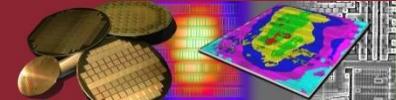
Light and semiconductor interaction

$$dI_f = -\alpha \cdot I_f \cdot dx$$

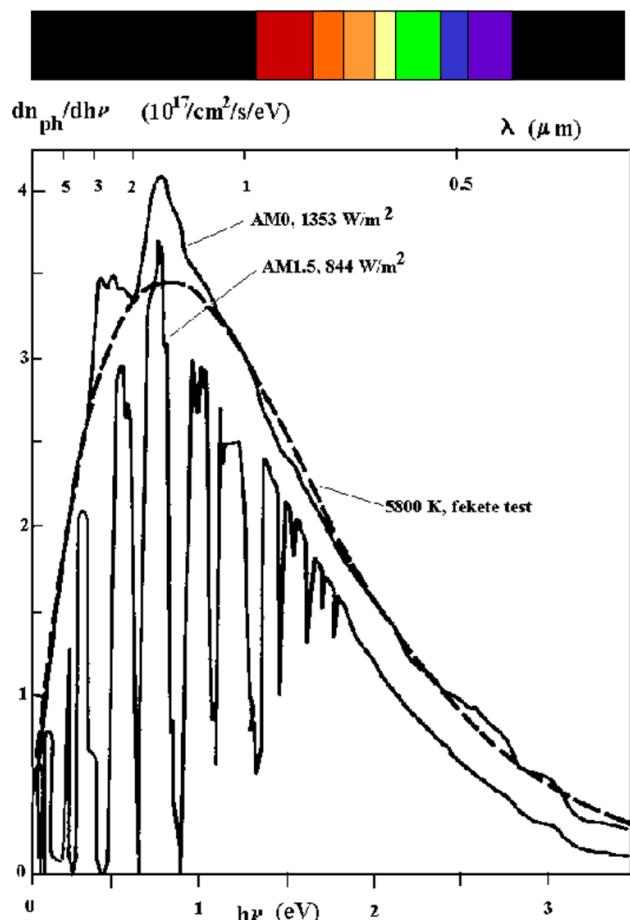


$$I_f = I_{f0} \exp(-\alpha \cdot x)$$

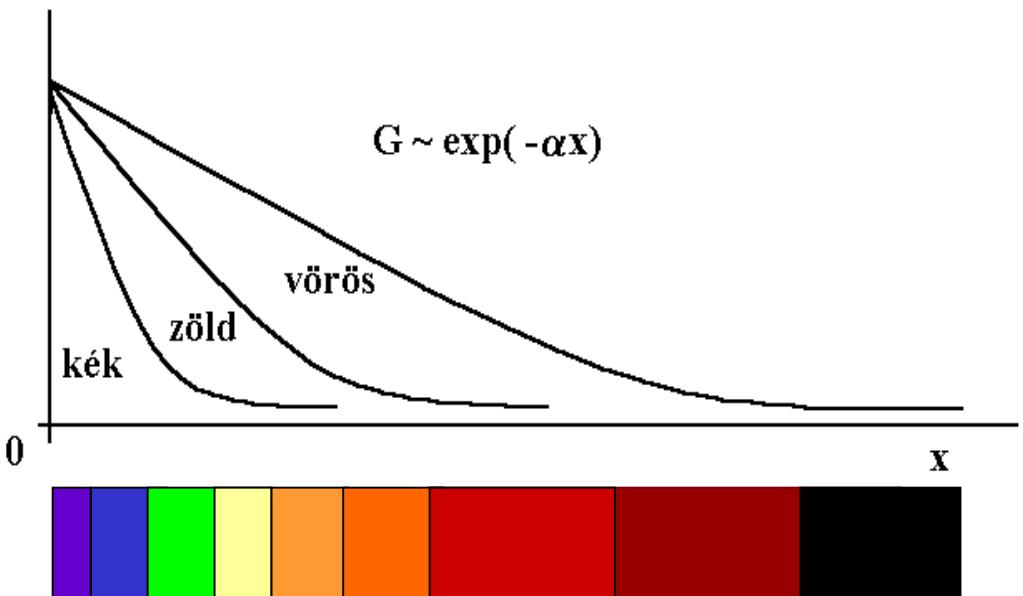




Generation rate



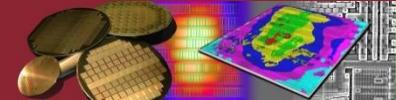
$G(\lambda;x)$



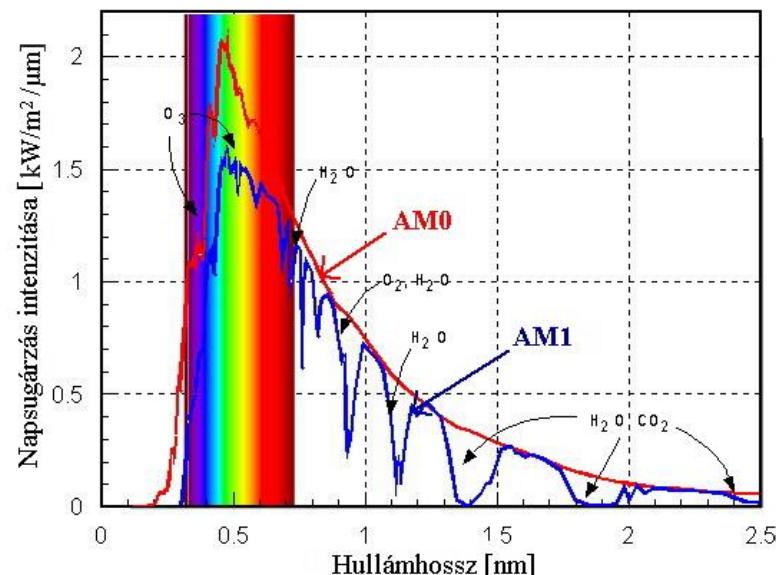
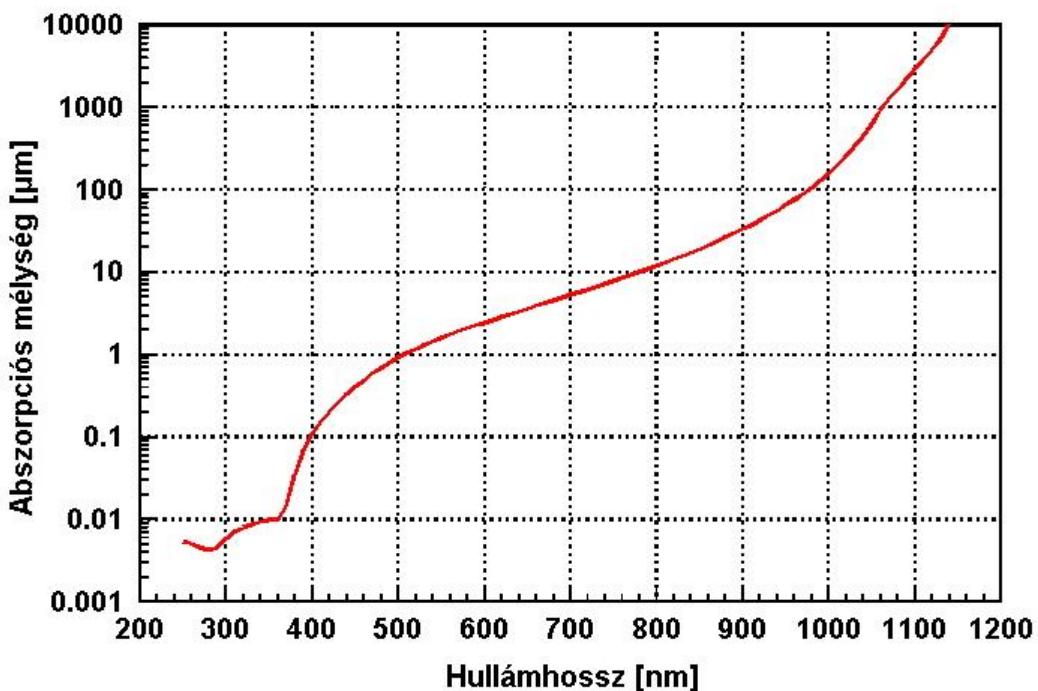
Calculation of the generation rate:

$$G(\lambda,x) = a(\lambda) \cdot F(\lambda) \cdot [1 - R(\lambda)] \cdot \exp(-\alpha(\lambda) \cdot x)$$



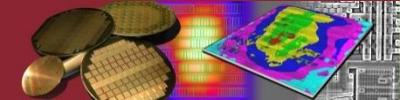


Absorption depth



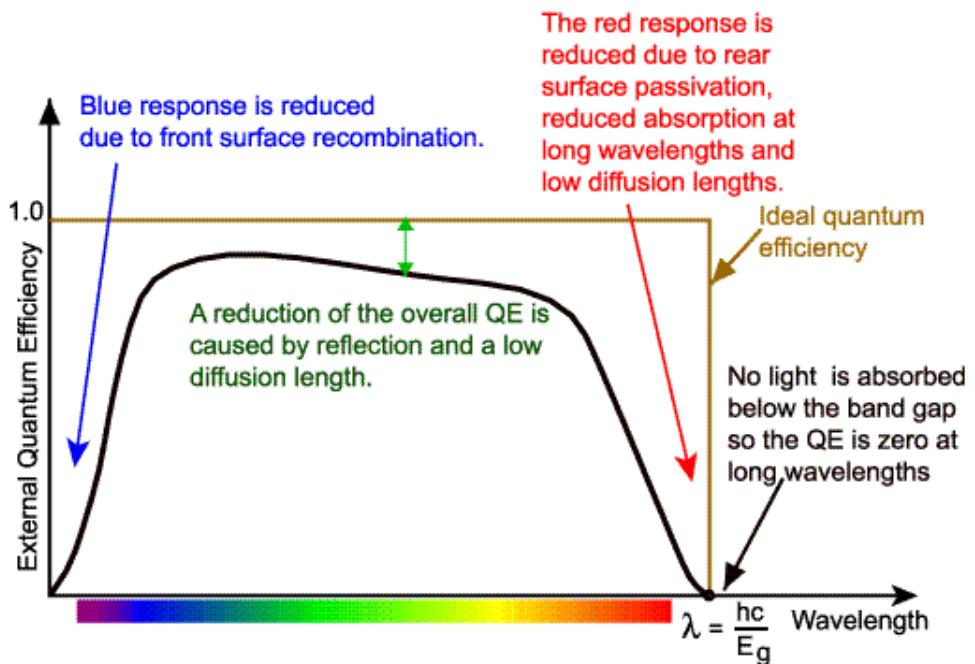
- ▶ UV and blue light have a high absorption coefficient → absorbed at the surface
- ▶ NIR wavelentghs have a lower absorption coefficient → absorbed in the depth of the material





Quantum efficiency

- ▶ The ratio of the number of extracted electrons and the number of irradiated photons
- ▶ Due to reflexion there is a difference between external (EQE) and internal (IQE) quantum efficiency
- ▶ The sum of all the extracted electrons is the short circuit current that almost equals the photocurrent
- ▶ It is complicated to measure the number of photons



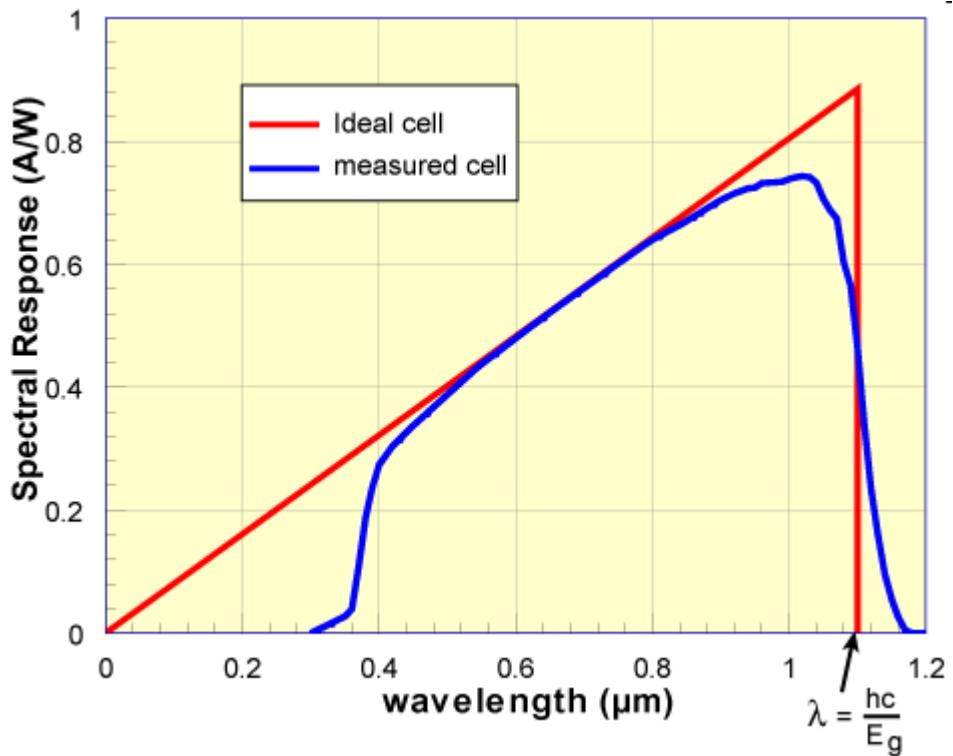
$$I_L = q \int_{(\lambda)} F(\lambda) \cdot [1 - R(\lambda)] \cdot IQE(\lambda) d\lambda$$





Spectral response

- ▶ A generált fotoáram és az adott hullámhosszú fény teljesítményének hányadosa, adott felületen
- ▶ A napelem rétegszerkezetének vizsgálatára alkalmas
- ▶ Technológia ismeretében az esetleges hibák kideríthetők (rossz felületi passziváls, BSF réteg)



$$SR = \frac{J_z}{P_{\text{fény}}} = \frac{I_z}{A \cdot P_{\text{fény}}}$$

$$SR(\lambda) = \frac{q \cdot \lambda}{h \cdot c} \cdot QE(\lambda) = 0,808 \cdot \lambda \cdot QE(\lambda)$$



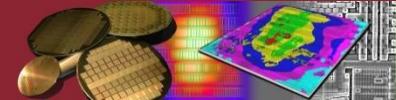


Spectral response – what is it for?

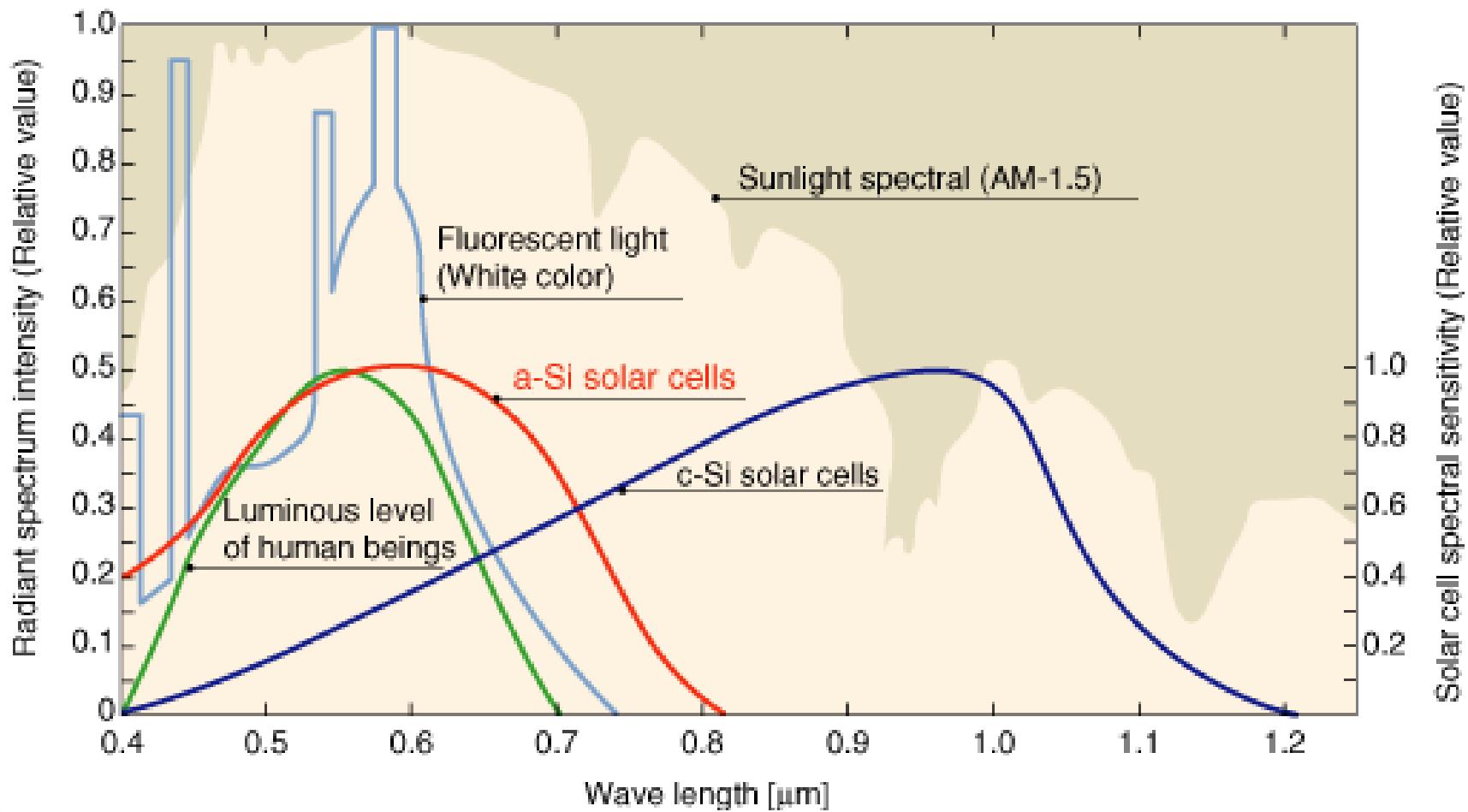
1. It is a „footprint” of the technology! (It shows the nature and location of the problems.)
2. If the spectral response is known, than the response for a given light source can be calculated:

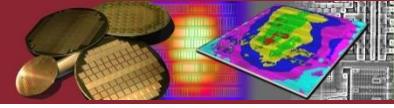
$$J_L = q \int_0^{\lambda_m} F(\lambda) [1 - R(\lambda)] SR(\lambda) d\lambda$$





Spectral response





EFFICIENCY LIMITS AND OPERATIONAL CONDITIONS



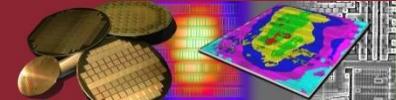


Loss factors

- ▶ Photons with lower energy than the band gap
- ▶ Above bandgap photon energy
- ▶ Broad solar spectrum
- ▶ Voltage reduction: due to its structure a solar cell can not produce an open circuit voltage of 1,12 V as its band gap would suggest, but only ca. 0,6-0,7 V
- ▶ Fill Factor: the I-V curve is not „square”, thus we have losses due to the exponential characteristic of the p-n junction: the diode is partially open at the maximum power point.



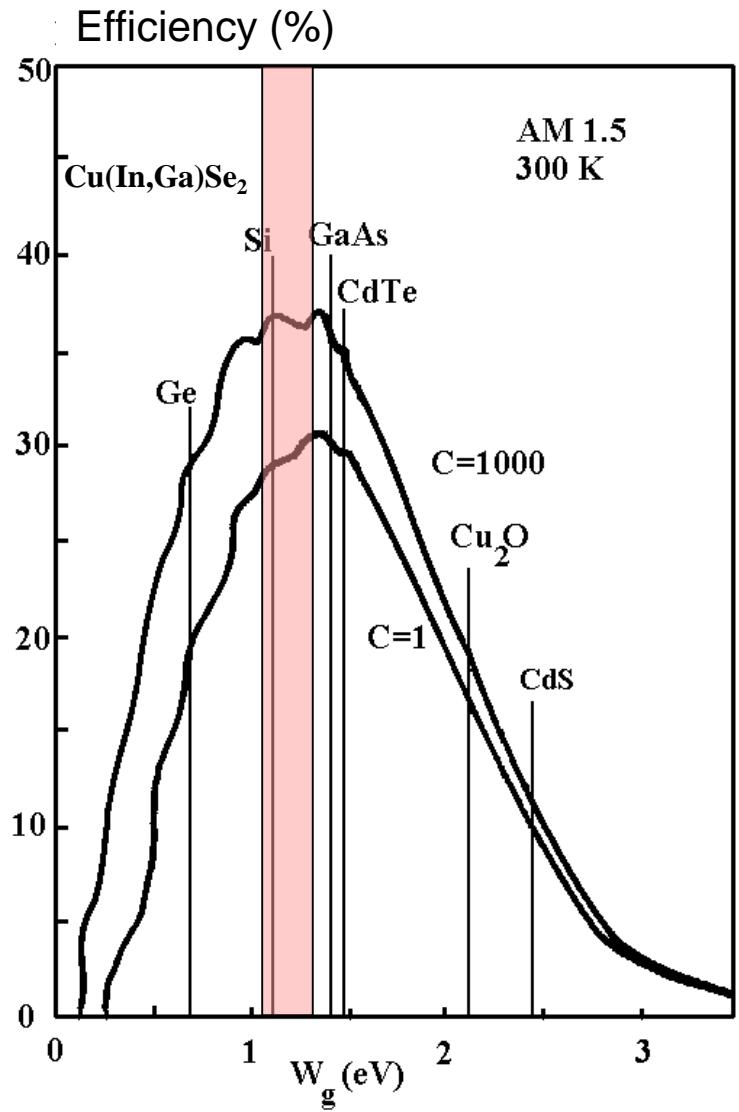
$\eta_{max} = 29\% \text{ (for silicon, AM 1,5 solar spectrum)}$

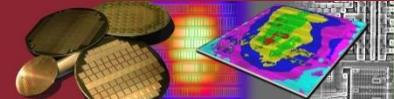


Choosing the semiconductor material

Optimal material

- ▶ AM1.5 Earth conditions
- ▶ Without concentration,
- ▶ 1000x concentration
- ▶ Theoretical limit for the conversion efficiency as function of the energy gap (Shockley-Queisser Limit)





Effect of illumination

Open-circuit condition ($I=0$)

$$U_0 = U_T \ln \left(1 + \frac{I_L}{I_S} \right) \approx U_T \ln \frac{I_L}{I_S} = U_T \ln \frac{kI_{f0}}{I_S}$$

10x-higher intensity:

$$\Delta U = U_T \ln 10 = 60 \text{ mV}$$

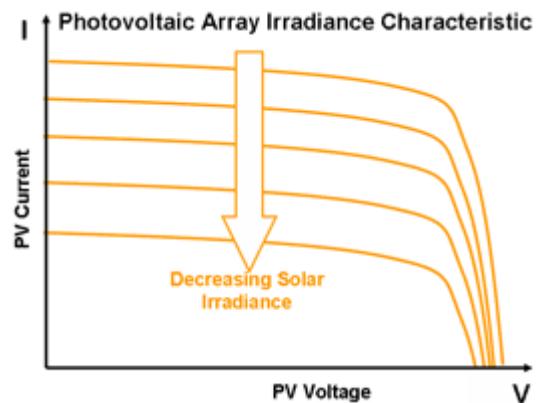
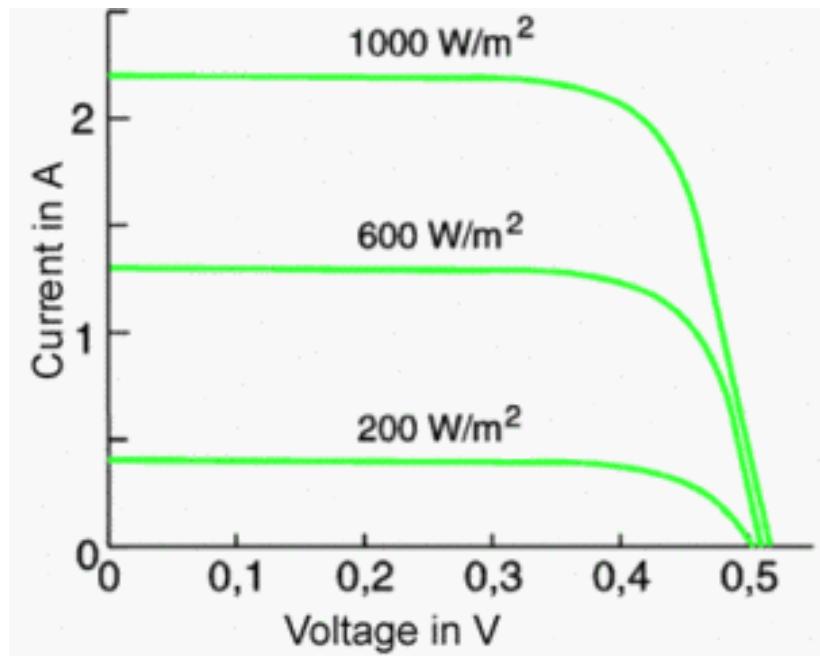
Short-circuit condition ($U=0$)

$$I_{sc} = -I_L = kI_{f0}$$

10x- higher intensity: $I_{sc,10} = 10 \cdot I_{sc}$

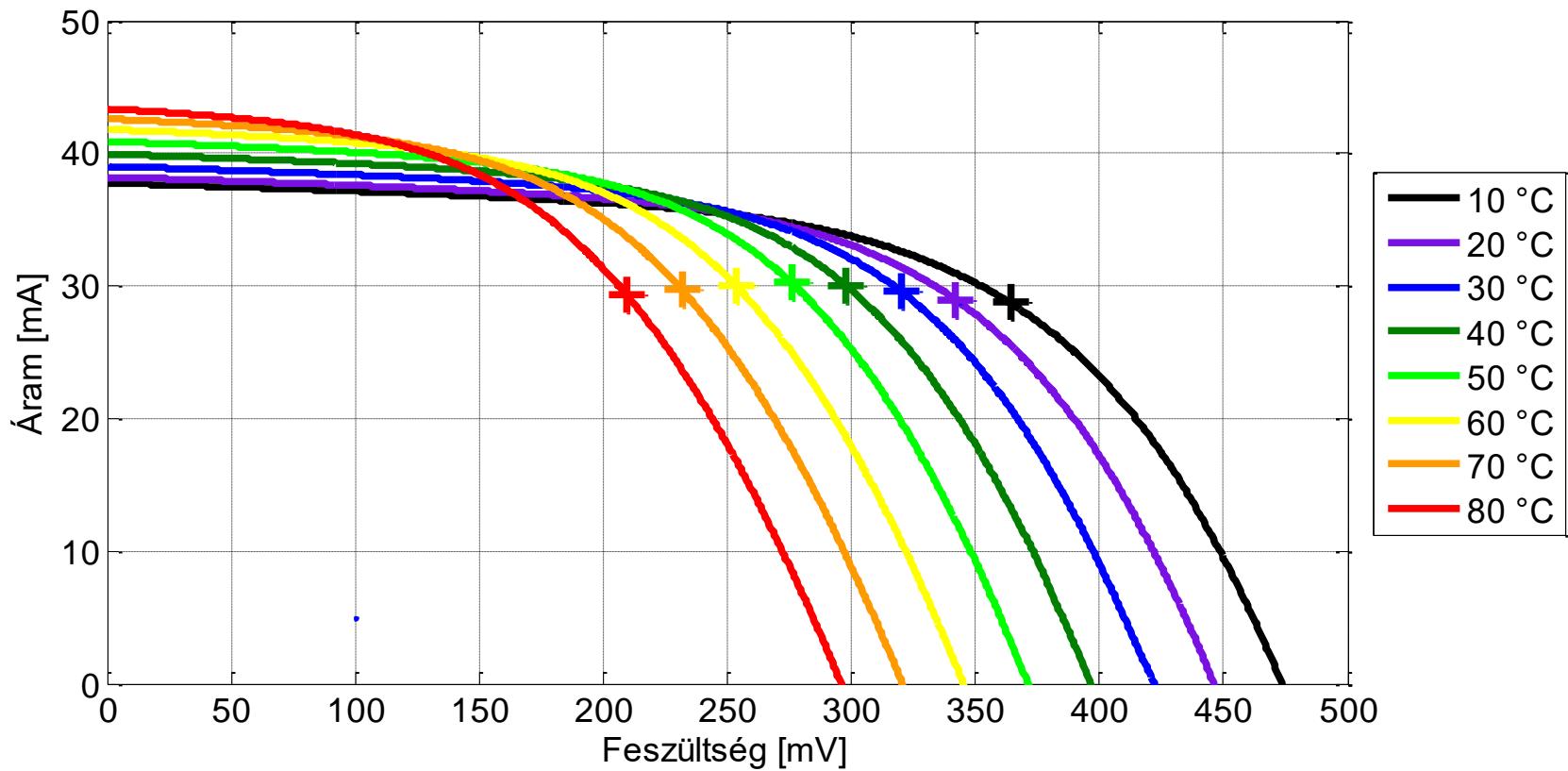


I_{f0} – photon flux (incoming photons/ sec)





Effect of temperature



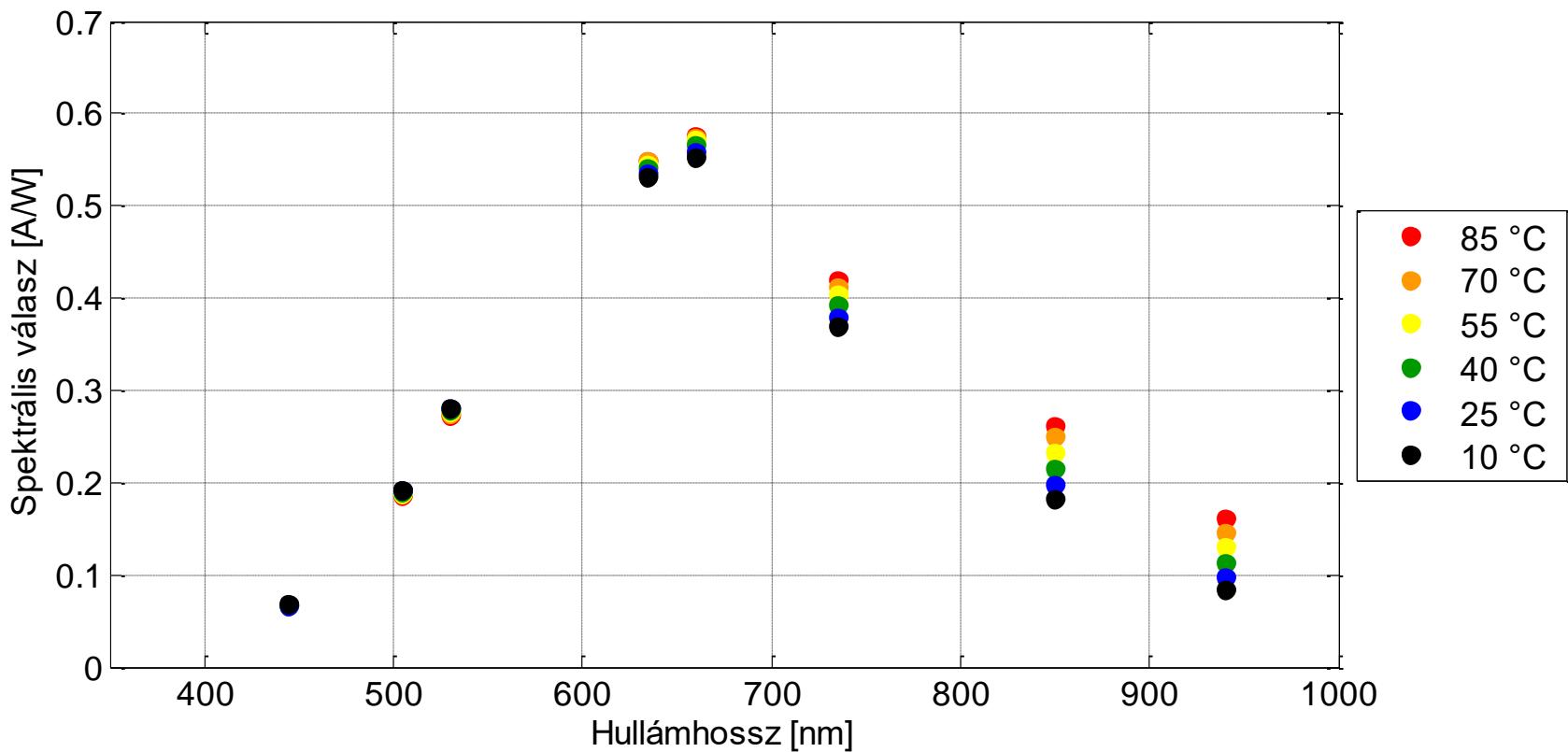
$$\frac{dI_{SC}}{dT} \approx 0,1..0,2 \%/\text{°C}$$

$$\frac{dU_{OC}}{dT} \approx 2 \text{ mV}/\text{°C}$$





Effect of temperature on the spectral response



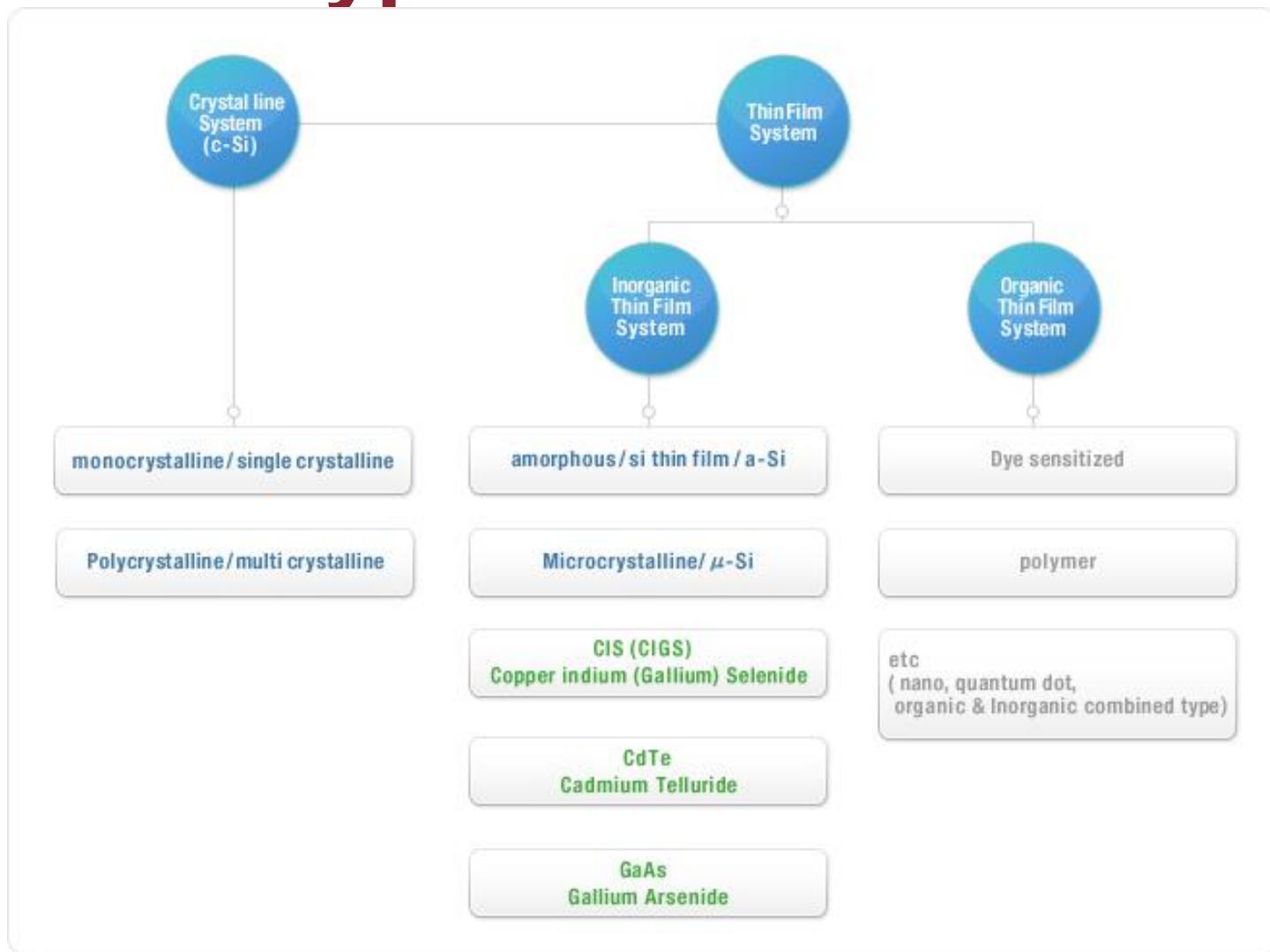


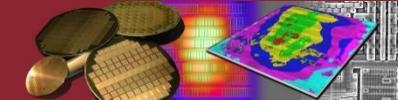
TYPES OF SOLAR CELLS





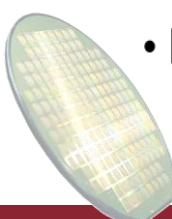
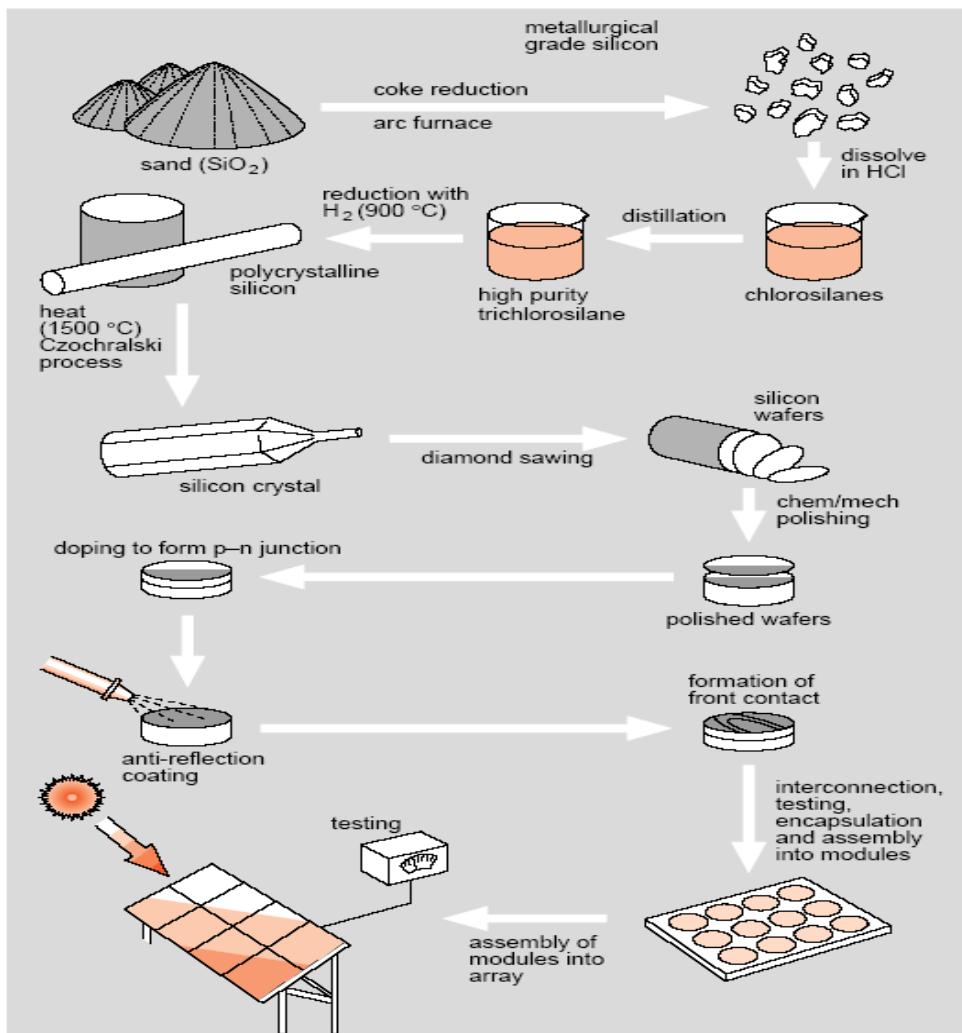
Solar cell types

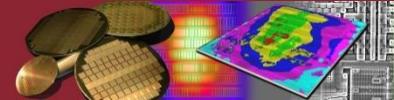




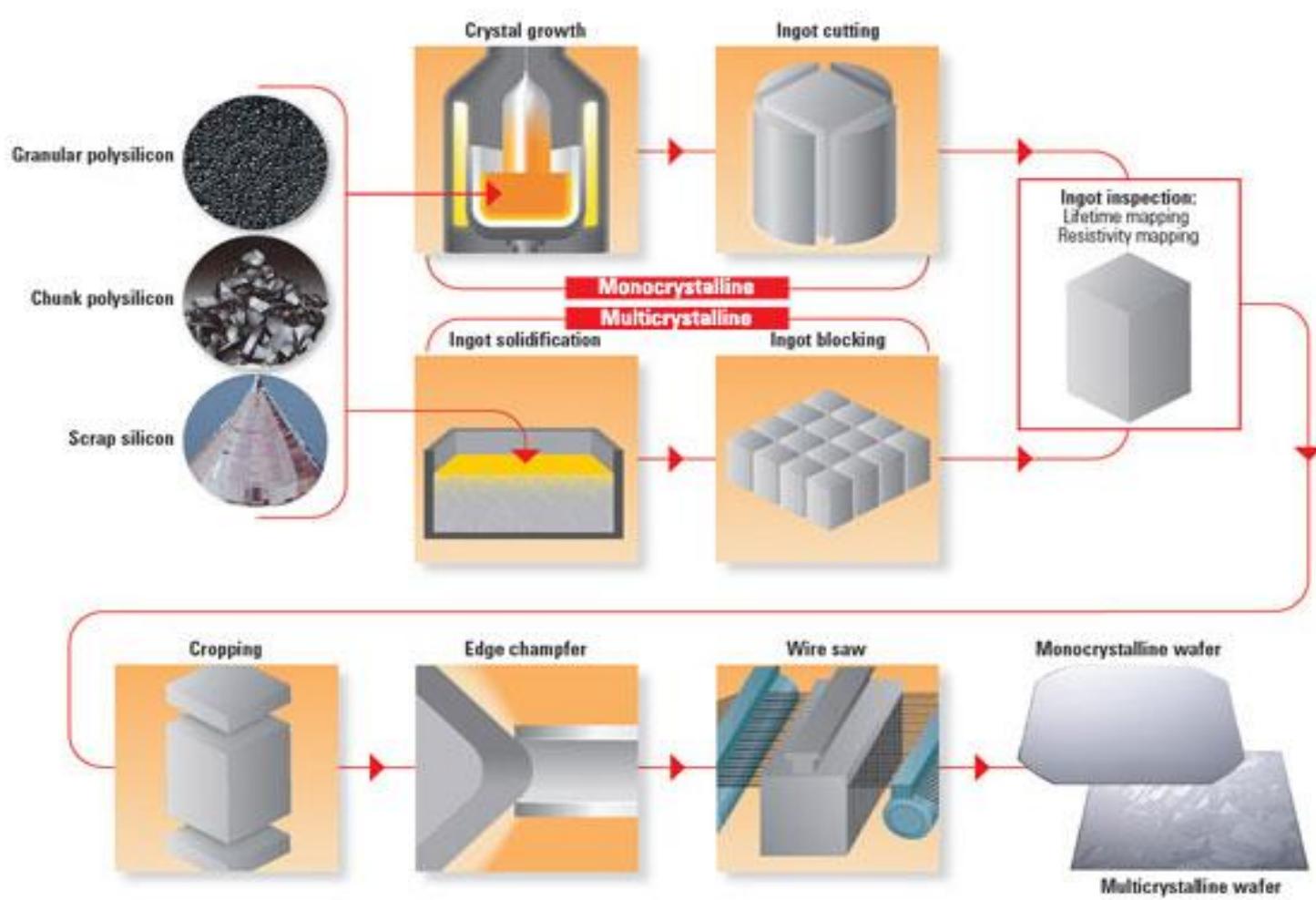
Process steps of monocrystalline solar cells

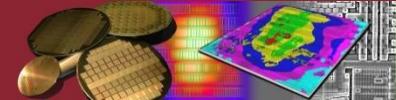
- Sand (SiO_2)
- Metallurgic Si
- Chlorsilanes (for example SiHCl_3)
- High purity chlorsilan
- Polycrystalline Si bulk
- Single crystal growth
- Slicing of the bulk (diamond saw)
- Ploishing of the surface
- Adalékolás
- ARC layer
- Contact layer
- Modul assembly and testing



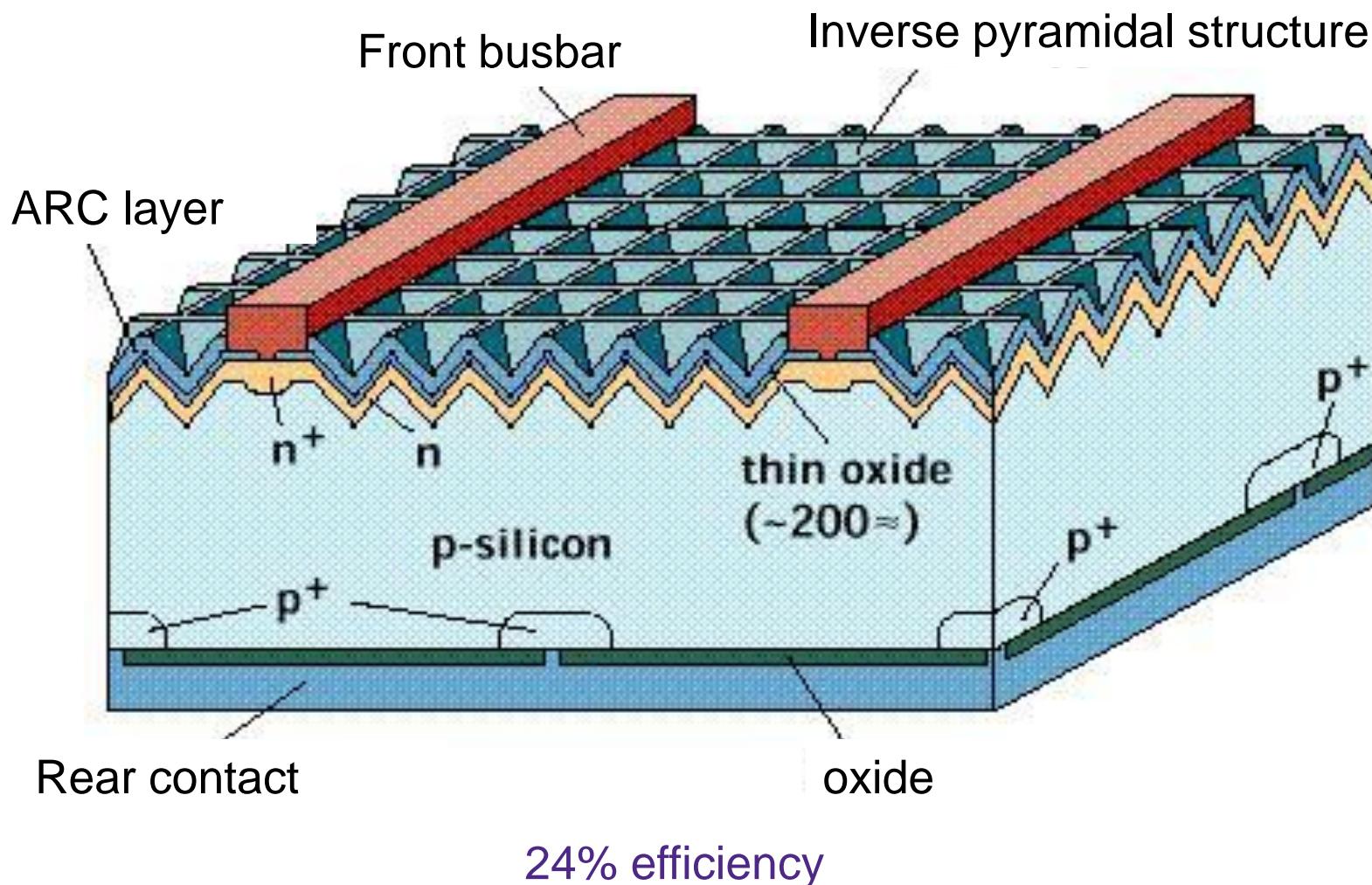


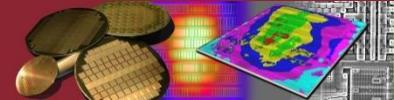
Crystalline wafer production



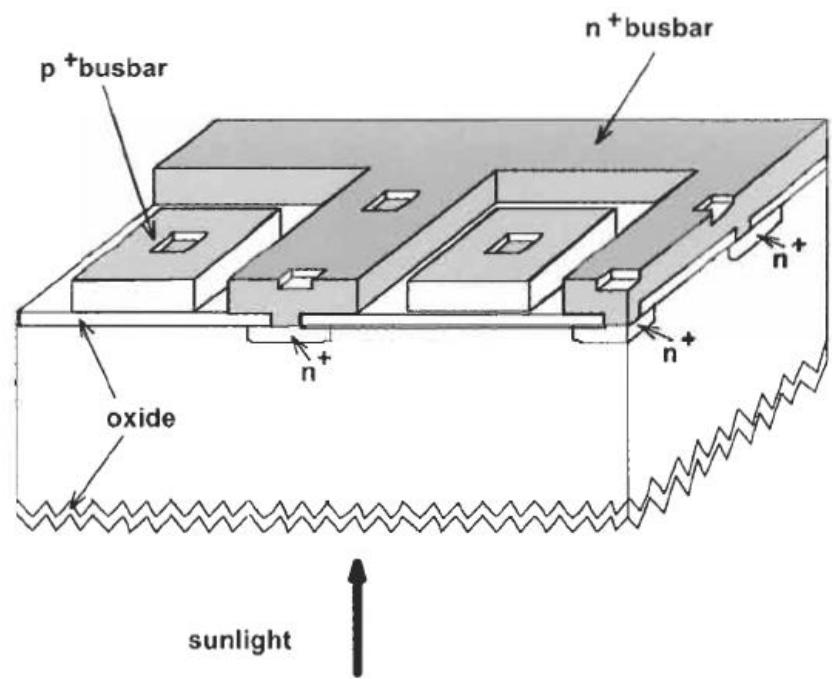
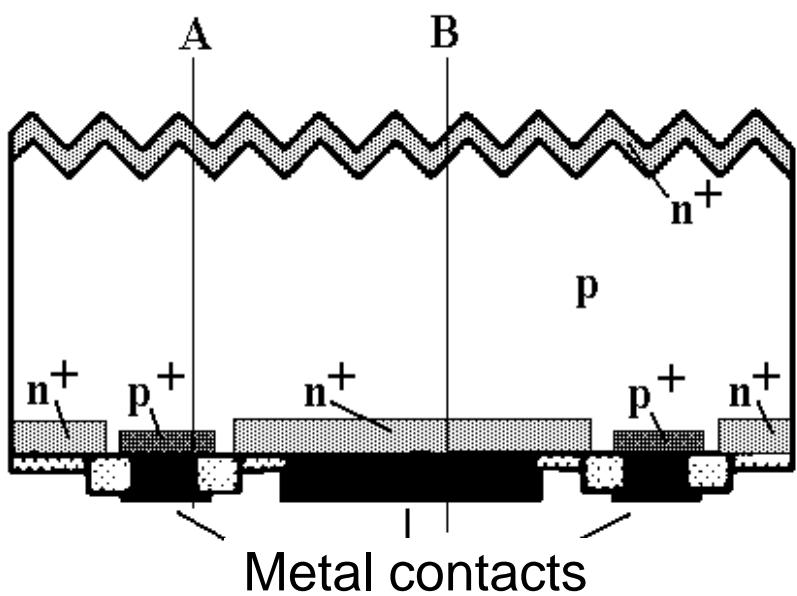


PEARL (passivated emitter and rear locally) cell



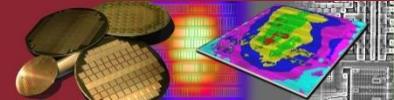


Interdigitated Back contact cells

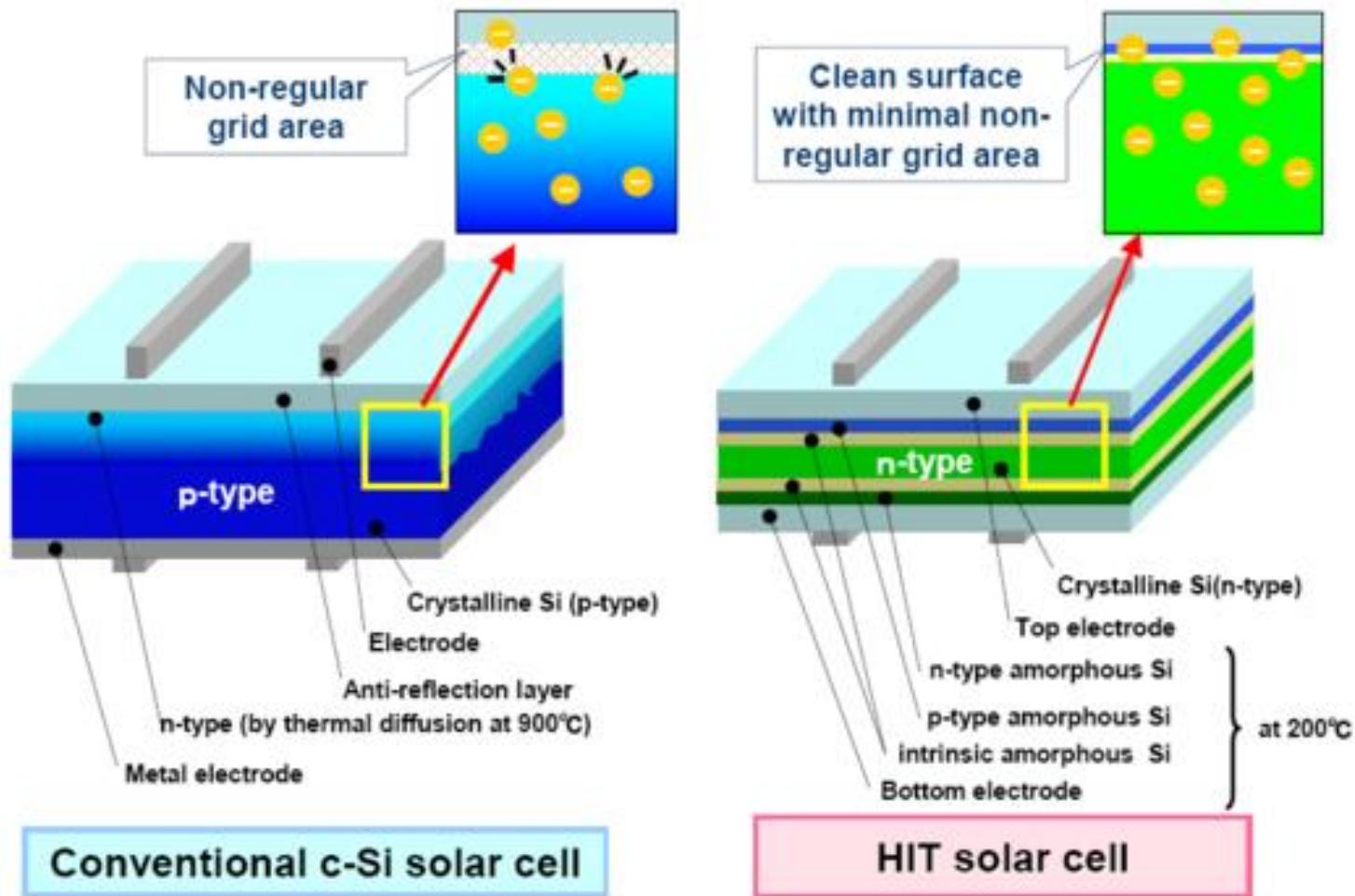


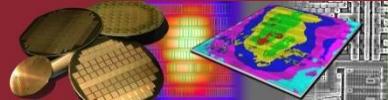
24% efficiency



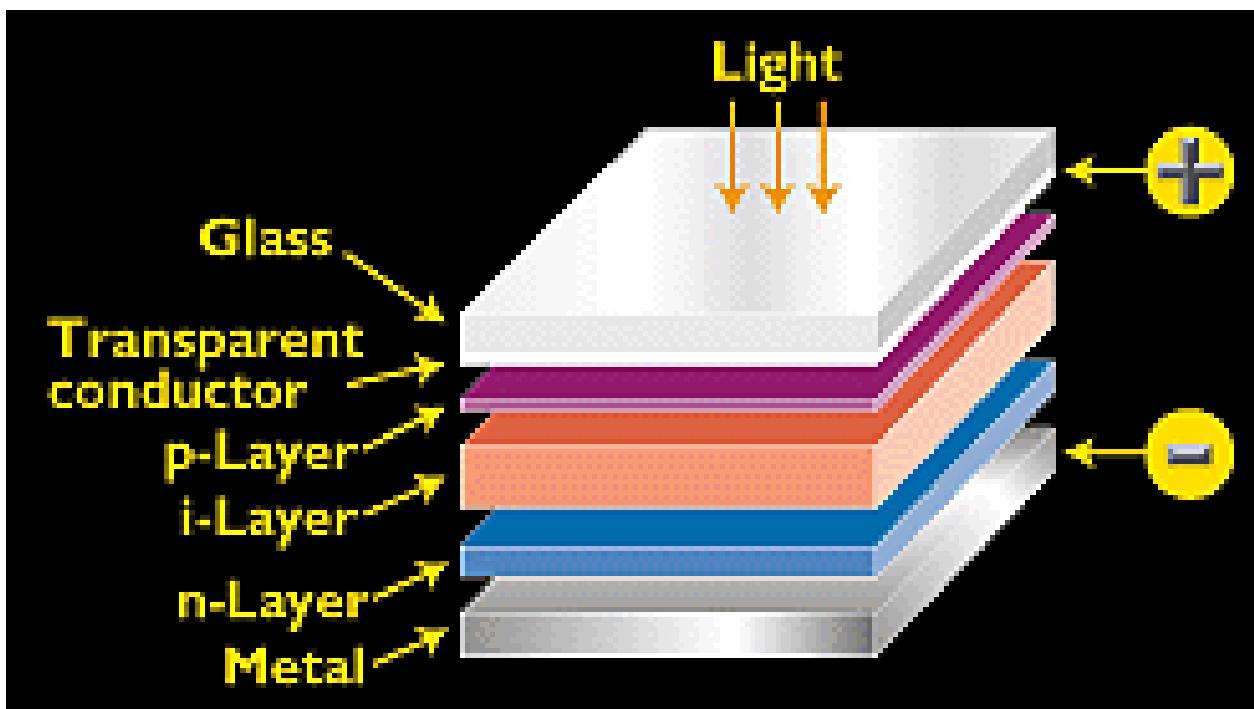


HIT cella (Sanyo)



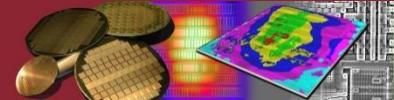


Amorphous Si solar cell (a-Si)

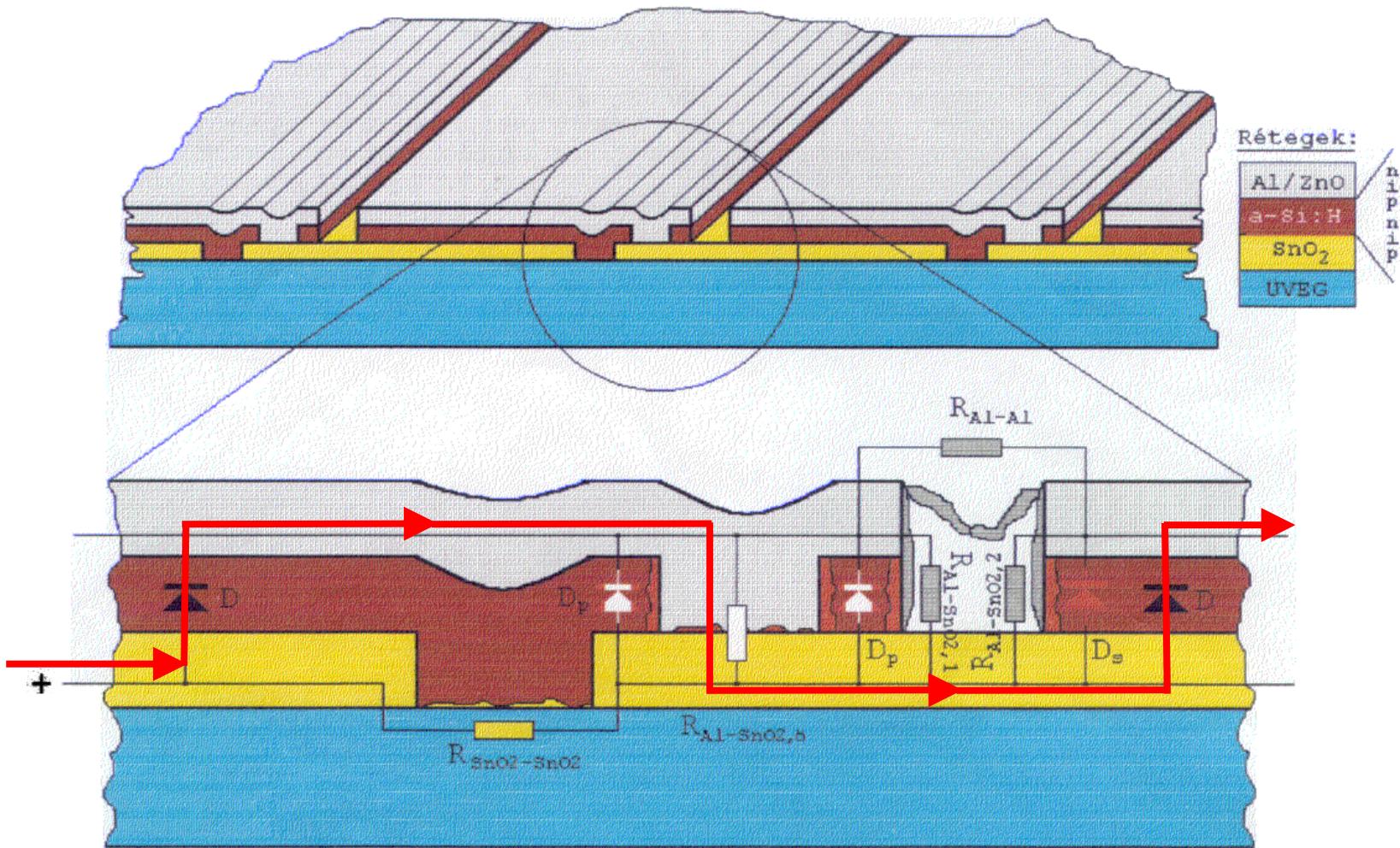


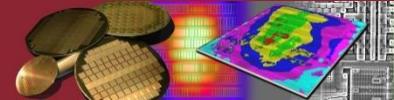
- ▶ ultra thin ($0,008 \mu\text{m}$) p^+ layer
- ▶ intrinsic layer ($0,5 - 1 \mu\text{m}$)
- ▶ thin ($0,02 \mu\text{m}$) n^+ layer



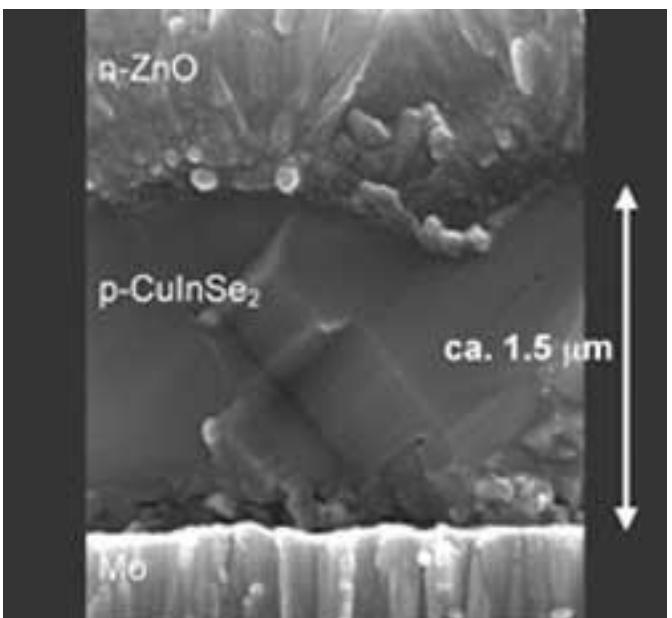


Amorphous Si solar cell (a-Si)





CIS (copper indium diselenid)

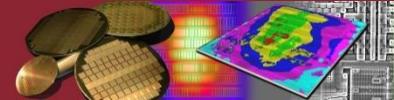


Cella cross-sectional view

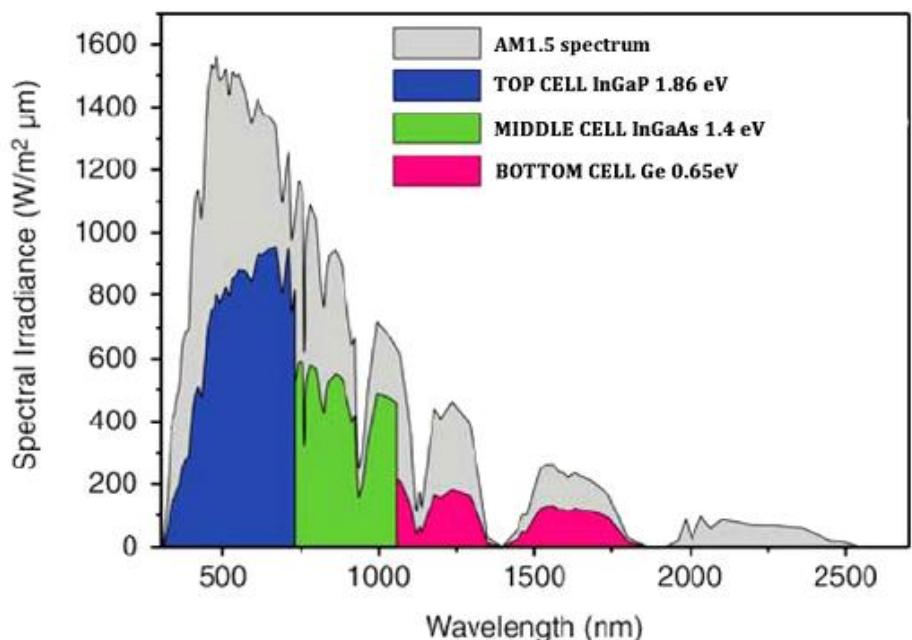
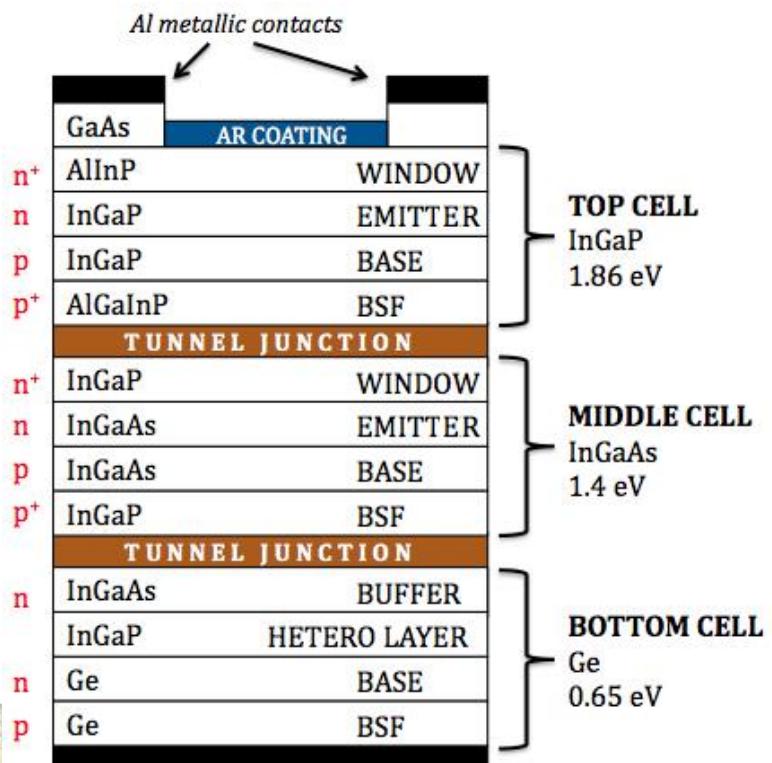
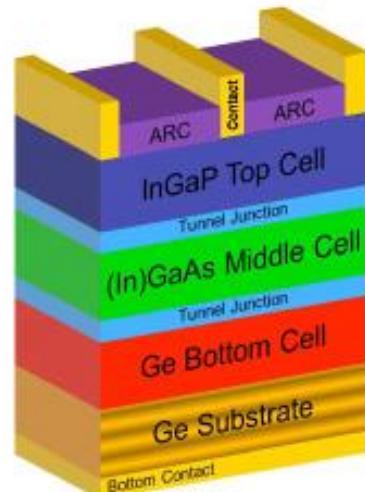


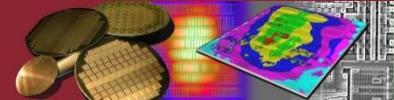
CIS stucture



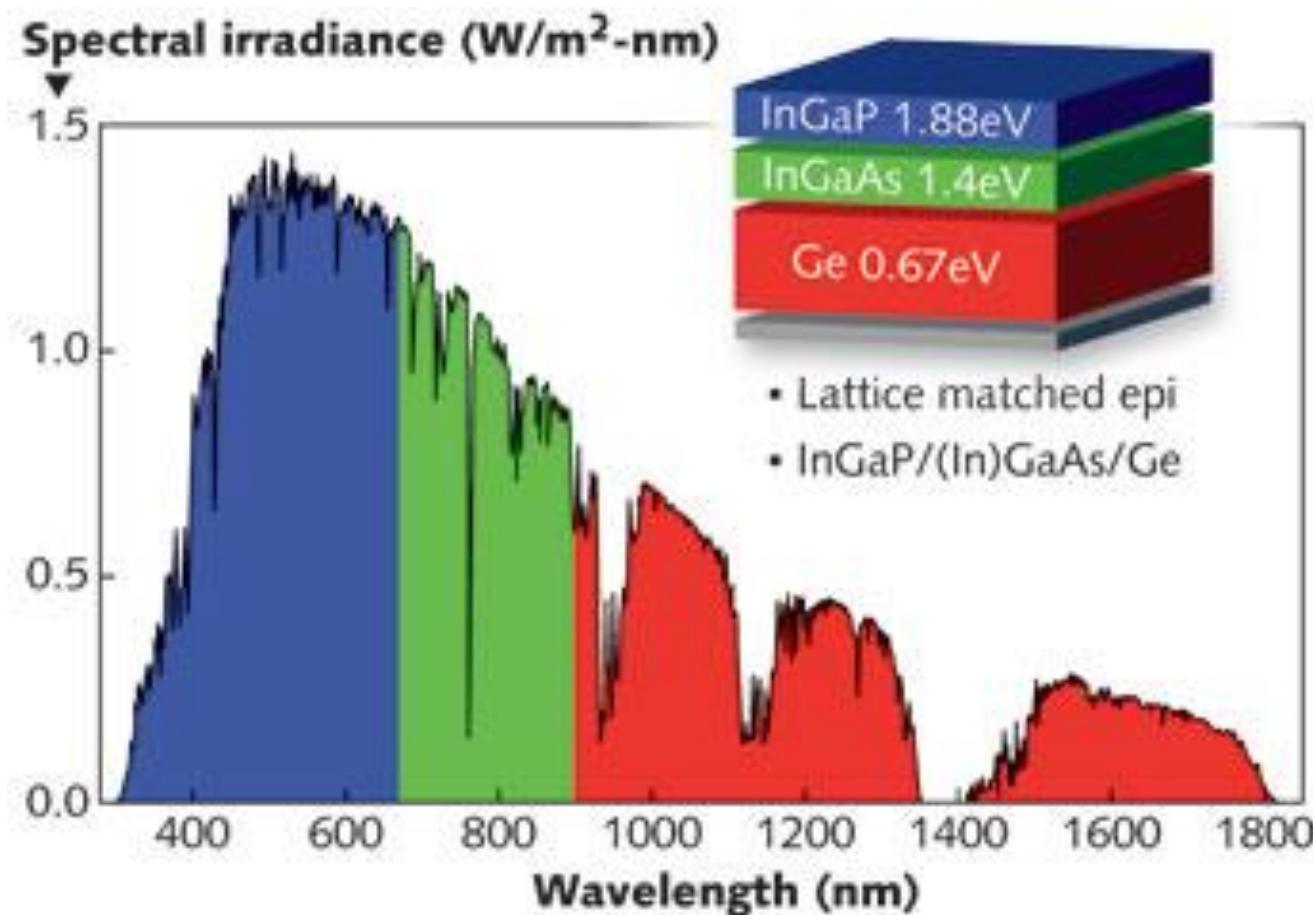


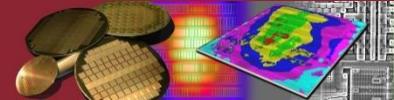
Multijunction solar cells





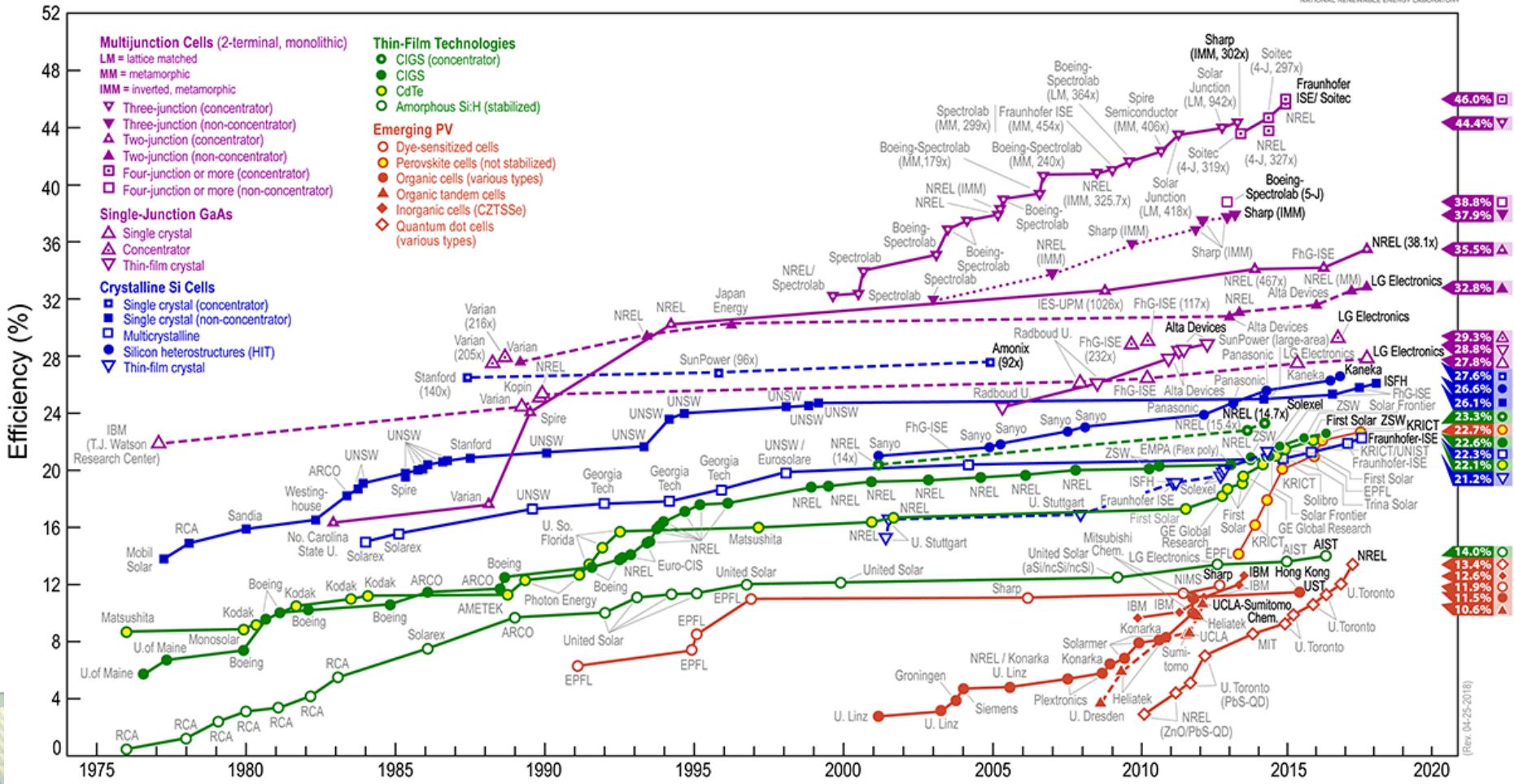
Multijunction solar cells





NREL efficiency chart

Best Research-Cell Efficiencies

NREL
 NATIONAL RENEWABLE ENERGY LABORATORY


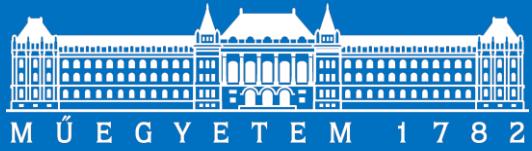


If you want to know more...

► Course: Solar Cells and Renewable Energy sources

- 4 credits
- 2 lectures (90 min) a week
- BME VIEEA99
- <https://portal.vik.bme.hu/kepzes/targyak/VIEEA99/en/>





Budapest University of Technology and Economics
Department of Electron Devices

Microelectronics, BSc course

Thermal laboratory

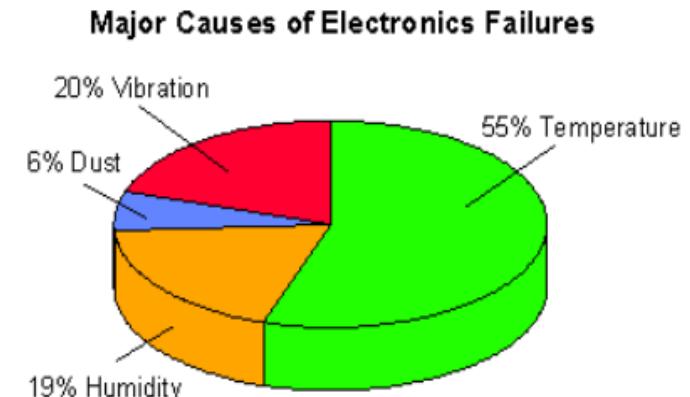
Temperature (C)

35.5 45.7 55.9 66.1

Dr. Bognár György, Hantos Gusztáv, Dr. Szabó Péter

Thermal problems? Why should we care?

- Higher performance requires devices with higher dissipation which calls for cooling equipments with higher efficiency
- What happens if the temperature increased?

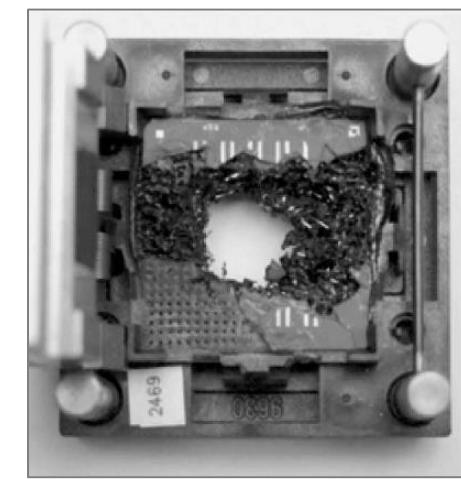
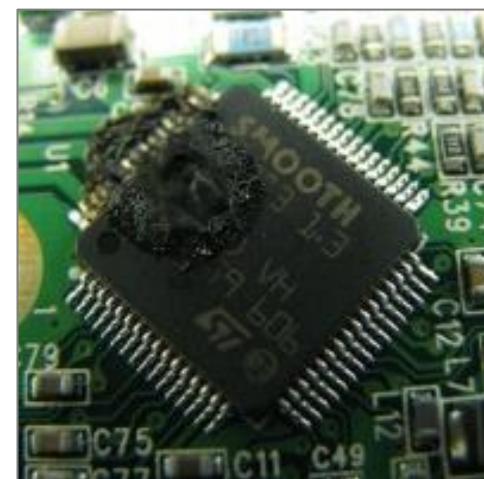
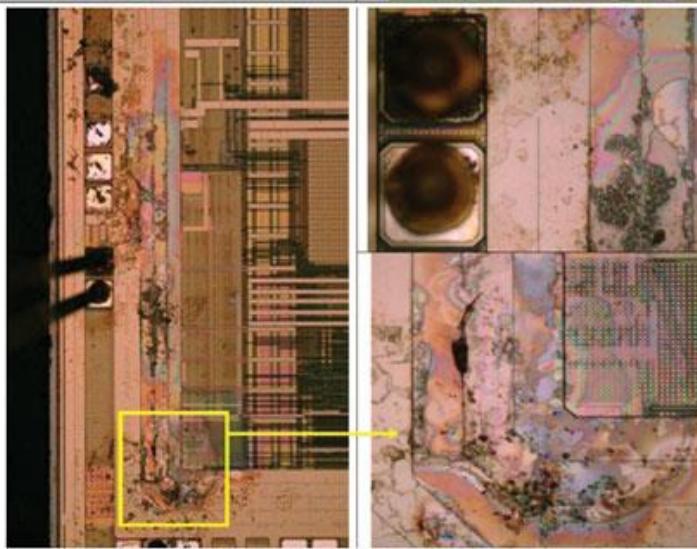
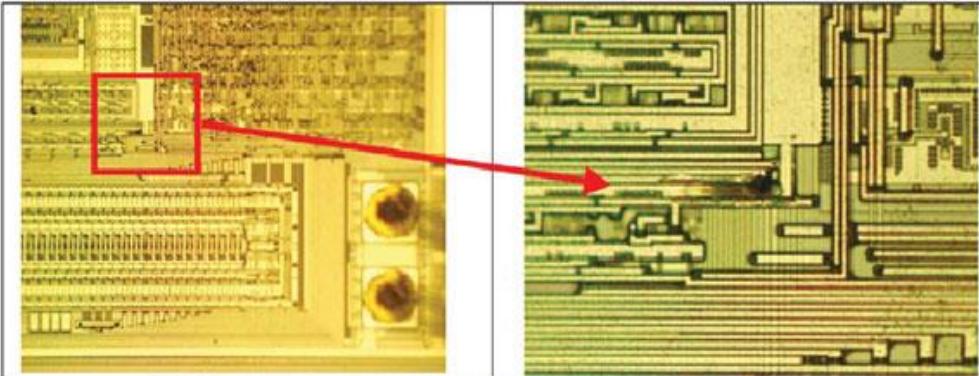


(Source : US Air Force Avionics Integrity Program)



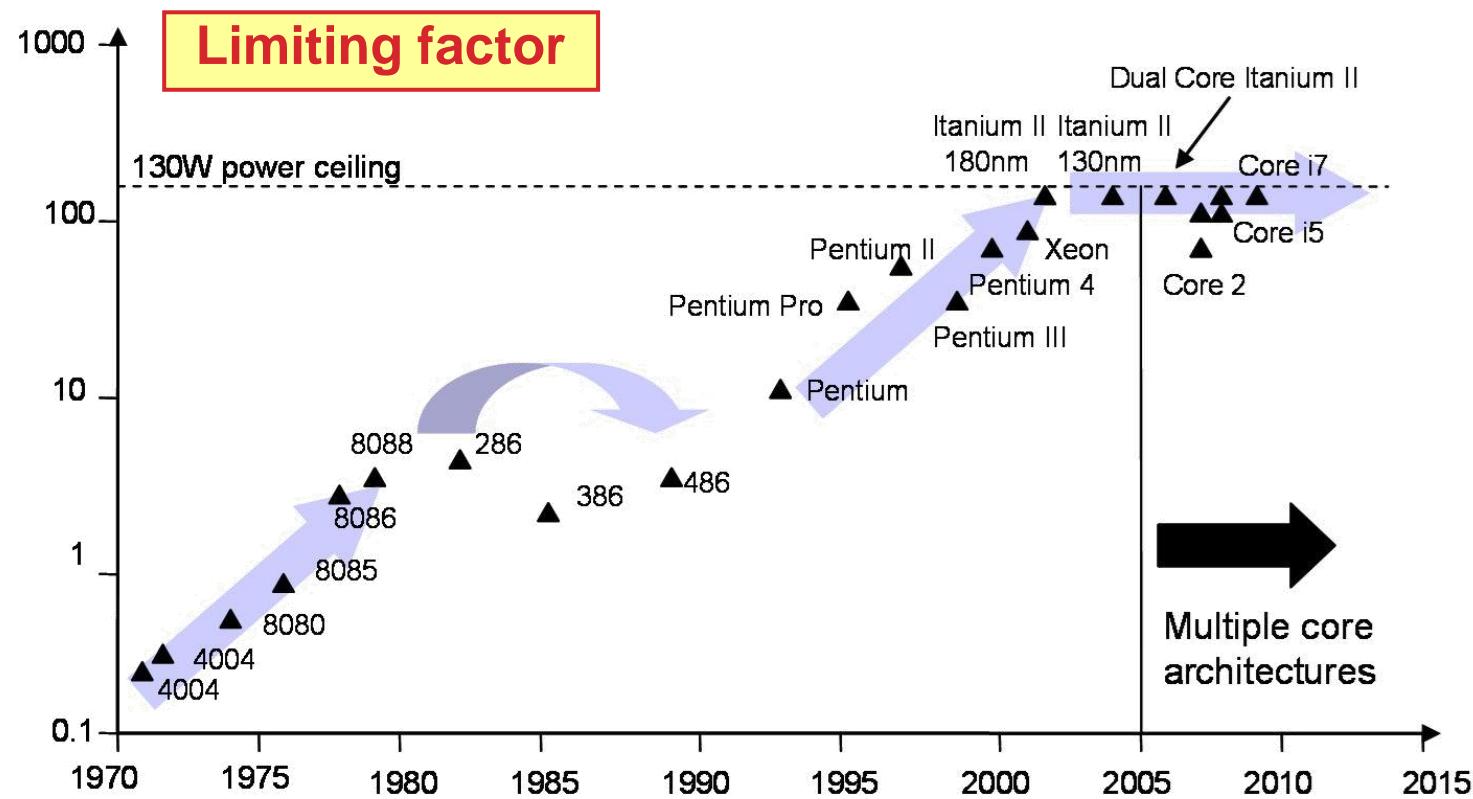
Thermal problems? Why should we care?

- Increased dissipation, thermal runaway



Power consumption trend

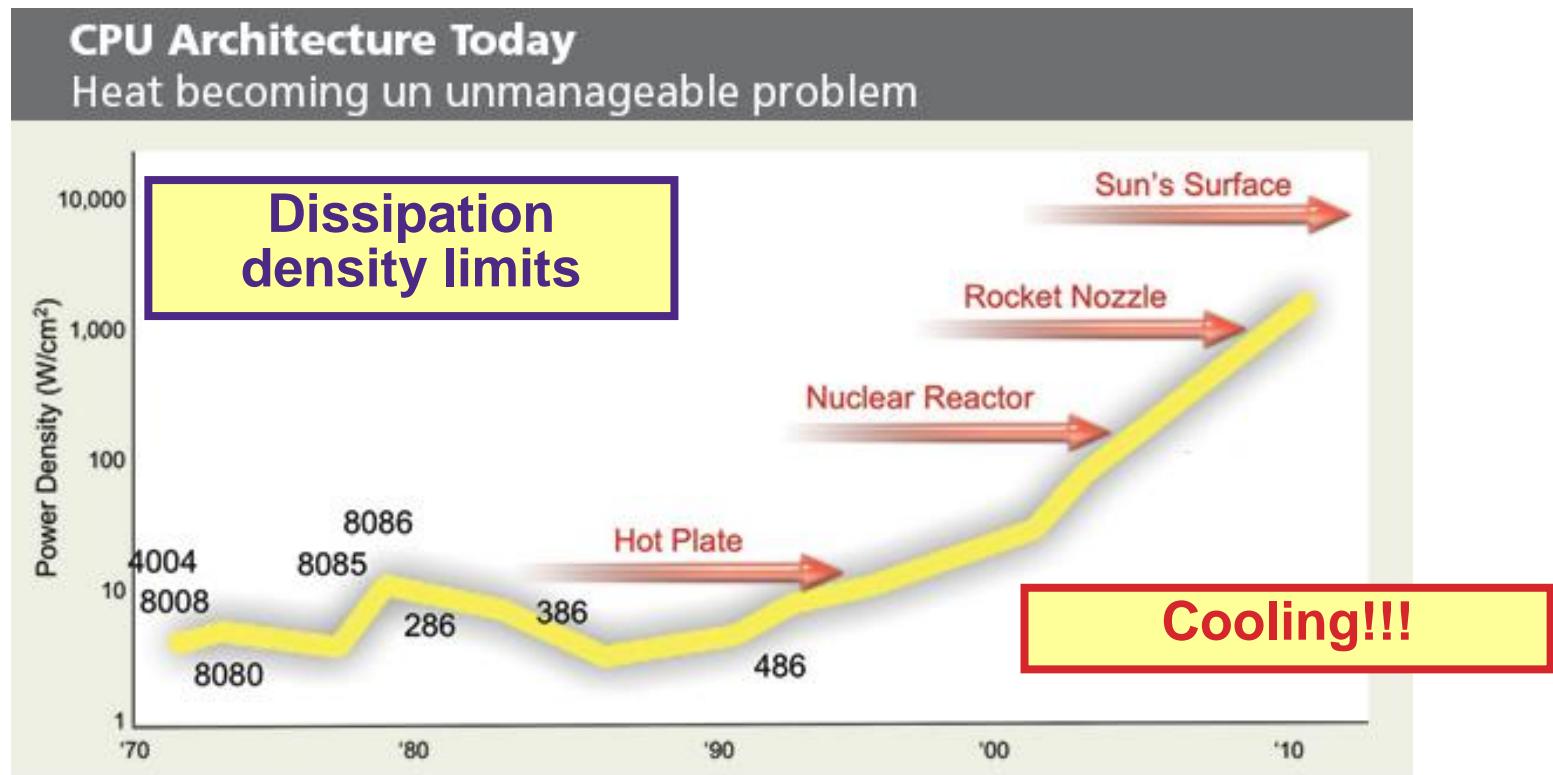
- Continuous increase in case of processors



Adapted from ARC 2010 presentation by Dr. Ram Krishnamurthy, Intel Research

Increase in dissipation density

- Power consumption growth faster than the die size

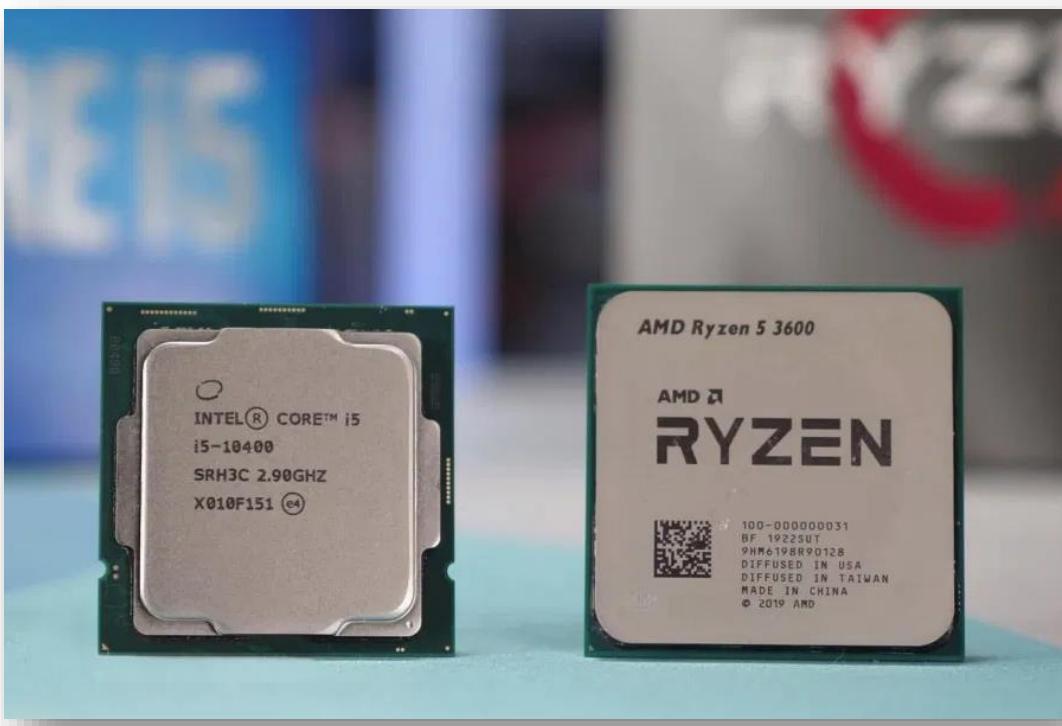


In CPU architecture today, heat is becoming an unmanageable problem.
(Courtesy of Pat Gelsinger, Intel Developer Forum, Spring 2004)

State-of-the-art packaging technologies

FCLGA package

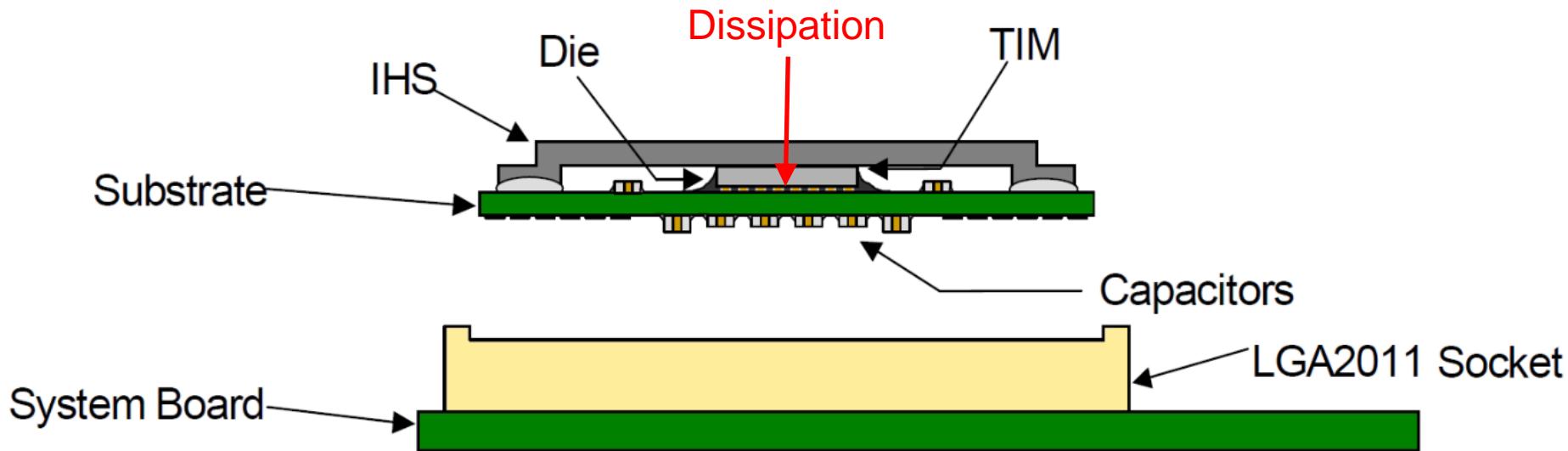
- Large number of pins
- ~100 W power, $V_{core} \sim 1$ V
- ~ 45...55 % of the pins are GND or VDD



Heat flow in state-of-the-art packagings

FCLGA packaging

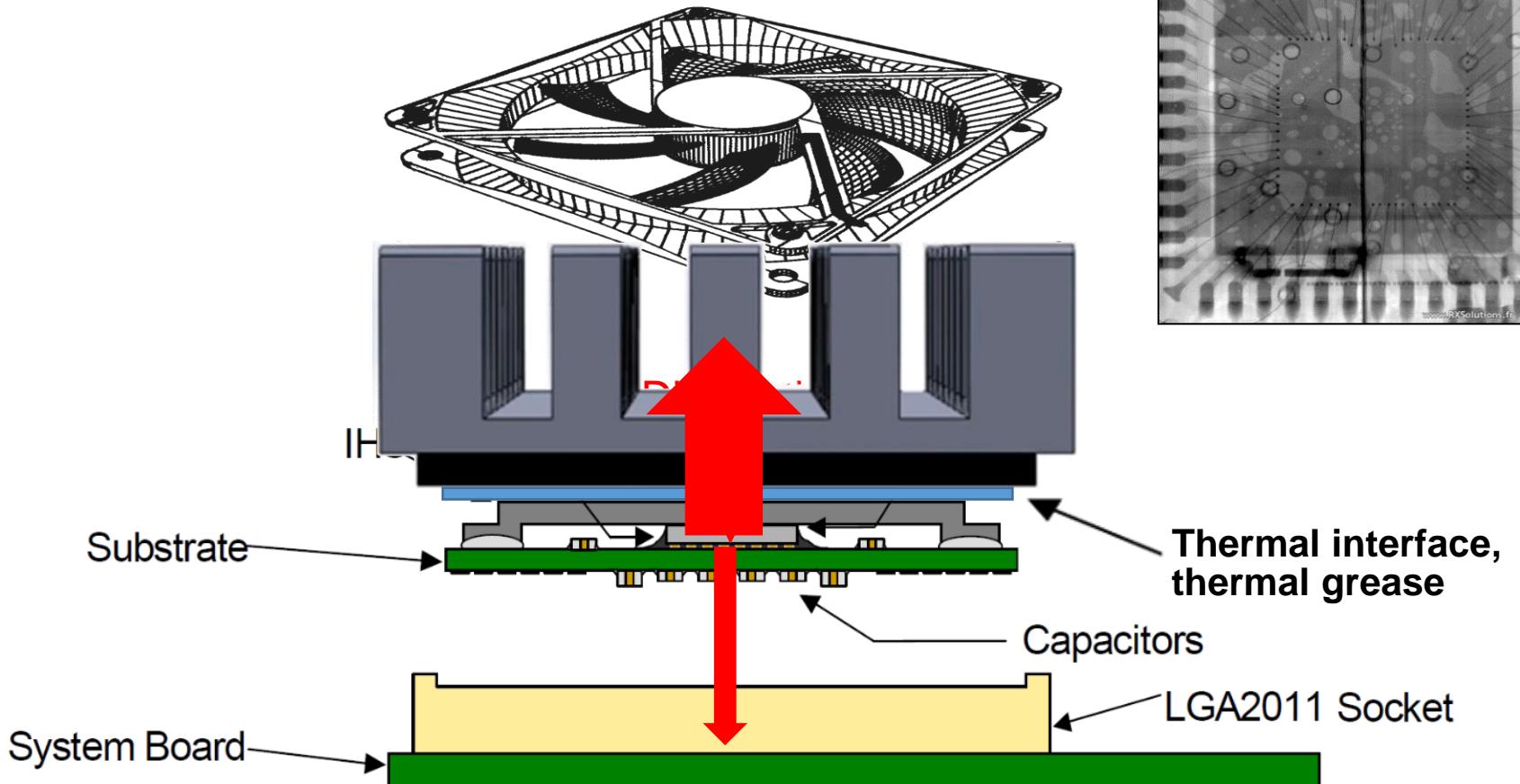
- Chip placed on organic/ceramics/silicon interposer,
- Flip-chip technique,
- 2D heat flow path,
- 1000...2000 pins



Heat flow in state-of-the-art packagings

FCLGA packaging

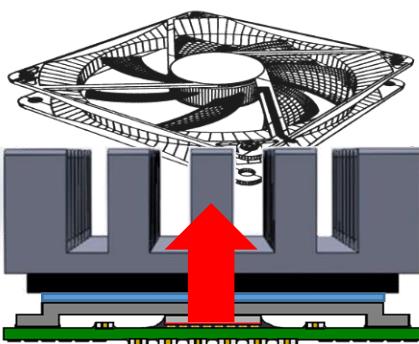
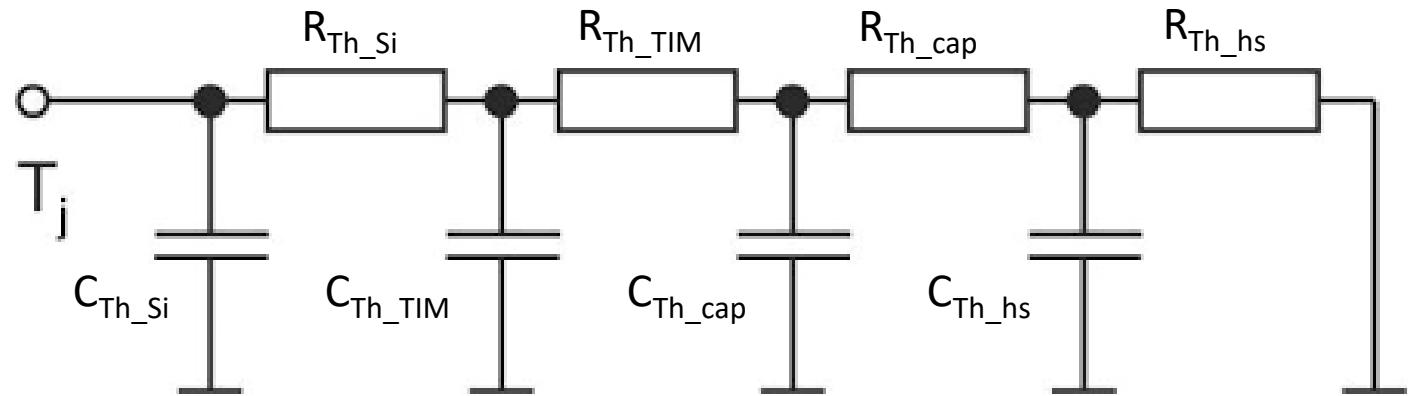
- Importance of the thermal interface material (TIM)



Heat flow in state-of-the-art packagings

FCLGA packaging

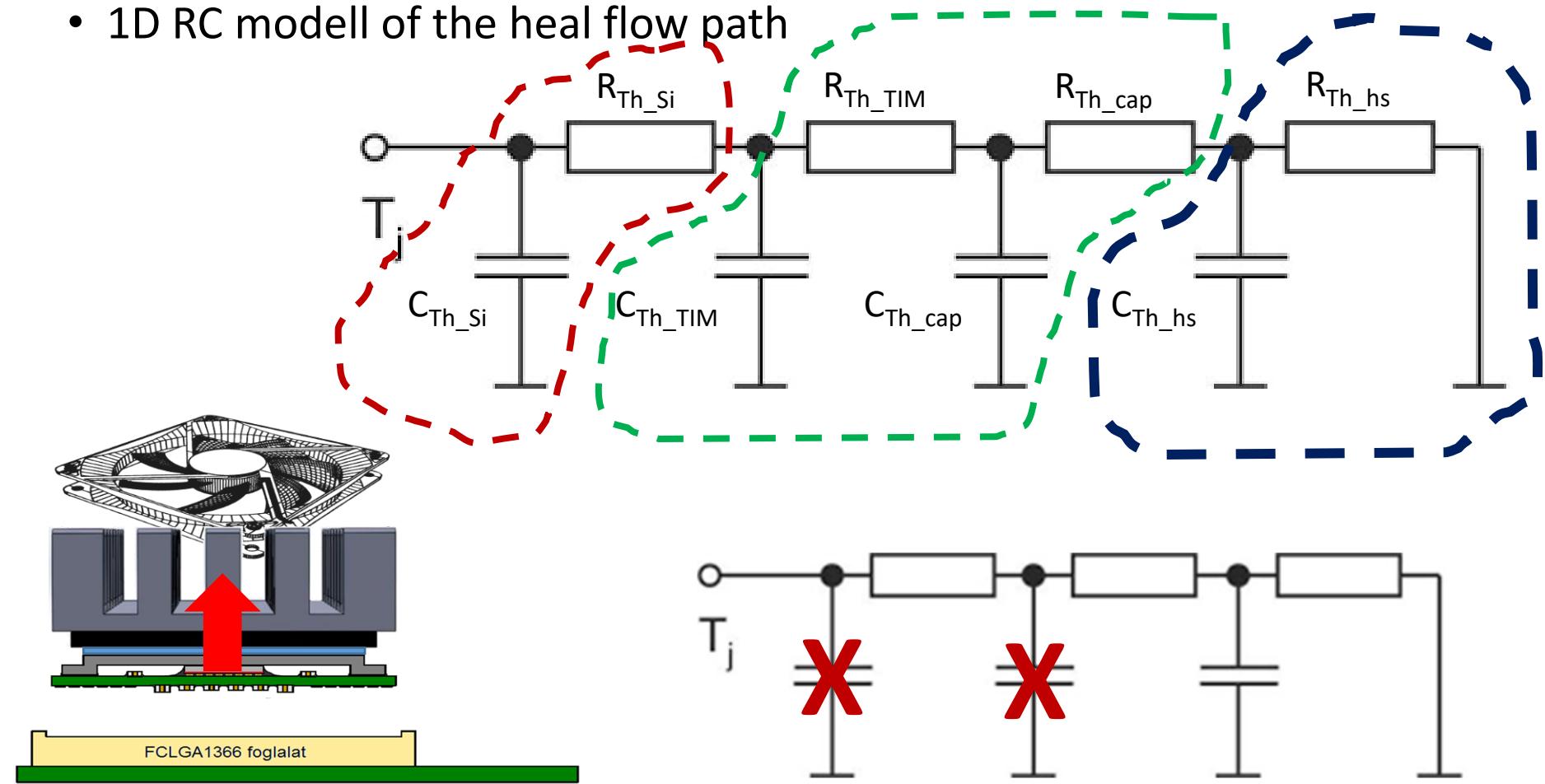
- 1D RC modell of the heat flow path



Heat flow in state-of-the-art packagings

FCLGA packaging

- 1D RC modell of the heat flow path



Heat flow in state-of-the-art packagings

- 3D structures (System-on-Package), More-than-Moore integration worsen the situation

DRAM és processzor közöstokozása mobil eszközökben

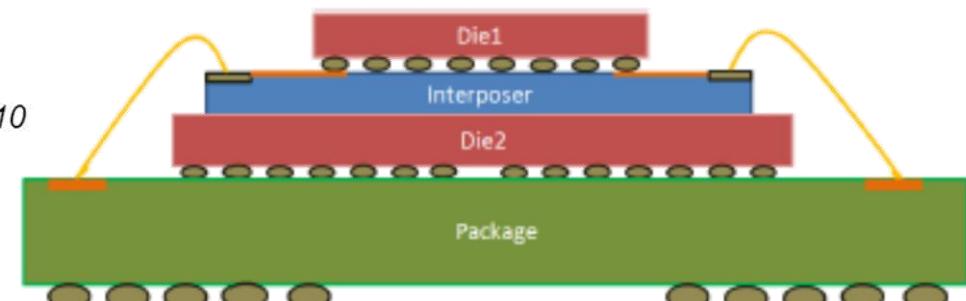
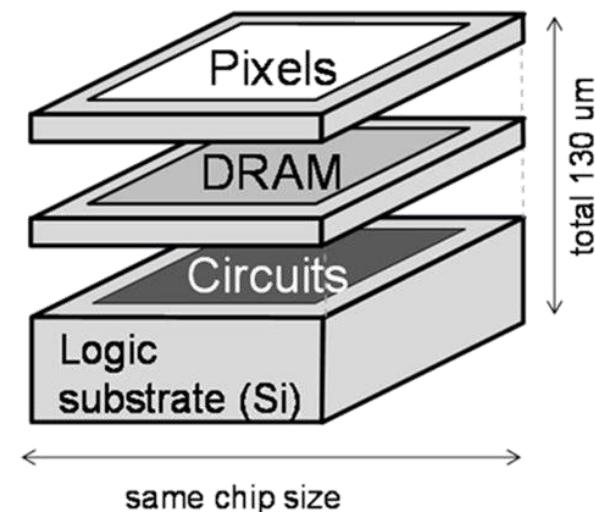


Tokozott Apple A4 rendszerchip



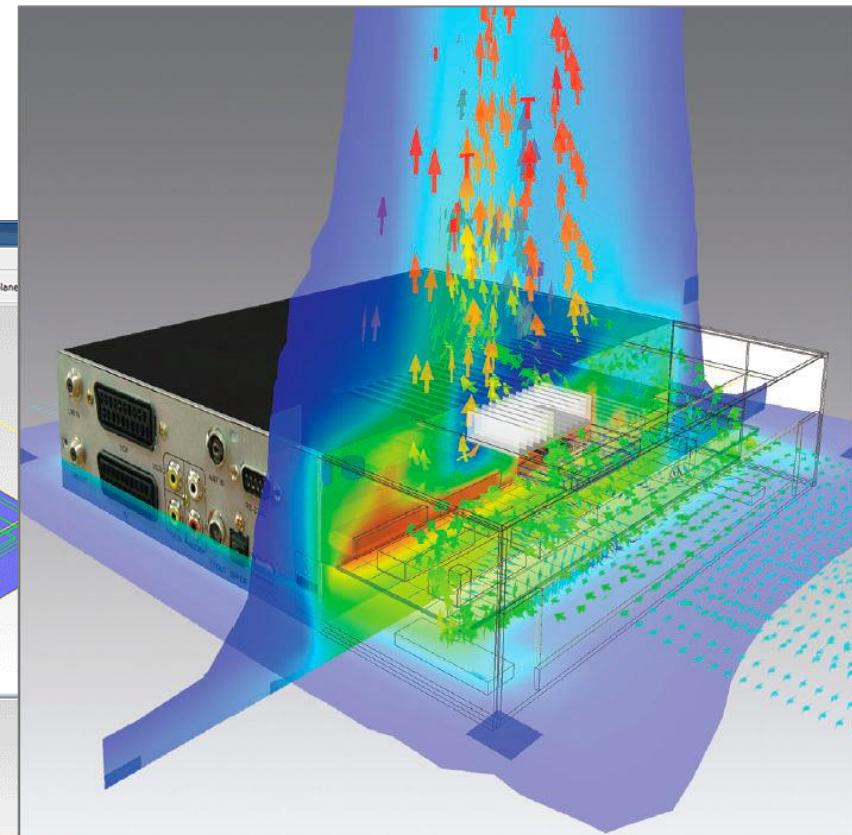
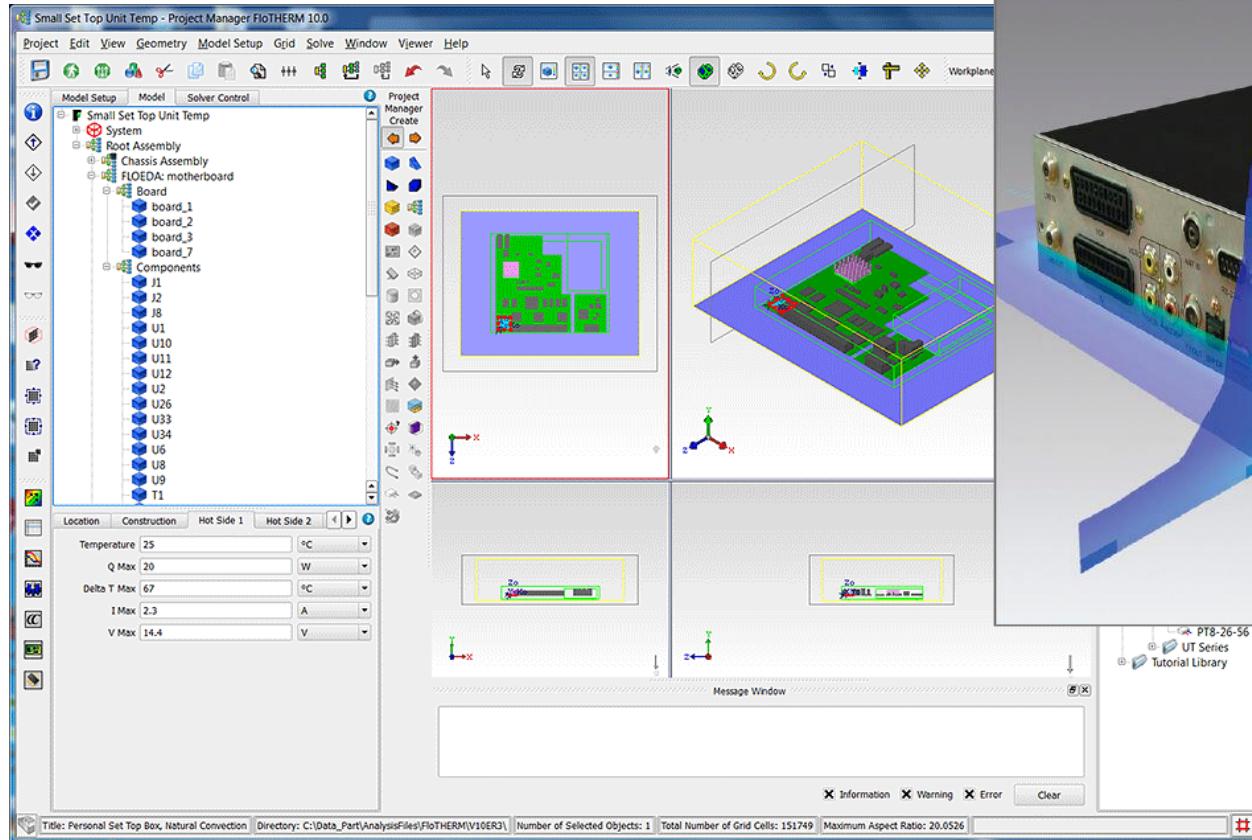
Tokozott Apple A4 rendszerchip keresztmetszeti ábra, iFixit 2010

Structure of modern image sensors



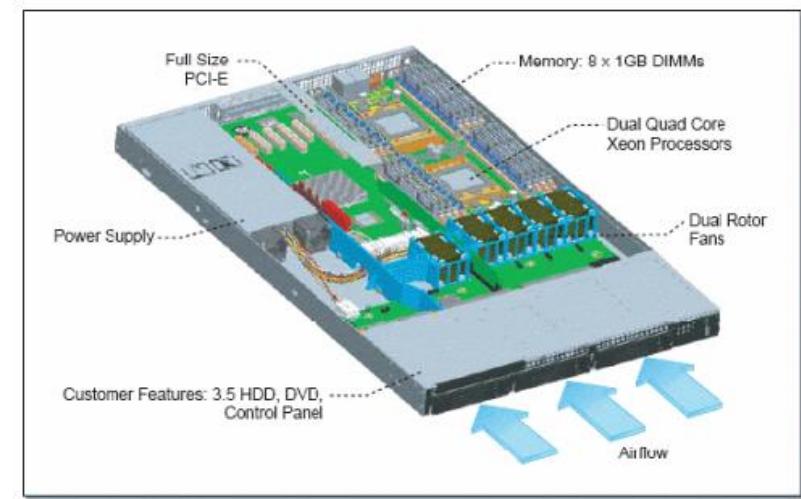
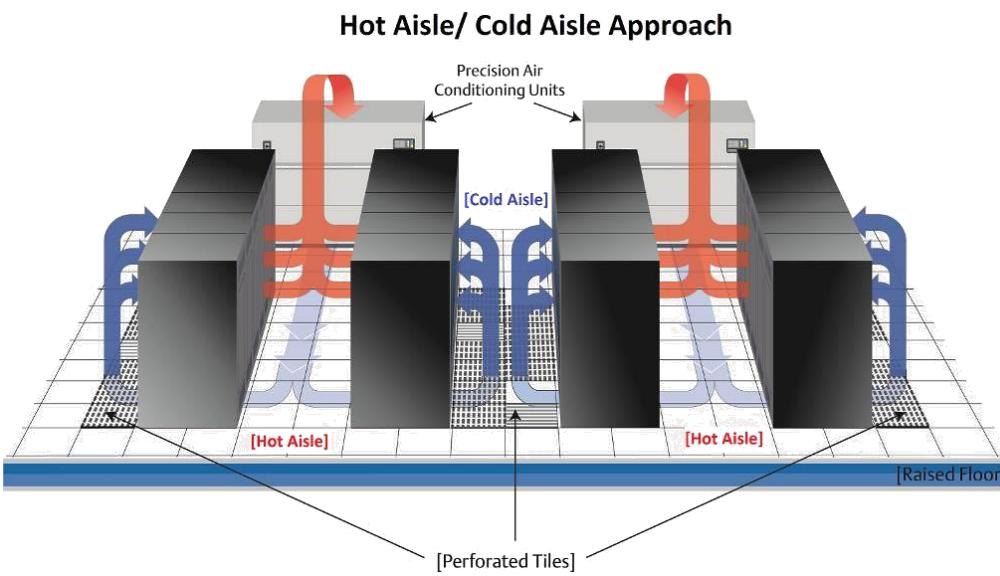
Thermal simulators

- Siemens Mentor - Flotherm CFD



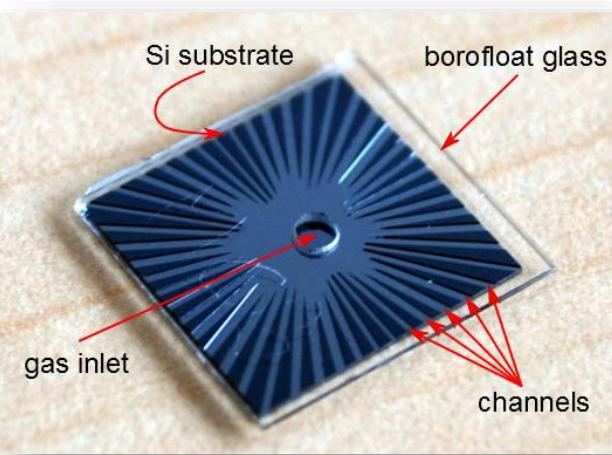
Data center cooling

- Data centers require unique room conditioning setup and power recycle
 - Increased dissipation in racks with little room for fans
 - Power usage can be compared to the national power supply

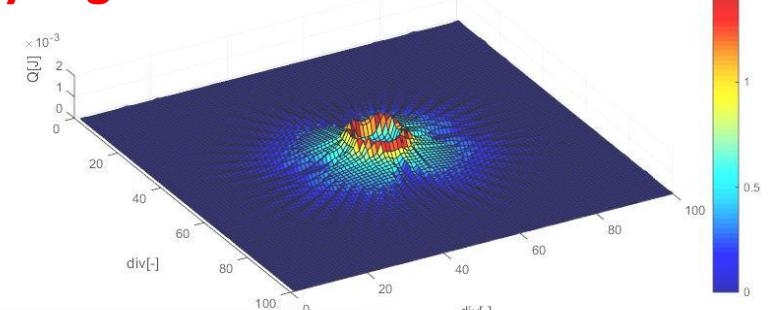


Microchannels for integrated cooling

- 2014-2018 - OTKA K 109202 – **Integrated thermal management for System-on-Package devices**

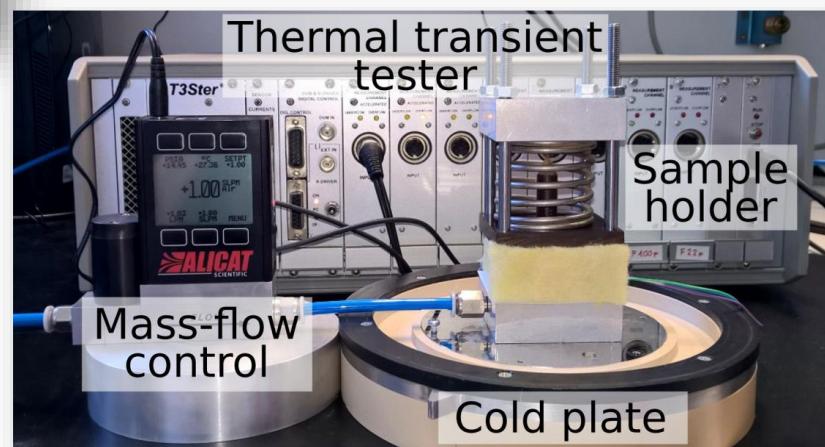


Identifying the value of heat transfer in every segment of the channels



CTM development

Manufacturing technology to integrate the cooling into the silicon itself



Measurement setup



*Budapest University of Technology and Economics
Department of Electron Devices*

Lab. 3

Sequential statements, Processes and Variables

Osama Ali

osamaalisalman.khafajy@edu.bme.hu

Ahmad Halal

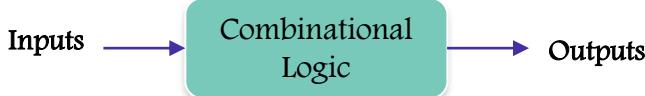
ahmadhalalftesah@edu.bme.hu

20 of May 2022



Sequential Statements

Can be used for building both combinational and sequential logic circuits.

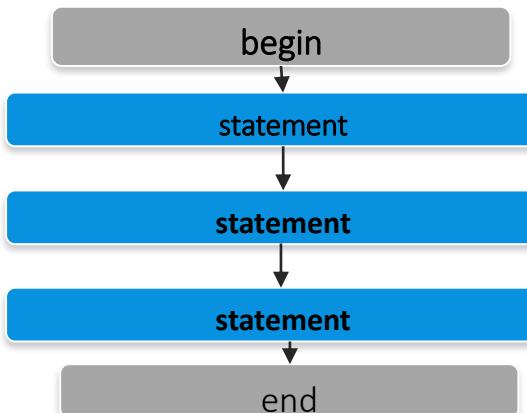


- The output of the circuit depends solely on the current inputs.
- The system requires no memory and can be implemented using conventional logic gates.



- The output does depend on previous inputs.
- A storage elements are required, which are connected to the combinational logic block through a feedback loop.

Unlike the concurrent statements, the sequential statements are executed line by line.



This is done by enclosing the sequential statements inside a VHDL construct known as a “**process**”.
A “process” can appear anywhere after the “begin” statement of the “architecture”.

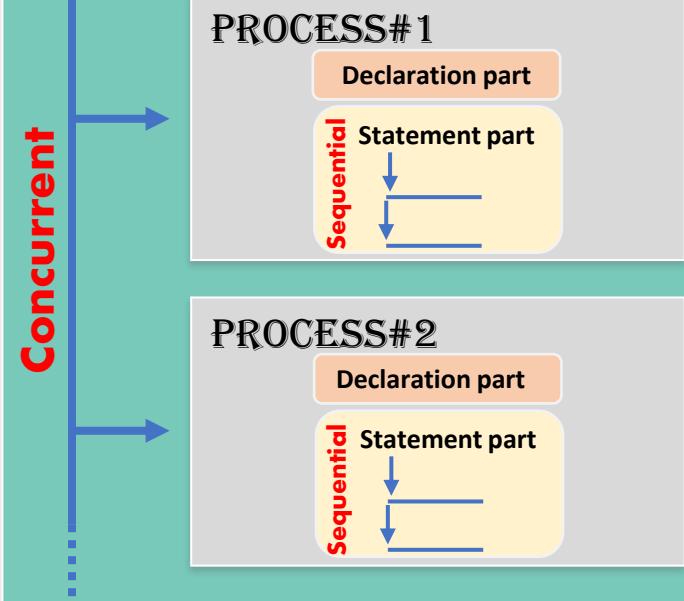
Properties of process Statement

ARCHITECTURE

- Process statement contains only sequential statements.
- The process statement is itself a concurrent statement.
- A process statement has a declaration section and a statement part.
- In the declaration section, types, variables, constants, subprograms, and so on can be declared.
- The statement part contains only sequential statements.
- All processes in an architecture behave concurrently.

DECLARATION PART

DEFINITION PART



END ARCHITECTURE

Sensitivity List

- The process statement can have an explicit sensitivity list that causes the statements inside the process statement to execute whenever one or more elements of the list change value.
- The process must have an explicit sensitivity list or, a WAIT statement. **OR...?**
- The sensitivity list **should** contain all input signals used in that process.

```
process (A,B)
begin
  if (A='1' or B='1') then
    Z <= '1';
  else
    Z <= '0';
  end if;
end process;
```

```
process
begin
  if (A='1' or B='1') then
    Z <= '1';
  else
    Z <= '0';
  end if;
  wait on A,B;
end process;
```

Signal Assignment Versus Variable Assignment

Signal

```
Architecture beh_1 of test is
    signal A,B,C: integer;
    signal Y, Z : integer;
    signal M, N : integer;
begin
    process (A,B,C,M,N)
    begin
        M <= A;
        N <= B;
        Z <= M + N;
        M <= C;
        Y <= M + N;
    end process;
end Architecture;
```

Variable

```
Architecture beh_1 of test is
    signal A,B,C: integer;
    signal Y, Z : integer;
begin
    process (A,B,C)
        variable M, N: integer;
    begin
        M := A;
        N := B;
        Z <= M + N;
        M := C;
        Y <= M + N;
    end process;
end Architecture;
```



If Statement

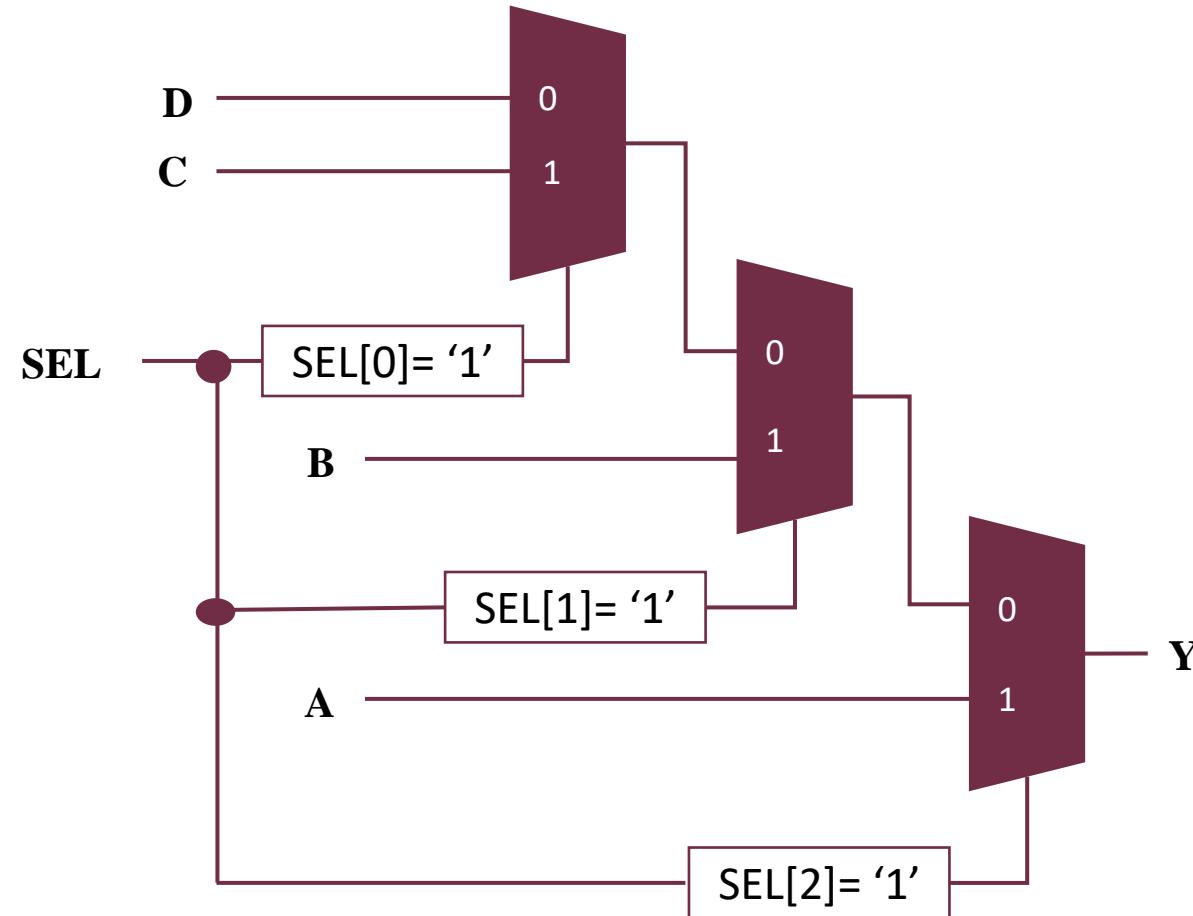
- Syntax:

```
if (condition 1) then  
  
    elseif (condition 2) then  
  
        elseif (condition 3) then  
  
            else  
  
        end if;
```

- “IF” statement evaluates each condition in order
- Statement can be nested
- **Spaghetti code** (Avoid using more than three levels of **if...else** statements)
- When defining the condition, use parentheses to differentiate levels of operations on the condition

If Statement

```
process (sel, a, b, c, d)
begin
if sel(2) = '1' then
    y <= a;
elsif sel(1) = '1' then
    y <= b;
elsif sel(0) = '1' then
    y <= c;
else
    y <= d;
end if;
end process
```



Generates a priority structure.

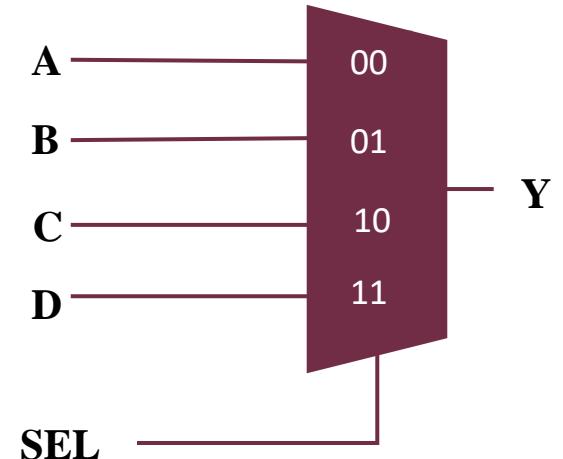
Corresponds to “when-else” command in the concurrent part.

Case Statement

- Syntax:

```
case expression is
    when choice1 => {statements}
    when choice2 => {statements}
    when others => {statements}
End case;
```

```
process (sel, a, b, c, d)
begin
case sel is
    when "00" => Y <=a;
    when "01" => Y <=b;
    when "10" => Y <=c;
    when others => Y <=d;
end case;
end process;
```



- “Case” statement is a series of parallel checks to check a condition.
- Statements following each “when” clause is evaluated only if the choice value matches the expression value.
- Corresponds to “with...select” in concurrent statements

Process statement

- Two types of processes
 - Combinatorial (asynchronous)
 - Clocked (synchronous)

Combinatorial Process (asynchronous)

- Generates combinational logic.
- All inputs must be present in the sensitivity list.

```
process (a, b, c)
begin
  x <= (a and b) or c;
end process;
```

Clocked Process (synchronous)

Any signal assigned under a clk' event generates a flip-flop

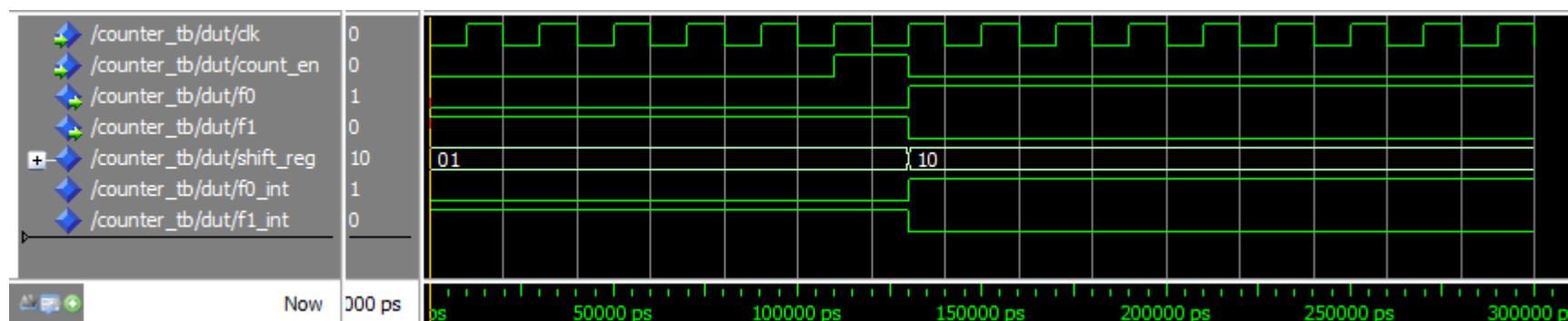
```
process (clk)
begin
  if (clk' event and clk = '1') then
    Q <= Data;
  end if;
end process;
```



- Clocked processes having an “else” clause will generate wrong hardware.

Bad Hardware Description

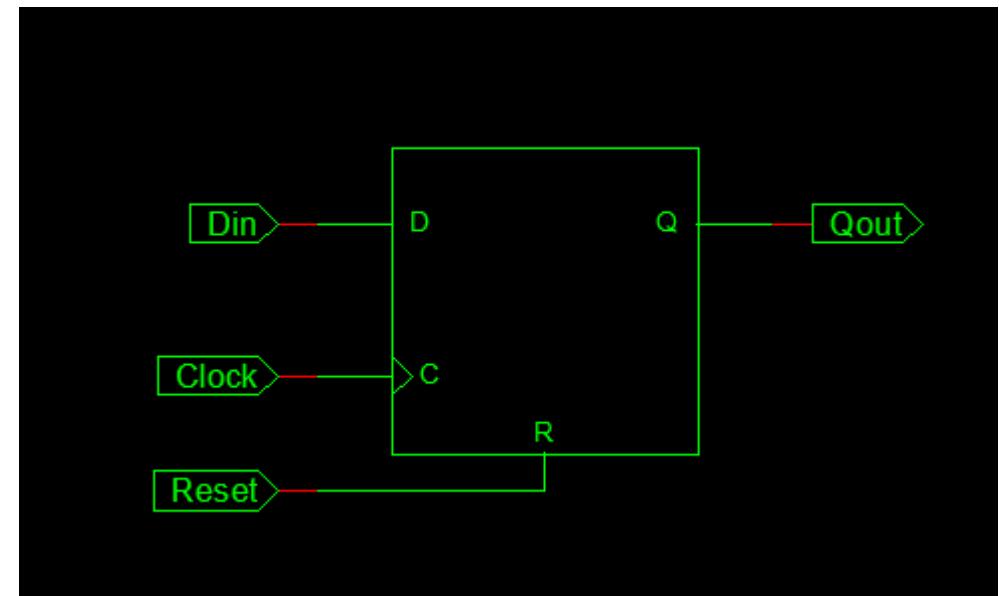
```
process (clk)
begin
  if (clk' event and clk = '1') then
    out1 <= a and b;
  else
    out1 <= c;
  end if;
end process;
```



Synchronous

- Synchronous Reset: Flip-flops are reset on the active edge of the clock when reset is held active.

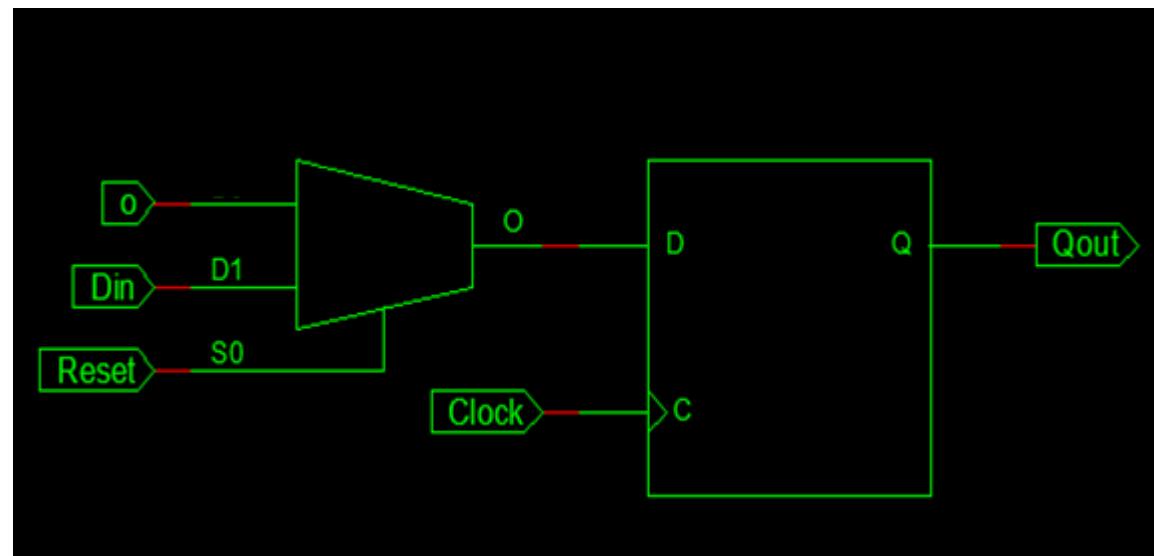
```
process (clk)
begin
    if (clk' event and clk = '1') then
        if (Reset = '1') then
            Q <= '0';
        else
            Q <= D;
        end if;
    end if;
end process;
```



Asynchronous

- Asynchronous Reset: Flip-flops are cleared as soon as reset is asserted.

```
PROCESS (clk, rst)
BEGIN
  IF (clk'EVENT AND clk='1') THEN
    IF (rst='1') THEN
      q1 <= '0';
    else
      q1 <= d;
    END IF;
  END IF;
END PROCESS;
```



Clock generation in the testbench

```
library ieee;
use ieee.std_logic_1164.all;

entity Lab_4 is
end Lab_4;

architecture TEST_behavior of Lab_4 is

-- Insert component declaration for device under test (DUT) here.

-- Insert signal declarations here.

SIGNAL CLKK : STD_LOGIC := '0';
SIGNAL RST : STD_LOGIC := '0';
SIGNAL D : STD_LOGIC := '0';

begin
-- Clock Generation
CLKK <= NOT CLKK AFTER 10 NS;
-- Insert component instantiation (i.e. port map statement) here.

stimulus: process
begin
-- Insert your testbench code here.

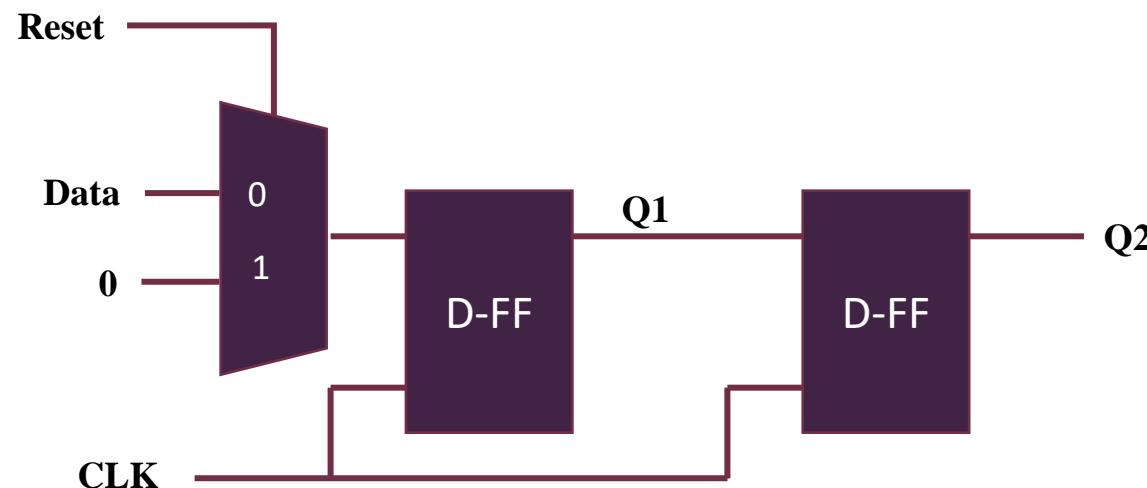
end process;

end TEST behavior;
```



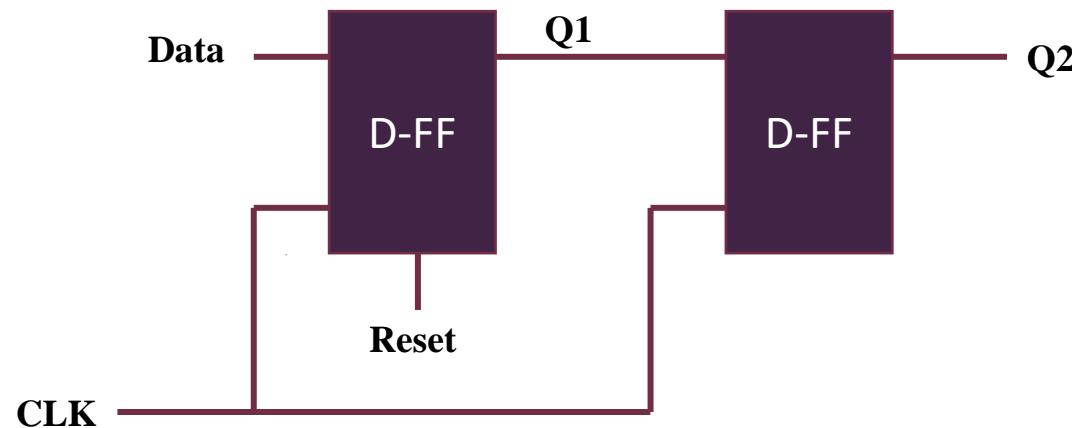
Write a VHDL code that can model the below Circuit of D flip-flop using behavioral modeling.

- The circuit consists of two D_flip-flops one with **synchronous reset** and **one without**.
- Create a new project and label it as “Lab_5_1”
- Download the VHDL testbench file (TestBench_Lab5.vhd.). Develop it to test and analyze the design behavior.



Write a VHDL code that can model the below Circuit of D flip-flop using behavioral modeling.

- The circuit consists of two D_Flip-flops one with **Asynchronous reset** and **one without**.
- Create a new project and label it as “Lab_5_2”
- Download the VHDL testbench file (TestBench_Lab5.vhd.). Develop it to test and analyze the design behavior.





EET

Microelectronics, BSc course

Introduction

SZABÓ Péter Gábor, assoc. prof.

BOGNÁR György, assoc. prof.

POPPE András, prof.





Microelectronics BSc. Course

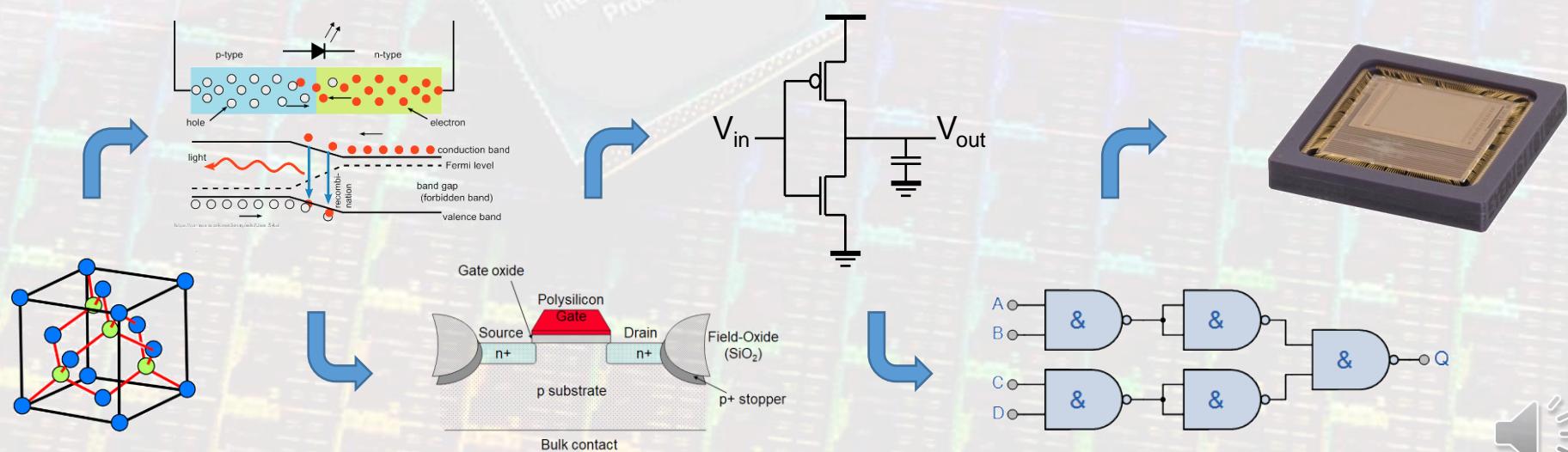
- ▶ Microelectronics usually means analog electronics design (mainly on chip level).
 - ▶ Analog electronics design in the BME curriculum: Electronics 1.
 - ▶ Microelectronics extends it with chip level considerations
 - ▶ A.S. Sedra, K.C. Smith, Microelectronic Circuits, Oxford Series in Electrical and Computer Engineering 6th Edition, ISBN-13: 9780199339136
- ▶ Related knowledge, skills
 - ▶ Semiconductor and quantum physics, solid-state physics, semiconductor manufacturing technology, Mixed-signal CAD design, simulation methodologies, packaging, modern VLSI circuits, etc.





Microelectronics BSc. Course

- ▶ New skills and knowledge, modern technologies, methodologies:
 - ▶ Operation, build-up, manufacturing of applied semiconductor devices (diode, transistor)
 - ▶ Logic gates and standard cells
 - ▶ Design methodology of modern ICs, VLSI circuits
 - ▶ MEMS and optoelectronic devices, etc.





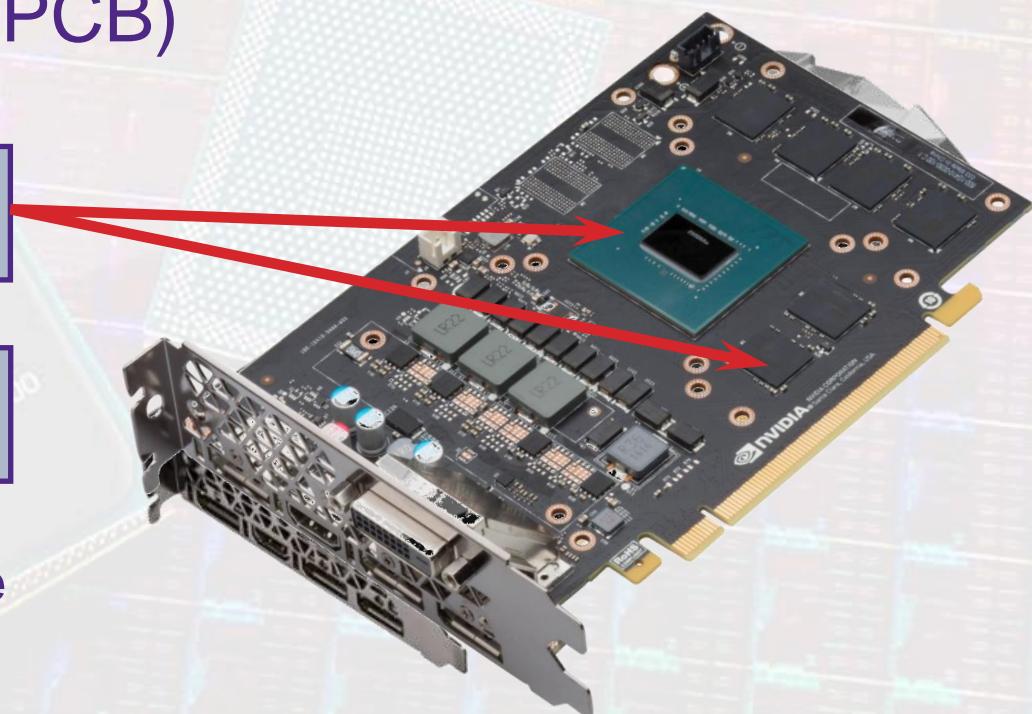
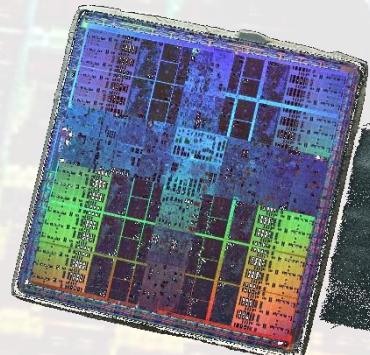
Basics

- ▶ Integrated circuits (IC-s) on a surface mounted printed circuit board (PCB)

We are interested in seeing
the inside of these

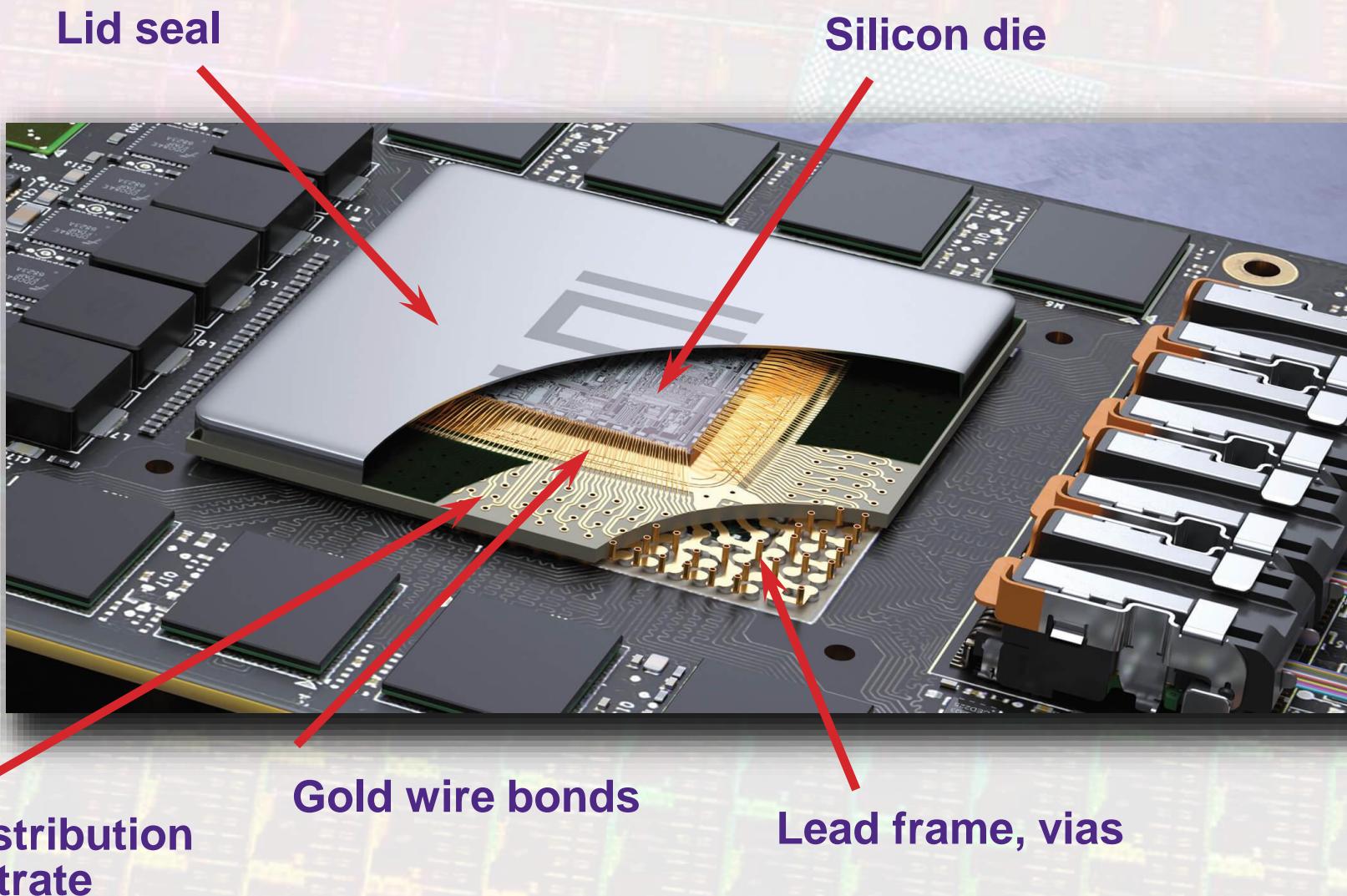
Let's have a look inside the
package!

There are silicon chips inside



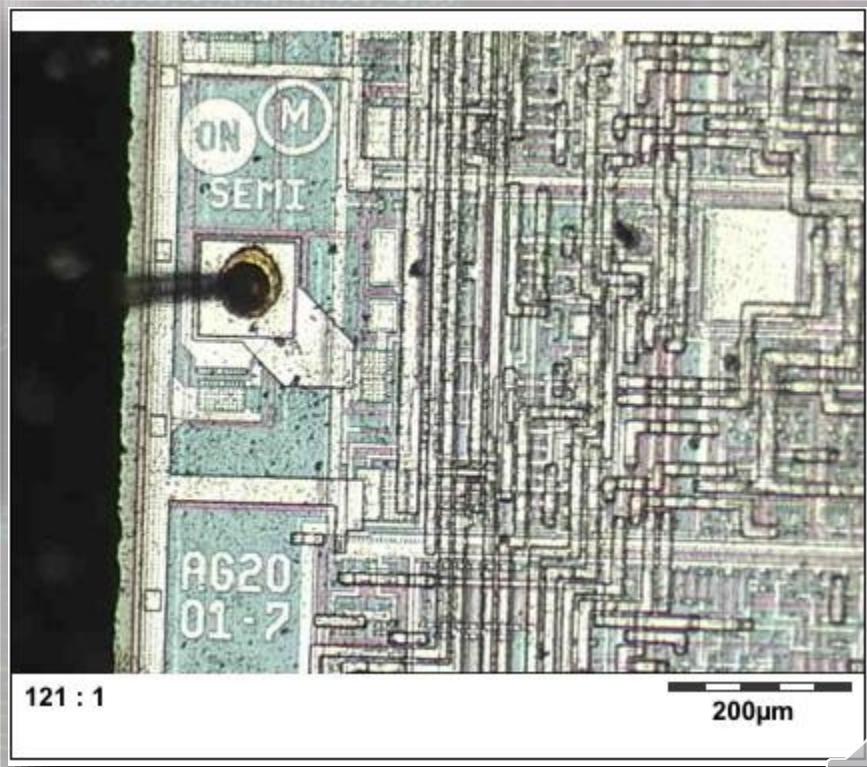
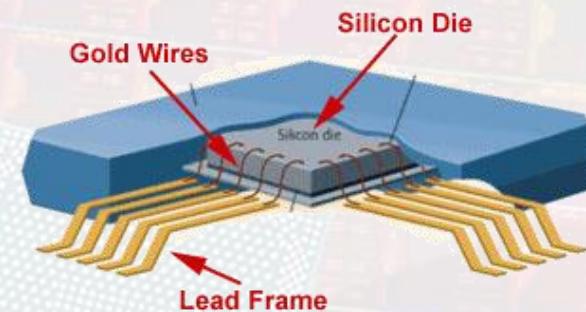
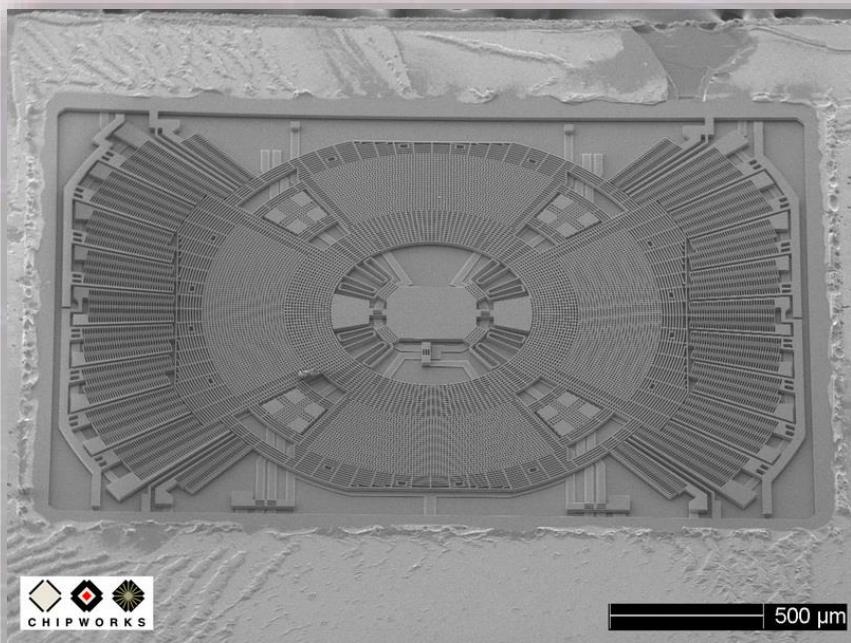


Inside the package





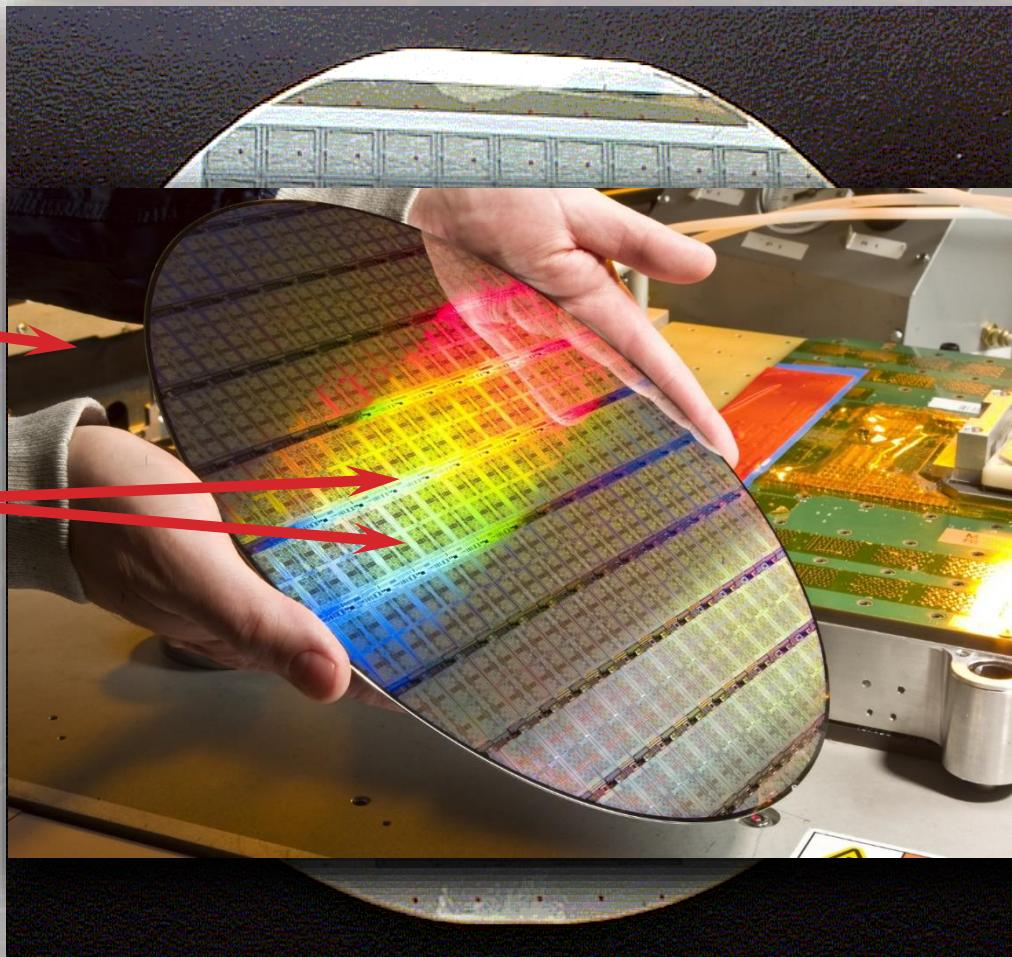
Inside the package





Basics

- Wafer, die or chip:
 - wafer made of perfect single Si crystal
- There are many identical dies (chips) on a wafer
- Wafer diameters: 15-20-25... cm or 4-6-8"
- 100...2000 dies/wafer, manufactured simultaneously





Si single crystal wafers





Si single crystal wafers

The screenshot shows a web browser window with the URL semi.org/en/wafer-size-transition-450-mm-5-misconceptions. The page features the SEMI logo and navigation links for About, Trending, Resources, Collaboration, and Events. Below the navigation is a large, blurred image of a city at night with streaks of light from traffic. The main content title is "Wafer-Size Transition to 450 mm? 5 Misconceptions". The text discusses common misconceptions about the transition to 450 mm wafers, mentioning Moore's Law and R&D costs.

Wafer-Size Transition to 450 mm? 5 Misconceptions

It seems that everyone in the industry "knows" that bigger wafers mean better productivity and more profits, and everyone "knows" that wafer size changes happen every 10 years, and everyone "knows" that bigger wafers are at least twice as productive as small wafers. However, many experts in the industry think that what "everyone knows" is not truly reality.

There is an industry-wide curiosity about the reality behind a wafer-size transition to 450 mm wafers. According to the ITRS, this transition should happen in 2012 to keep the industry on Moore's Law, and a few companies are pushing hard to make this happen.

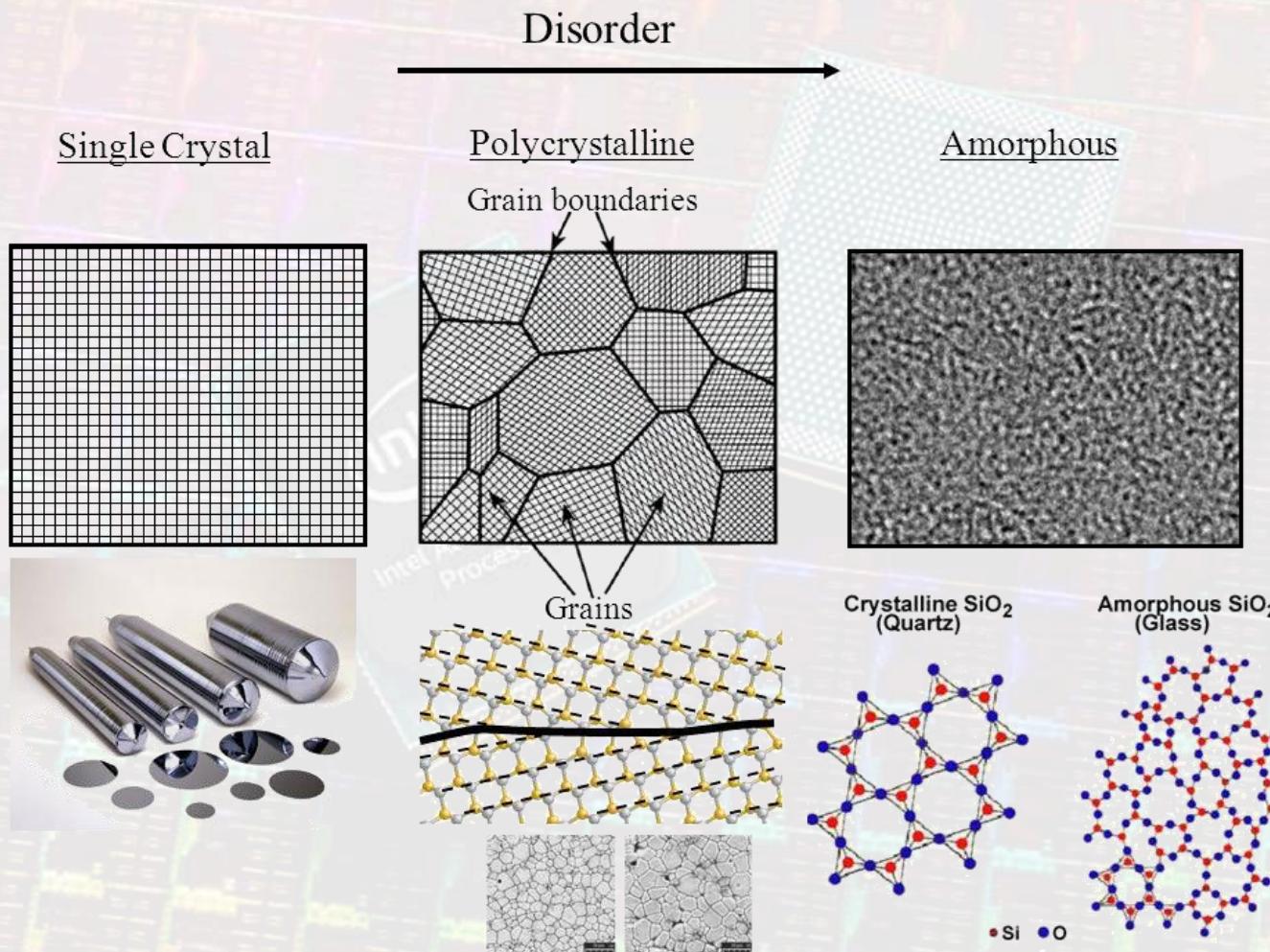
However, the fundamental assumptions that are driving the push for this transition are overstated. Moreover, limited research and development dollars available to tool and equipment suppliers means that investing R&D capital prudently is essential to survival of many companies in the semiconductor supply chain. A white paper released by SEMI in 2005 shows that semiconductor industry R&D dollars are becoming more constrained as advanced process R&D costs rise.

A transition to 450 mm wafers is an extremely expensive and risky proposition—estimates run to well over \$25 billion at the high end. The semiconductor industry simply cannot afford to make such an expensive investment based on future





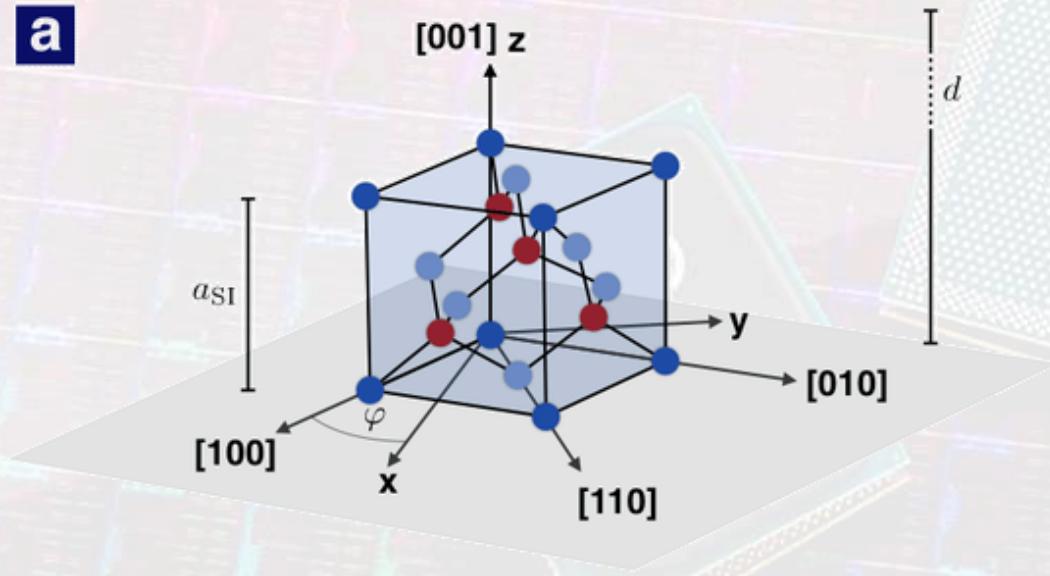
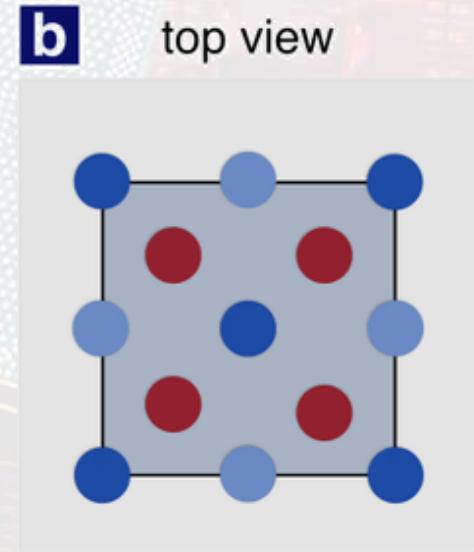
Crystalline structures



Published by [Blanche Parsons](#)



Crystalline structures

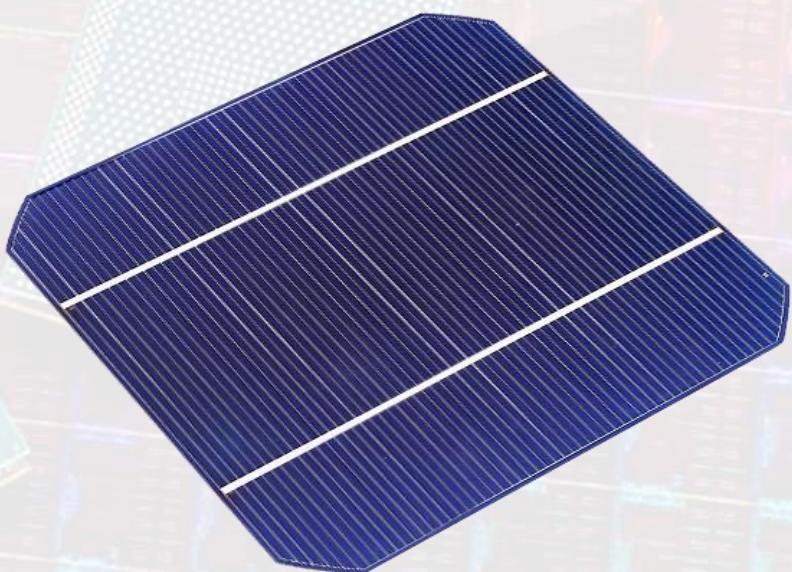
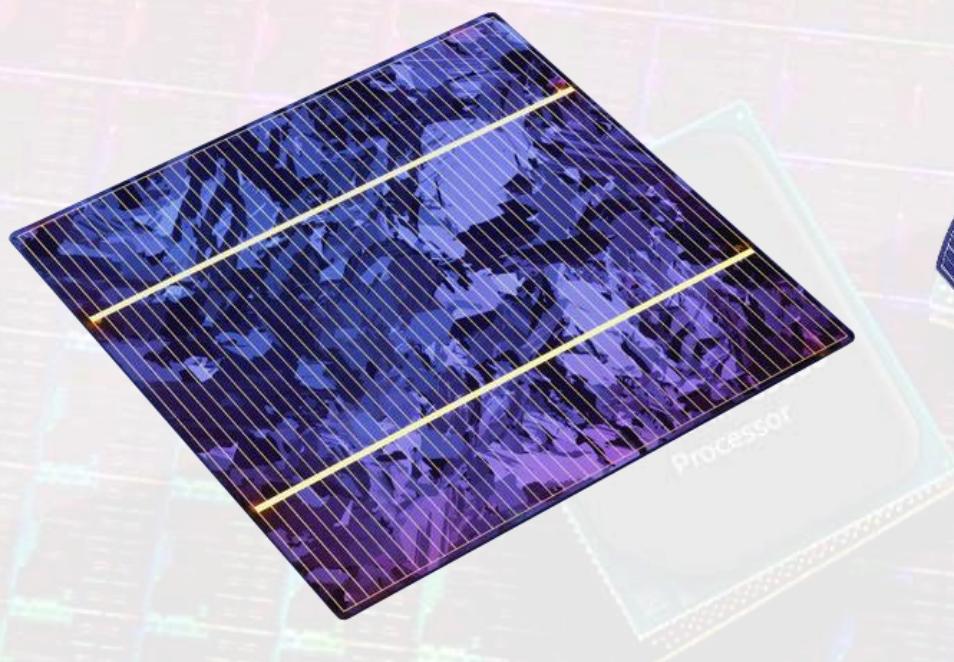
a**b**

- Brendel, Christian. (2019). Topologically Protected Transport of Phonons at the Nanoscale.





Results of crystallization

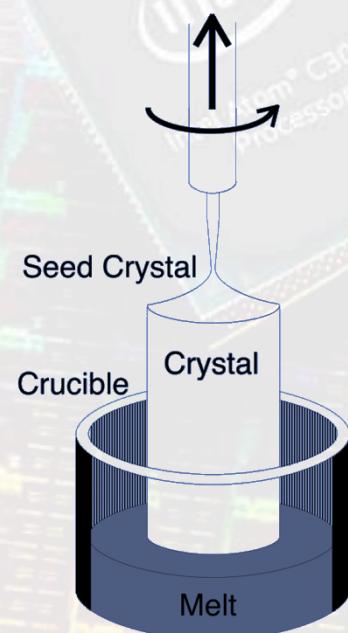
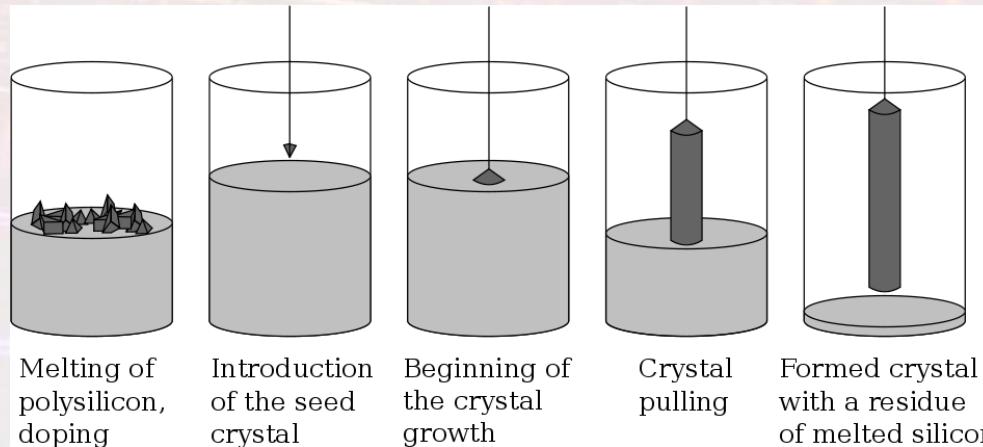


- Poly-Crystalline Solar Cell
- Mono-Crystalline Solar Cell



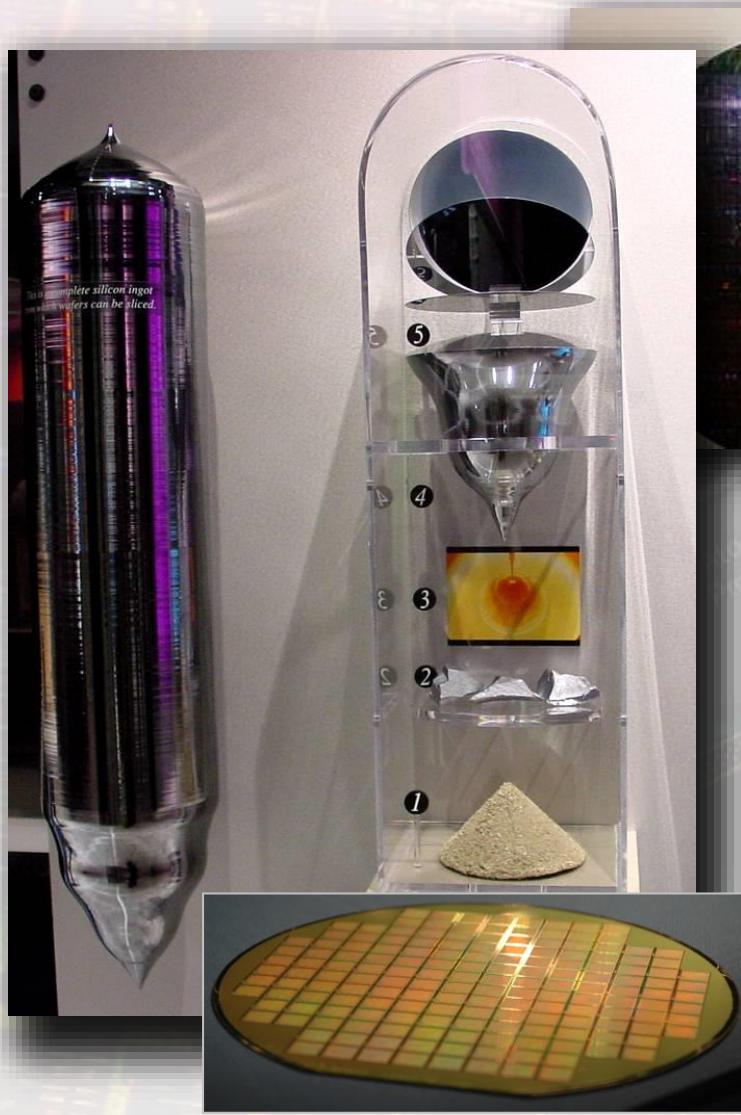


Manufacturing of single crystal wafers

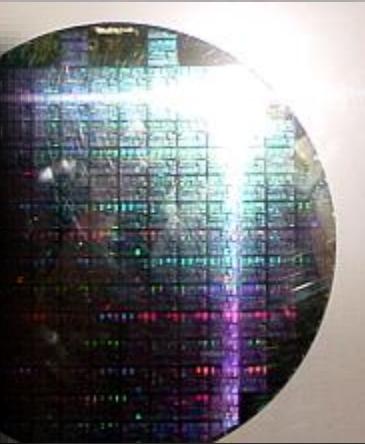




Si single crystal, wafers (2007: 12")



Finished Si wafer

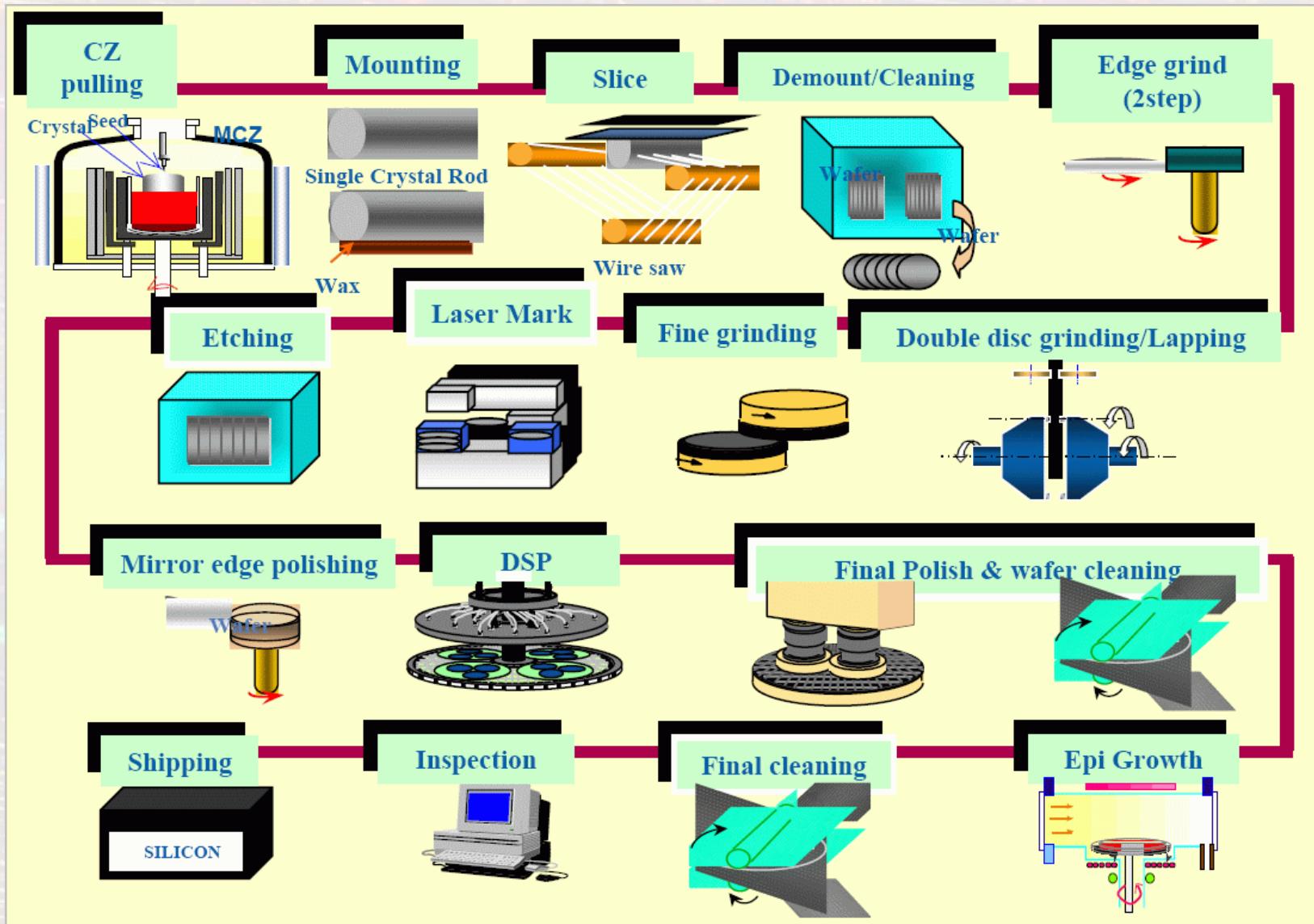


8" wafer with
Pentium
processors, Intel
Museum



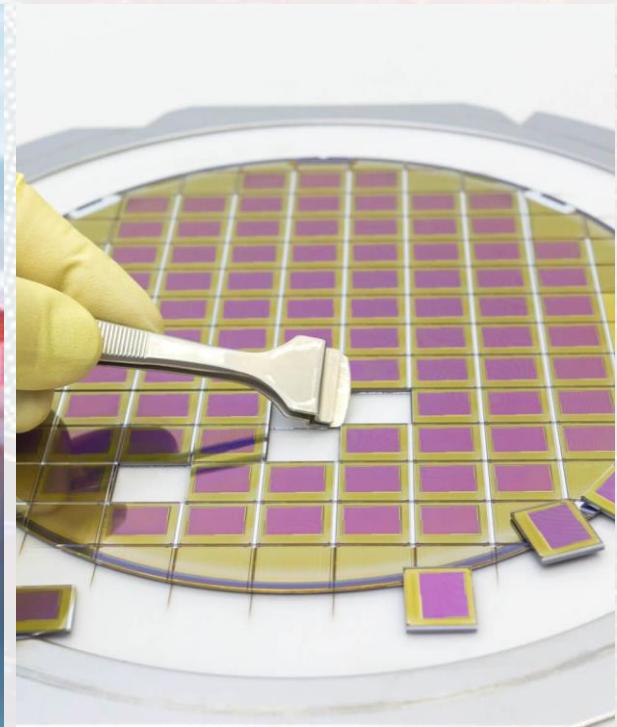
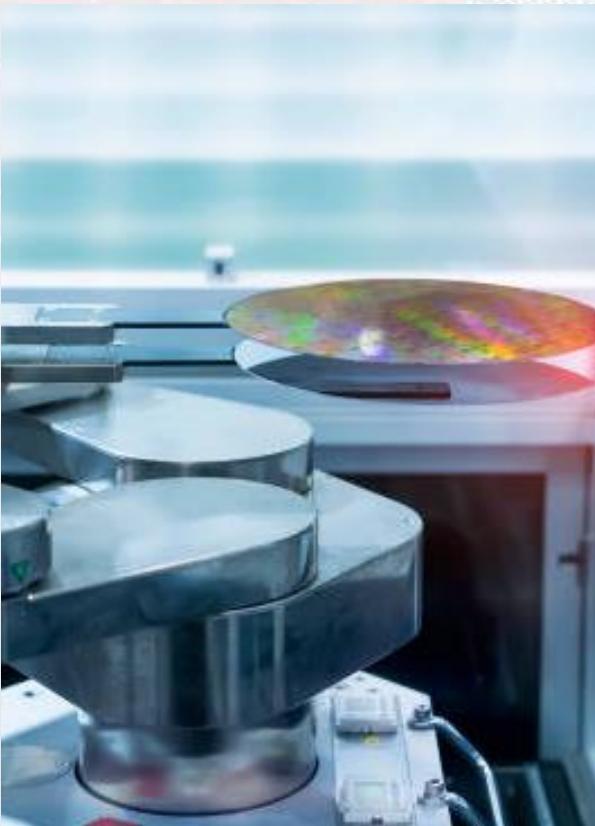


Manufacturing of wafers





Manufacturing of wafers





Basic processing principles

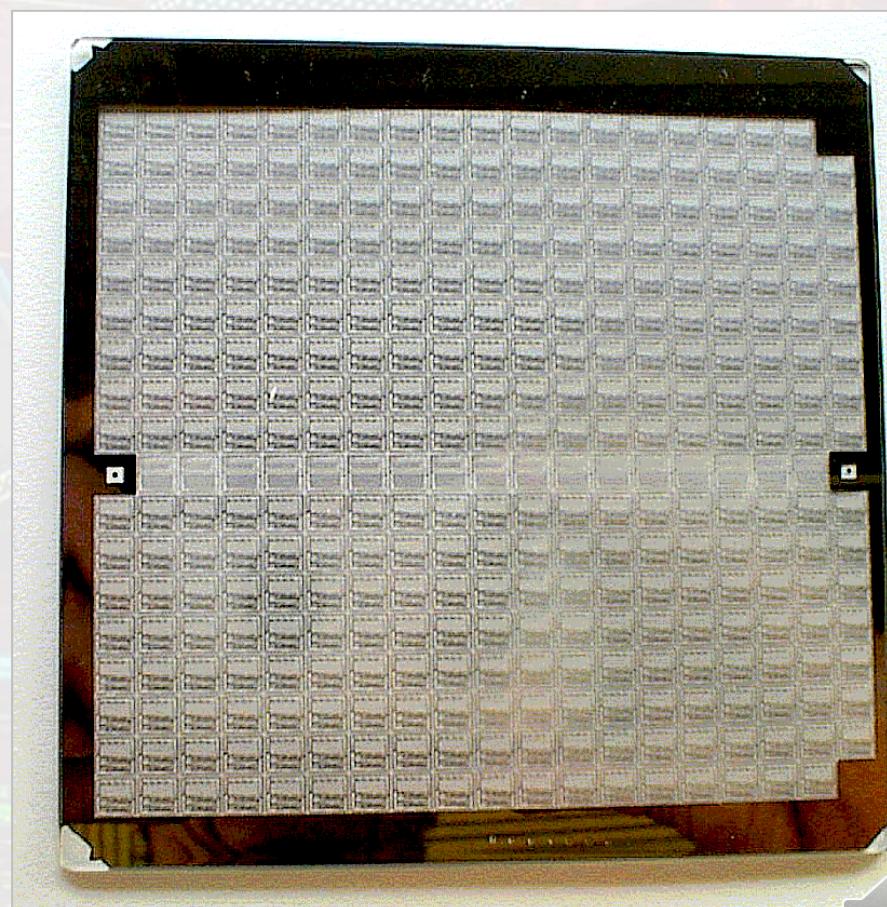
- **Layer growth or deposition:** new material layer is formed over the entire surface of the wafer
- **Patterning:** some patterns are formed in the deposited layer
 - deposition of a photo-sensitive lack (photoresist)
 - photographing the pattern onto the lack
 - developing the photoresist: pattern formed in the resist layer
 - transferring the pattern from the resist to the material layer underneath by some kind of **etching**
 - removal of the resist
- **In-depth deposition of external material:** ion implantation (formerly: diffusion)





Patterning

- The original pattern is on a so called photo-mask
 - made of chromium on glass substrate
 - many times larger than a chip
- Need for high level of accuracy:
 - 7nm over 30cm!
 - accuracy: $10^{-8}:1$
- Visible light:
 - $\lambda=0.3\text{-}0.6 \mu\text{m}$
 - deep UV needed! (193nm)
 - extreme UV (13.5nm)





Making tiny structures (patterning)

- Basic process: **photolithography**. Small features of a mask are photographed onto a lack called photo-resist



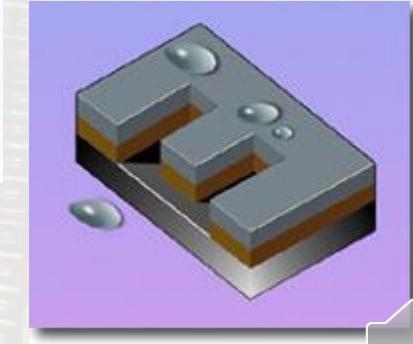
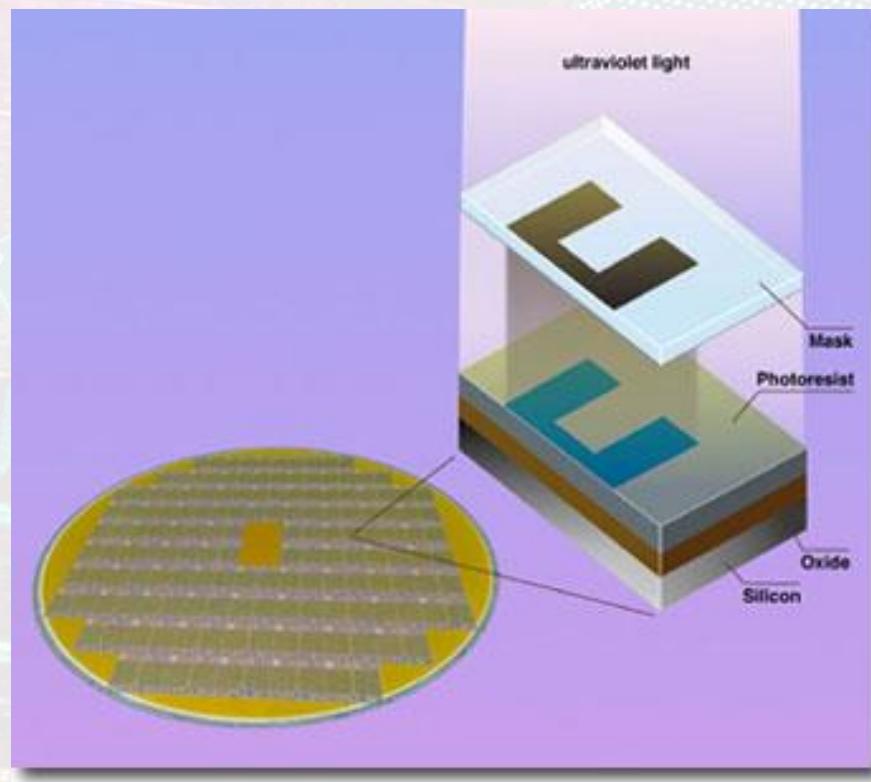
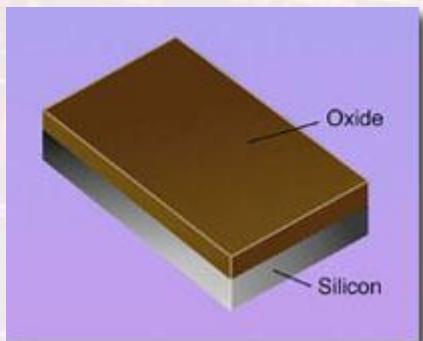
- ▶ UV light is used
- ▶ Resist is not sensitive to yellow

Semiconductor Technology Lab,
Microelectronics branch at DED



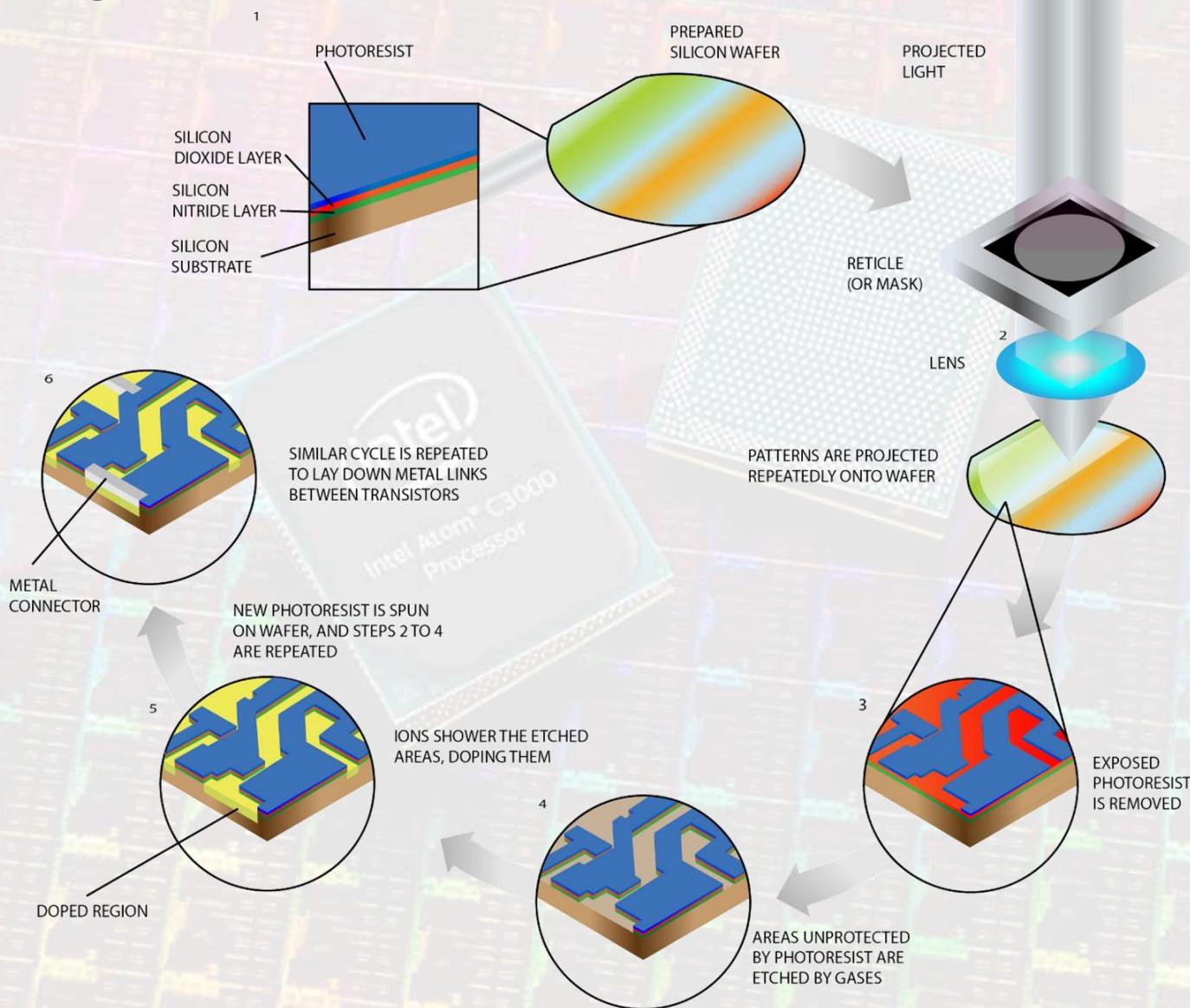


Patterning: photolithography





Patterning





Typical types of patterns on a chip

Metallization

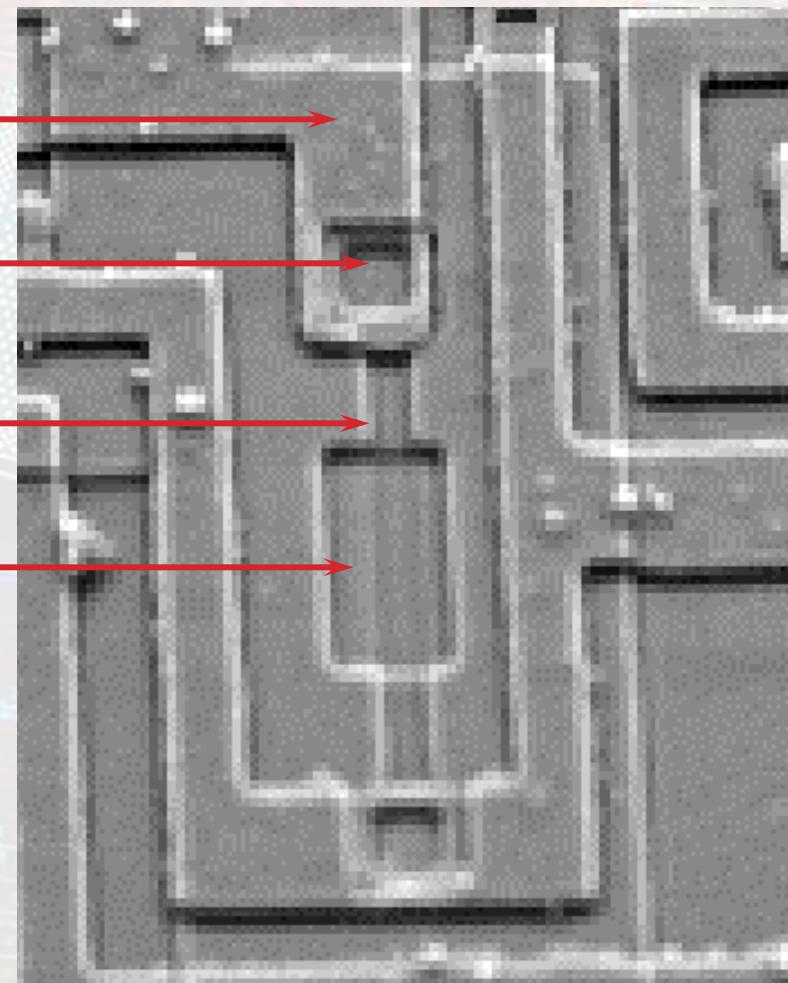
Contact window

P diffused (doped) region

N diffused (doped) region

One process: 15..25 masks

Problem: alignment of masks



SEM of a resistor in an analog IC

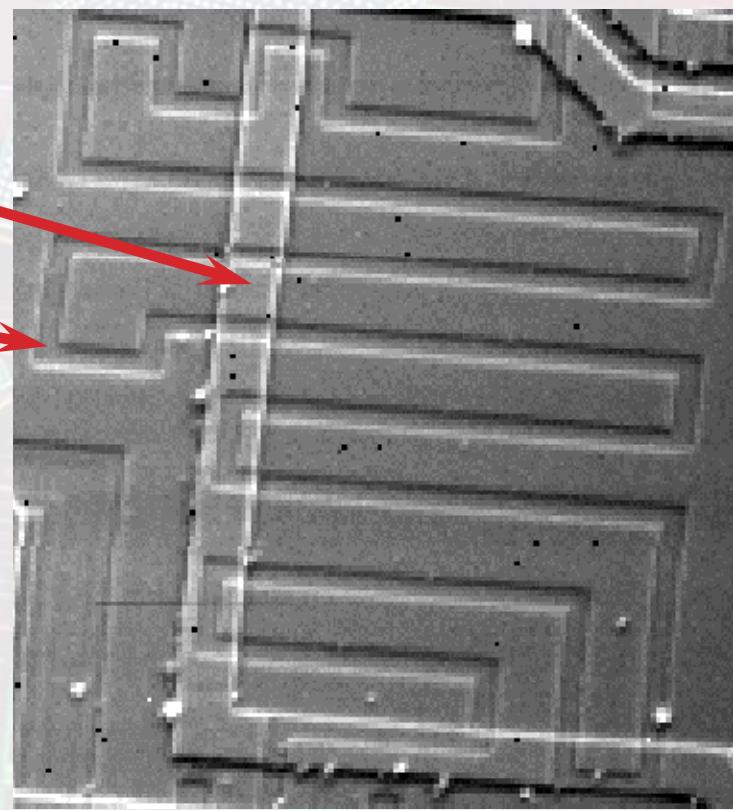




Basics

- **Minimal feature size (MFS)** – the smallest (thinnest) feature that you can create on the top of a silicon wafer (substrate)

Metallization line ("wire")
Doped region ("diffusion line")
MFS in the early days: 15-20 μm
MFS today: 7-10 nm or even less



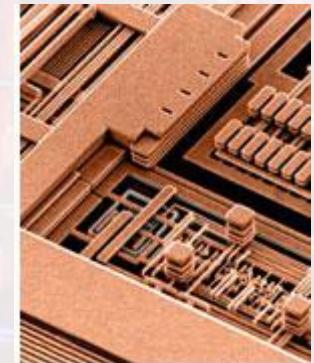
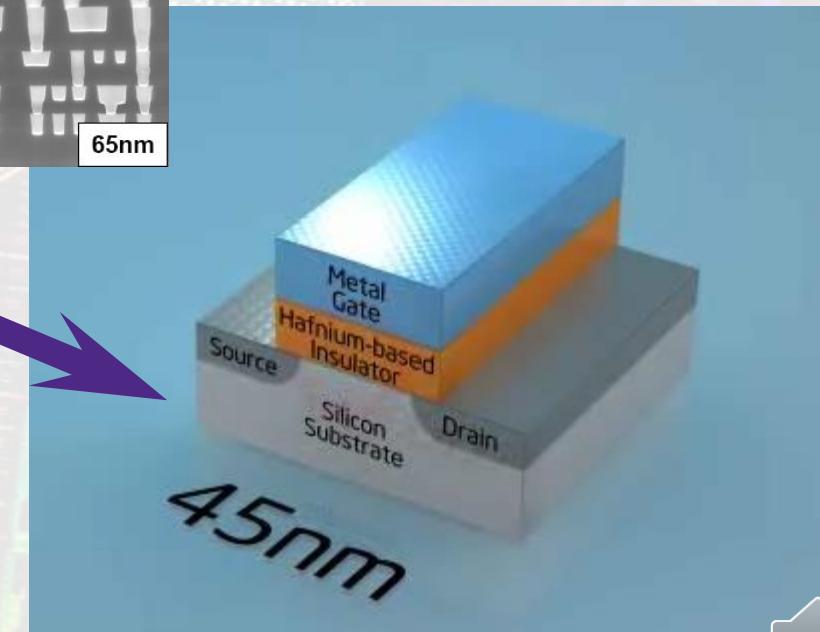
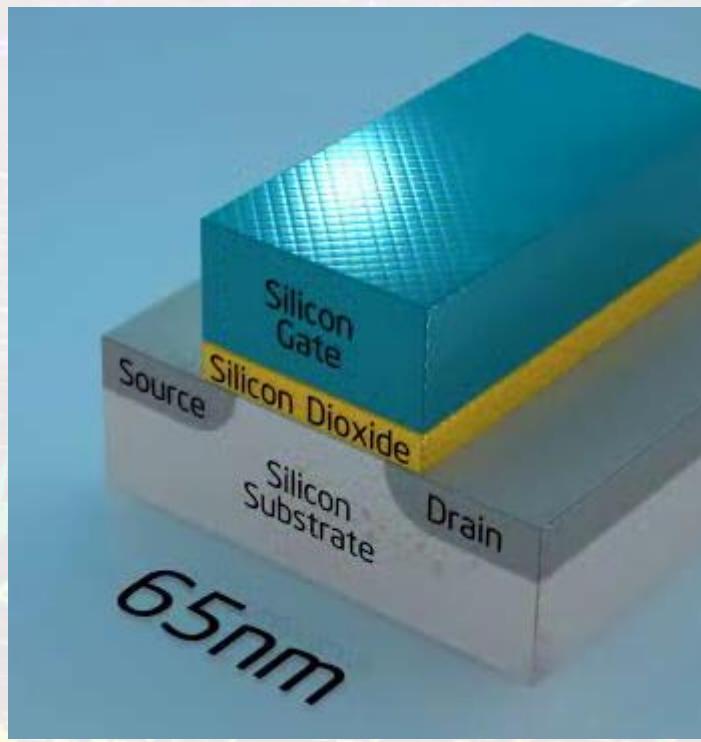
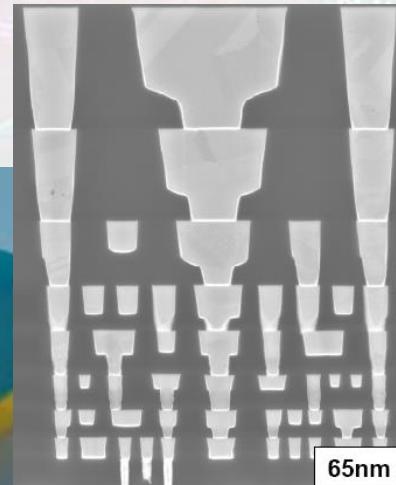
SEM microphotograph





Basics

- Minimal feature size (MFS) 2007/2008, Intel:



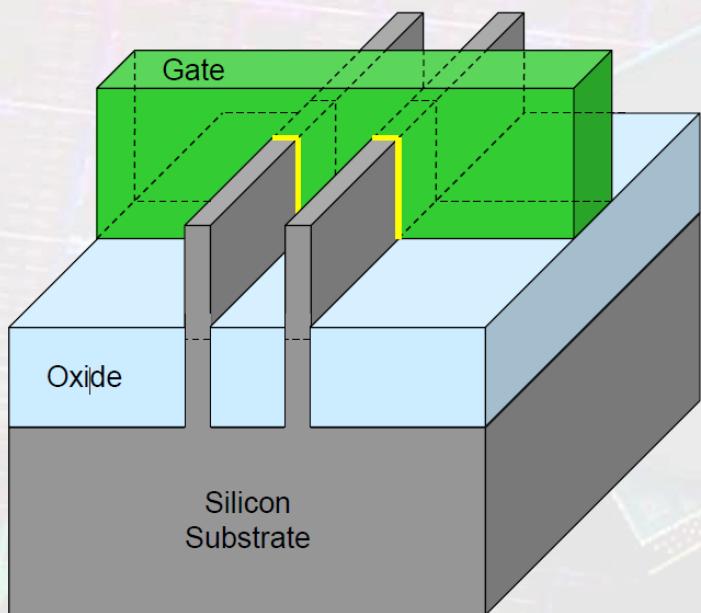
Many metal interconnects instead of polysilicon



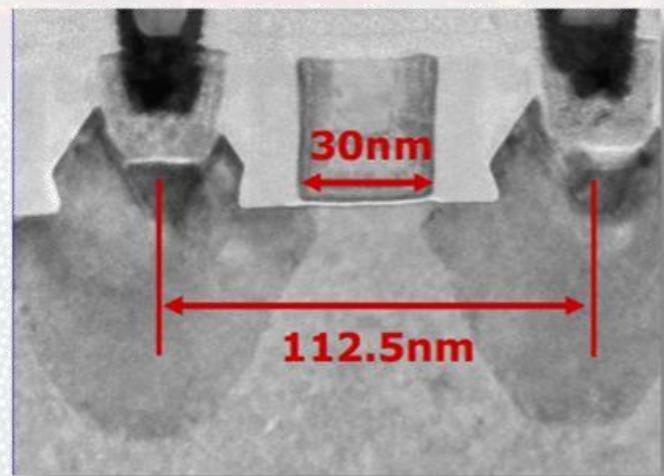


Basics

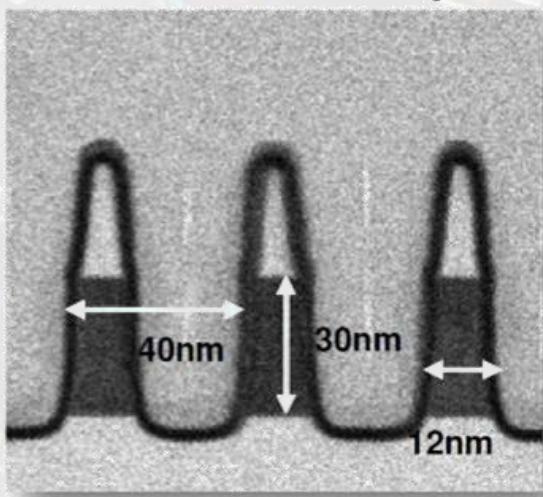
- **Minimal feature size (MFS)**
Intel 2012, 2014, 2018 (10nm)



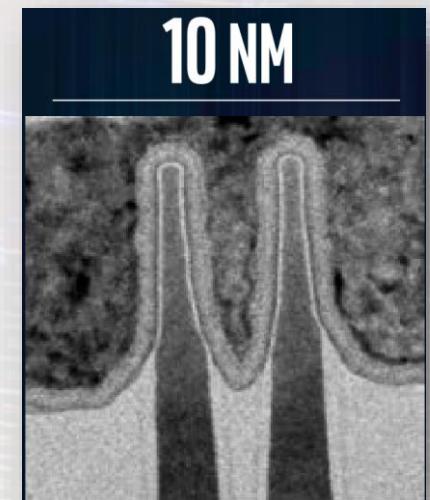
32nm Planar Example



22nm FinFET Example



10 NM

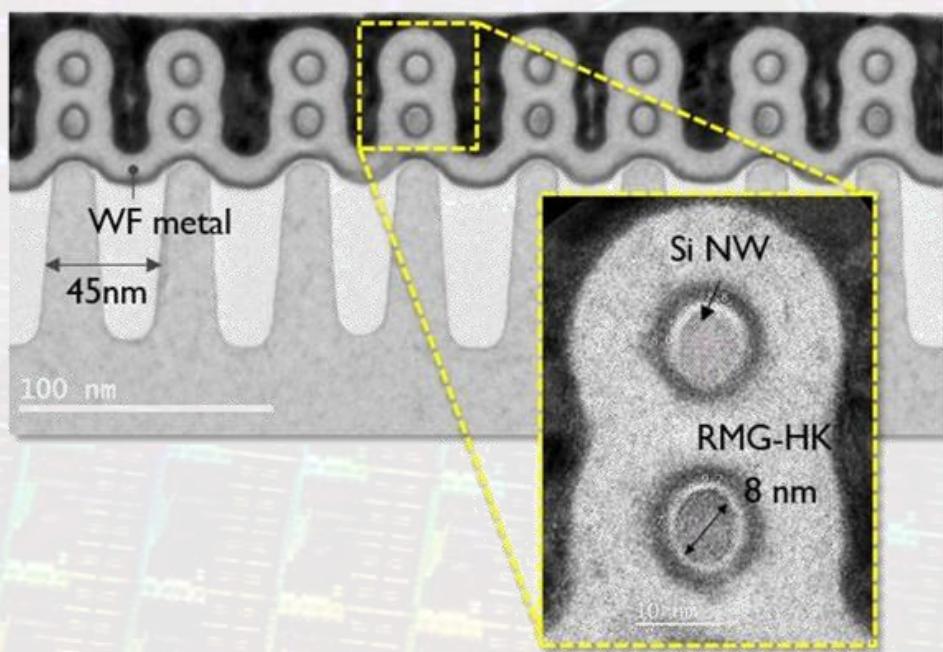




Basics

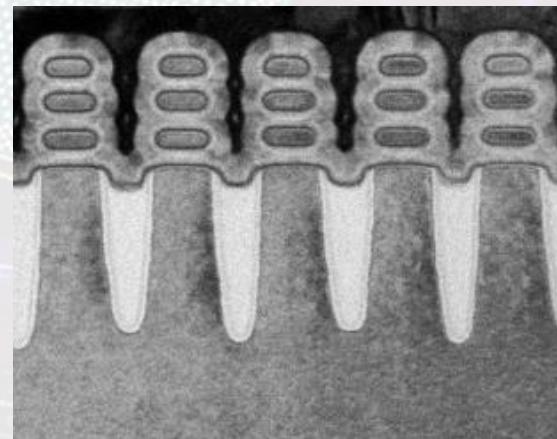
- **Minimal feature size (MFS)**

GAA IMEC demonstration
5nm (2017)



<https://semiengineering.com/going-to-gate-all-around-fets/>

GAA Samsung & IBM
5nm (2019)



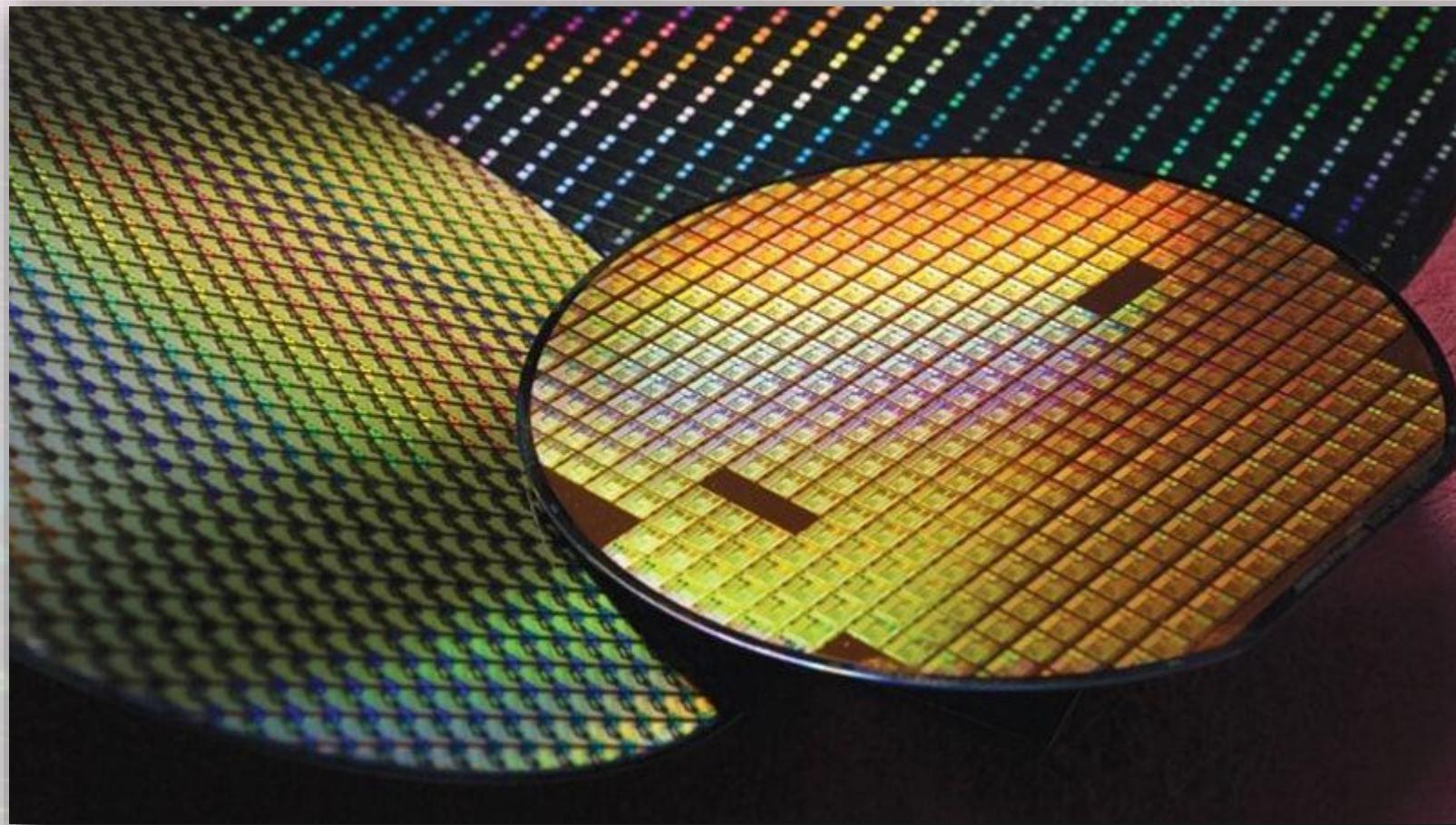
<https://www.eetimes.com/samsung-plans-3nm-gate-all-around-fets-in-2021/#>





Basics

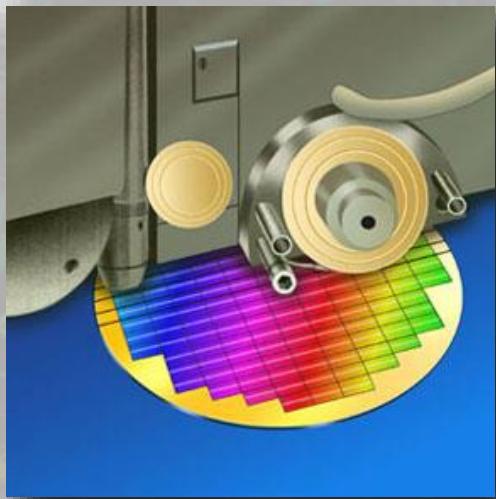
- Finished wafers before dicing



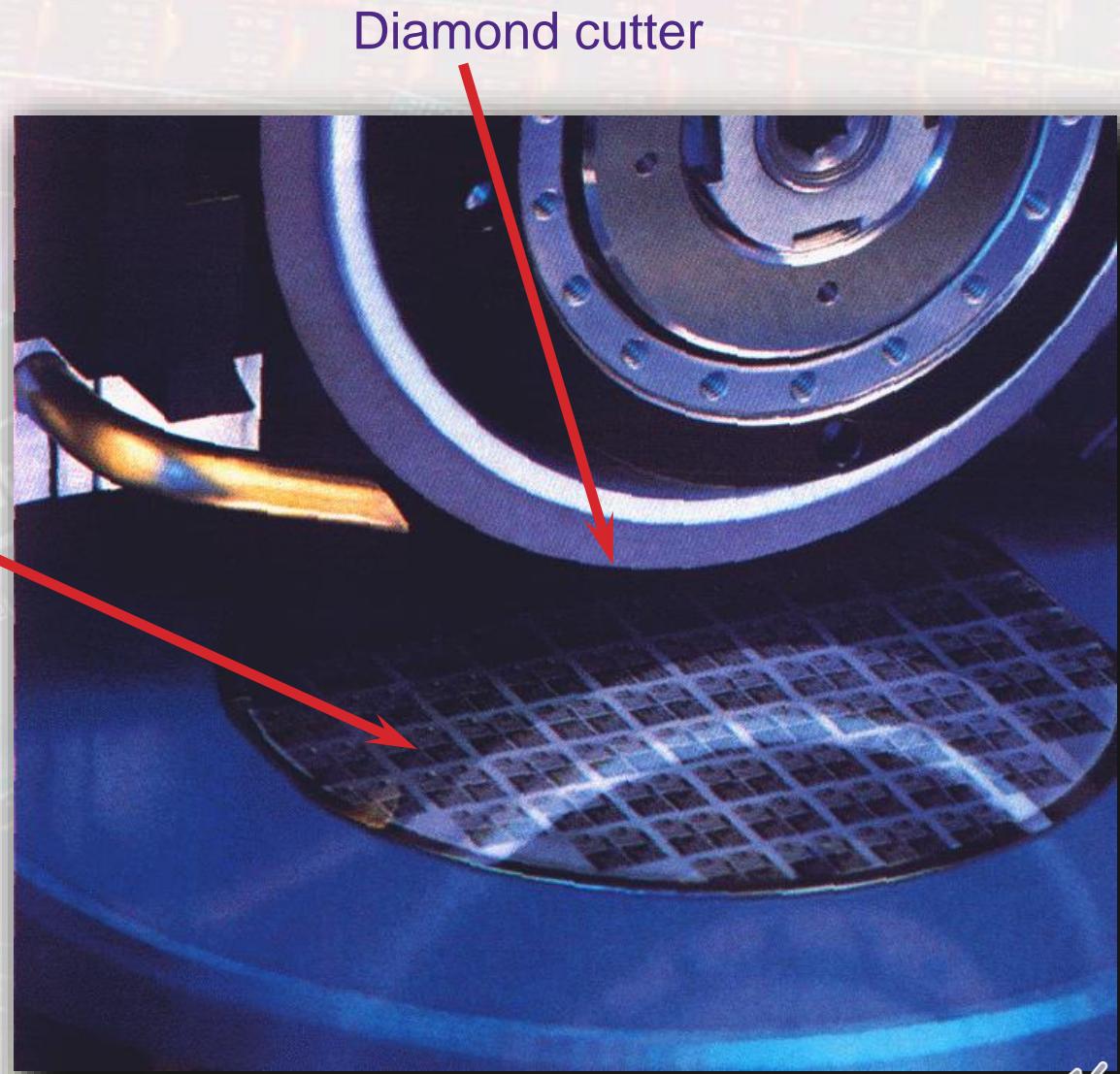


Basics

- Dicing of a wafer



A finished wafer with dice (chips) on top





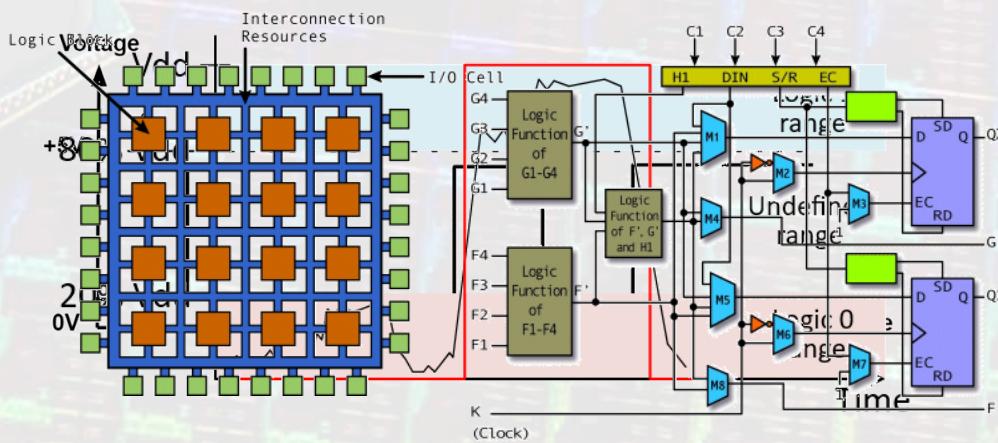
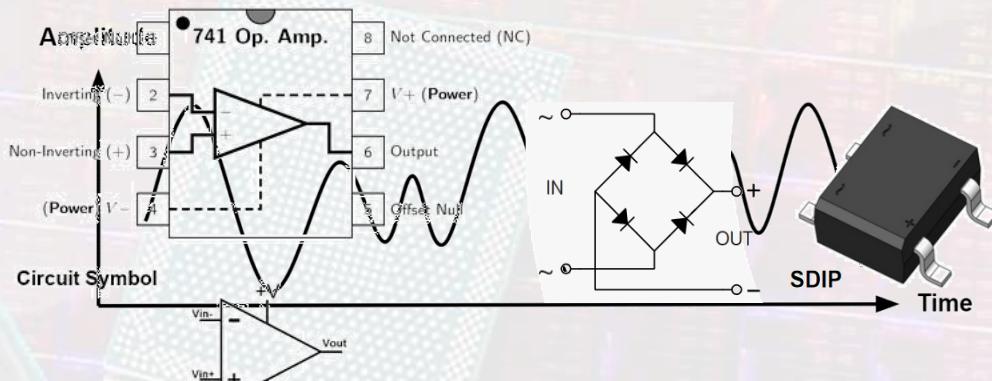
Basics

Analog integrated circuits

- Analog signal is time-varying and generally bound to a range and there is an infinite number of values within that continuous range.
- Operation amplifiers, converters, comparators

Digital integrated circuits

- A digital signal represents data as a sequence of discrete values
- It can only take on one value from a finite set of possible values at a given time.
- If distorted, it still can be read correctly
- CPUs, GPUs, microcontrollers, DSPs, FPGAs

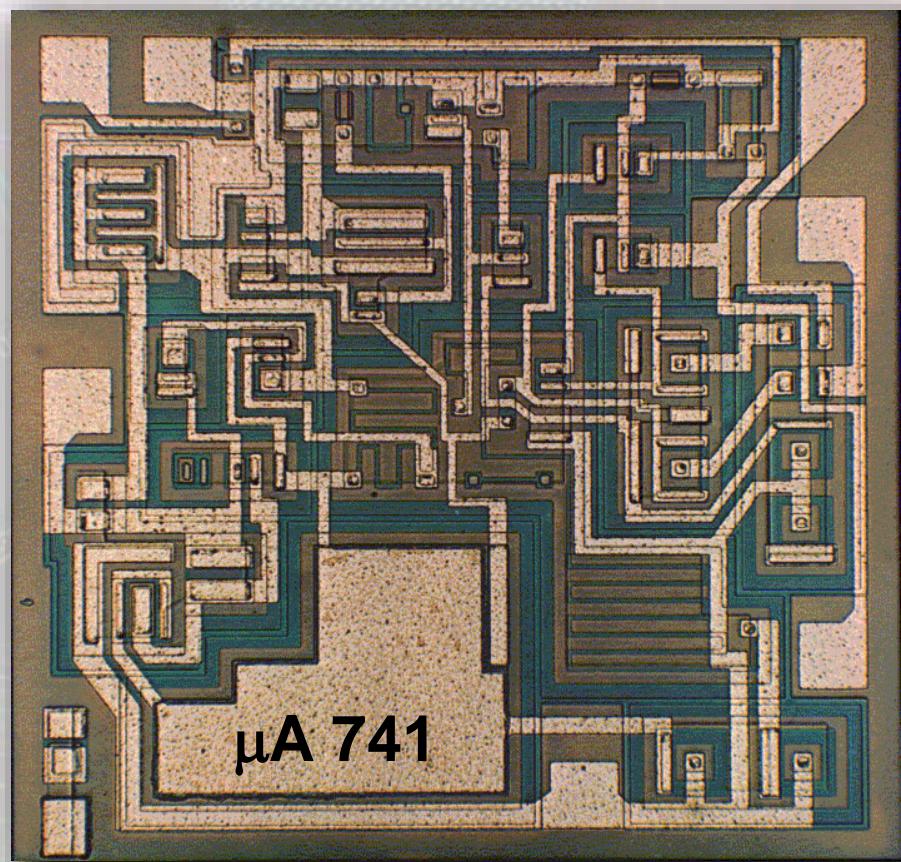
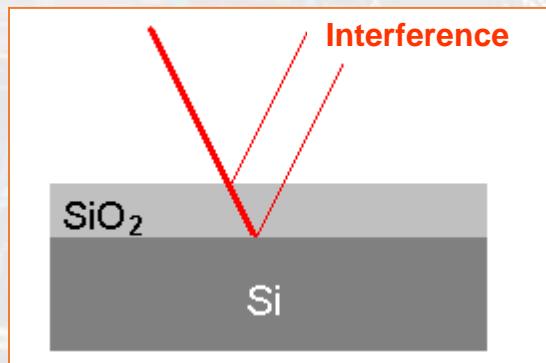




Basics

- A simple analog IC – microphotograph through an optical microscope

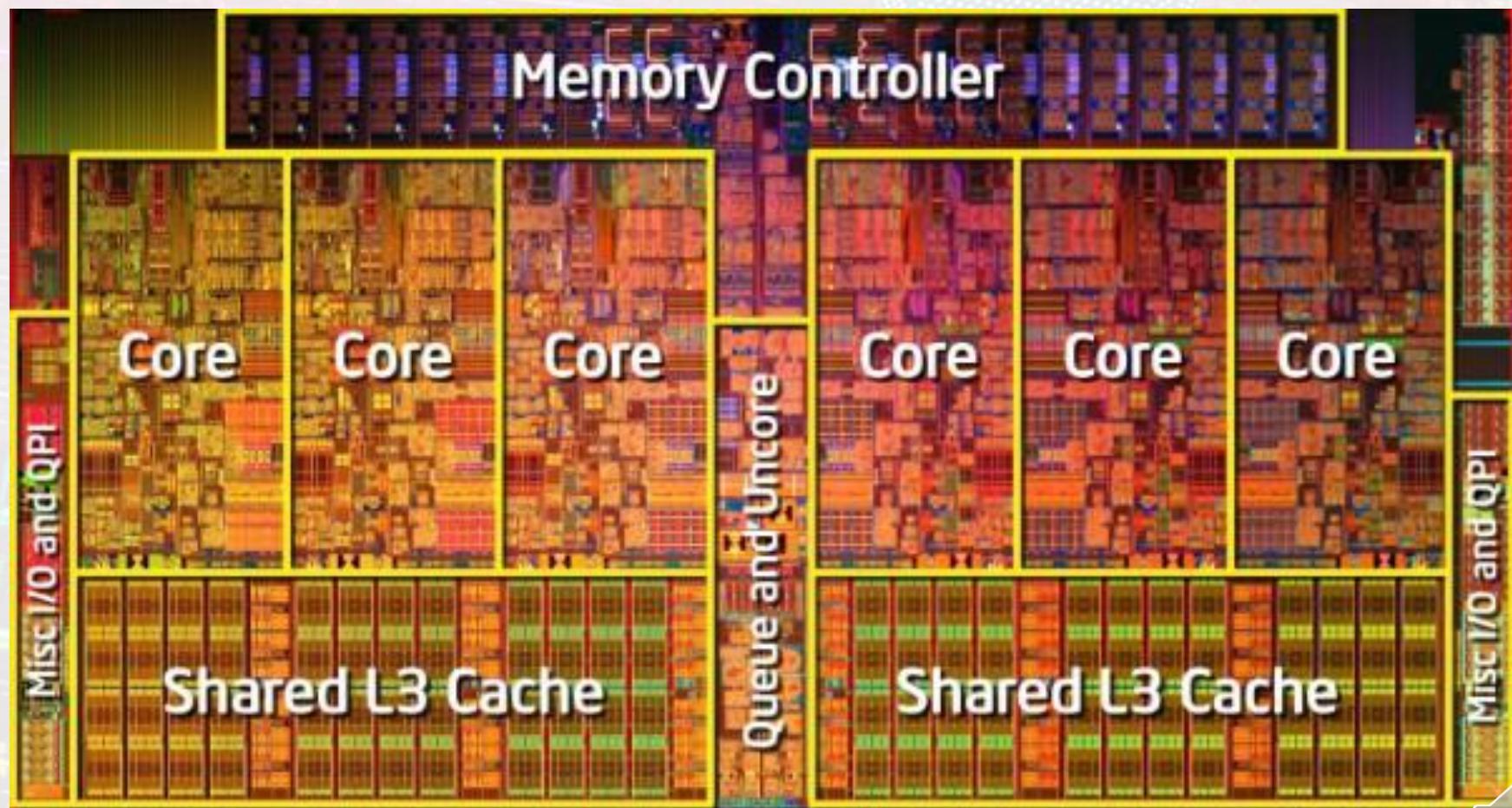
Features seem in different colors due to interference through SiO_2 layers of different thickness:





Basics

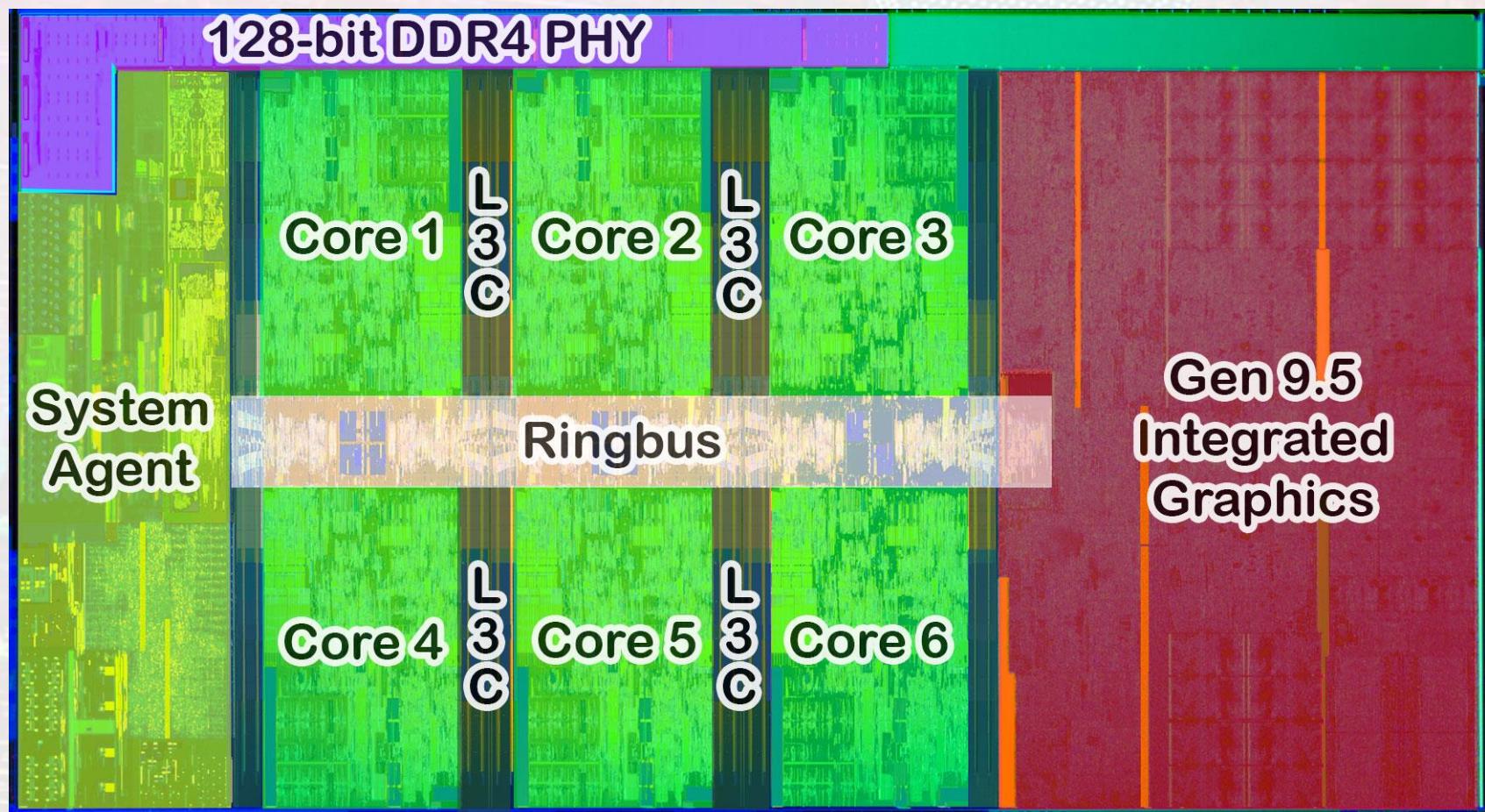
- Intel Xeon 5600, 434 mm²





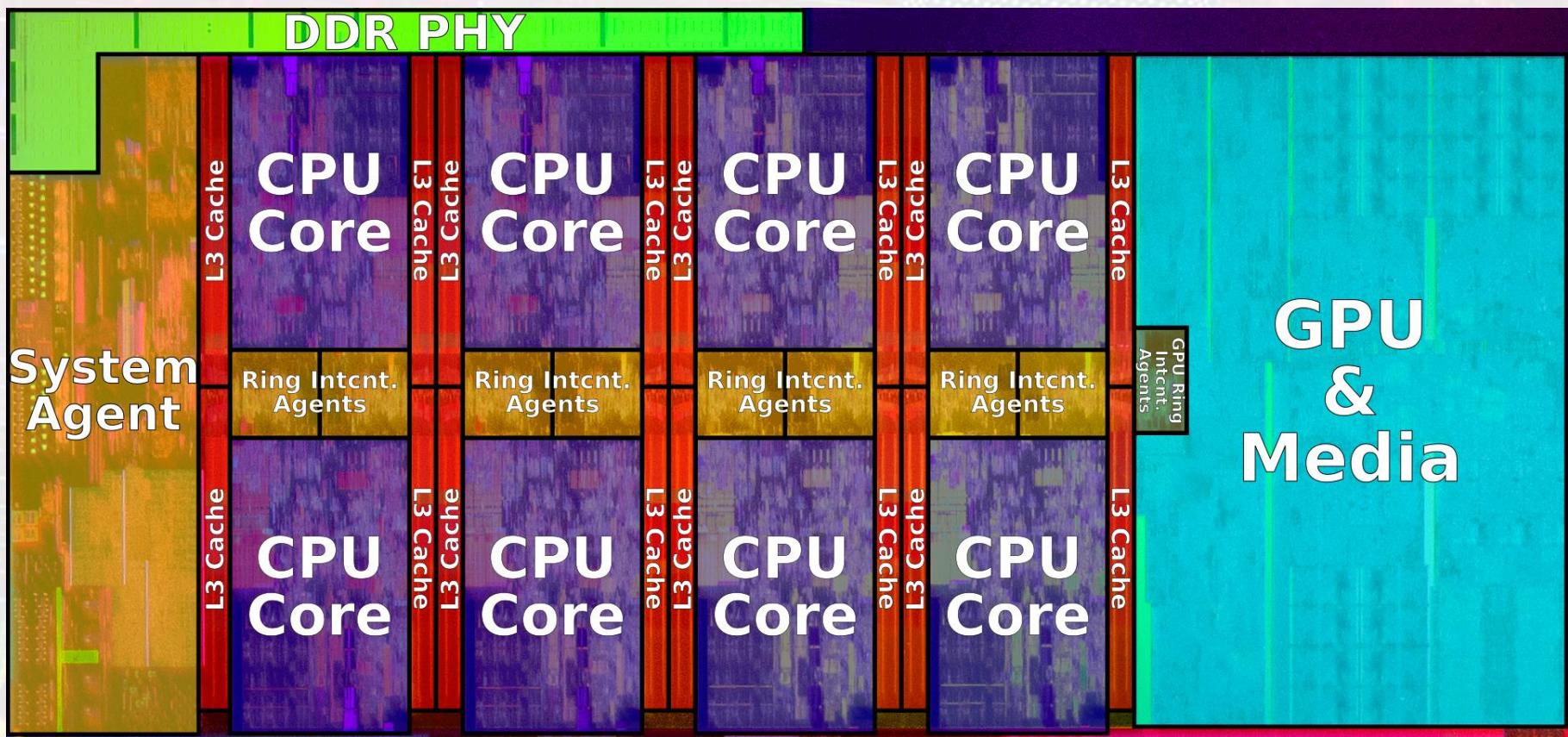
Basics

- Intel Core i5-8400, 150 mm²



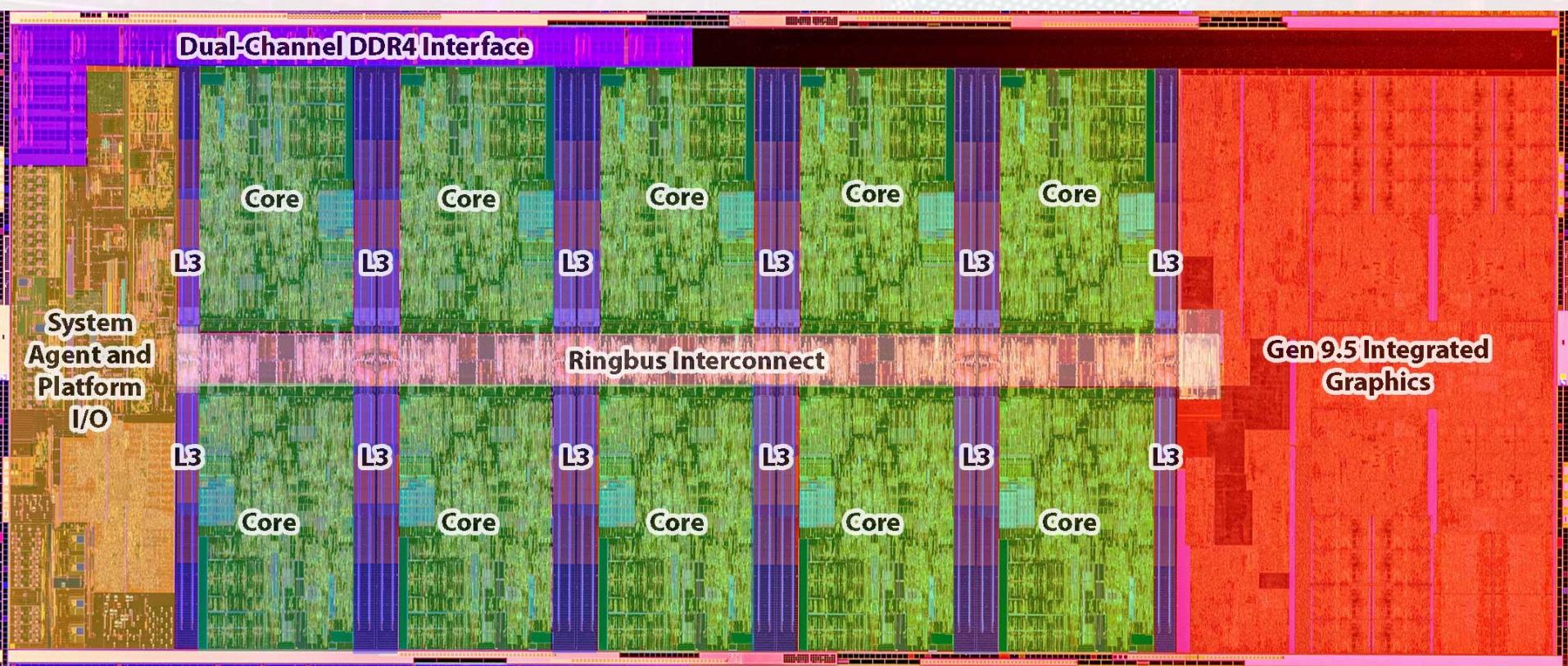
Basics

- Intel Core i9-9900K, ~177 mm² (unknown)



Basics

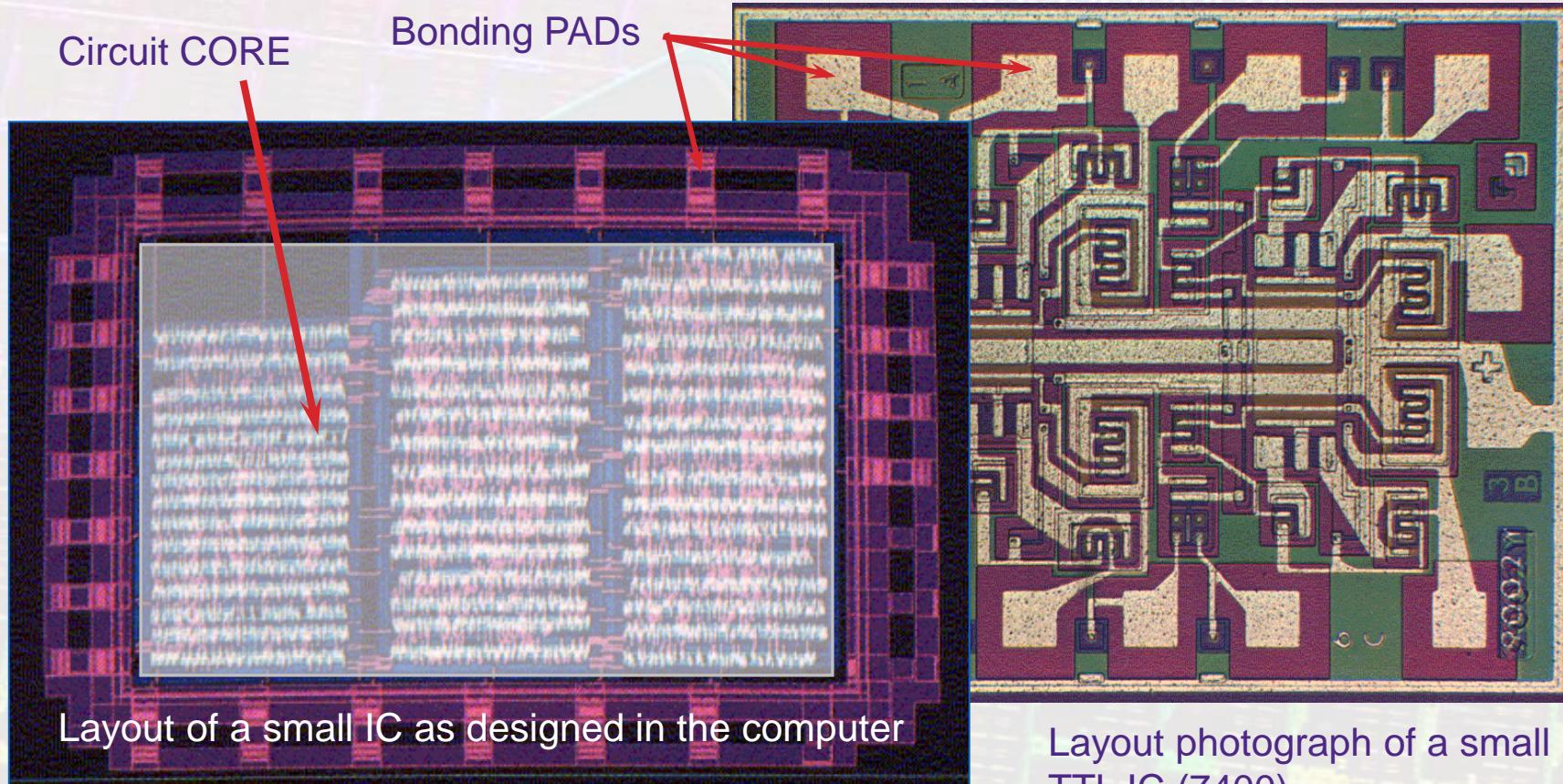
- Intel Core i9-10900K, ~198.4 mm²





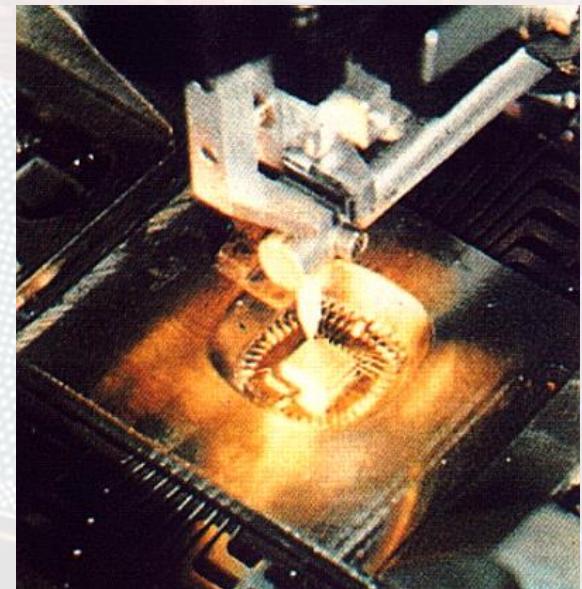
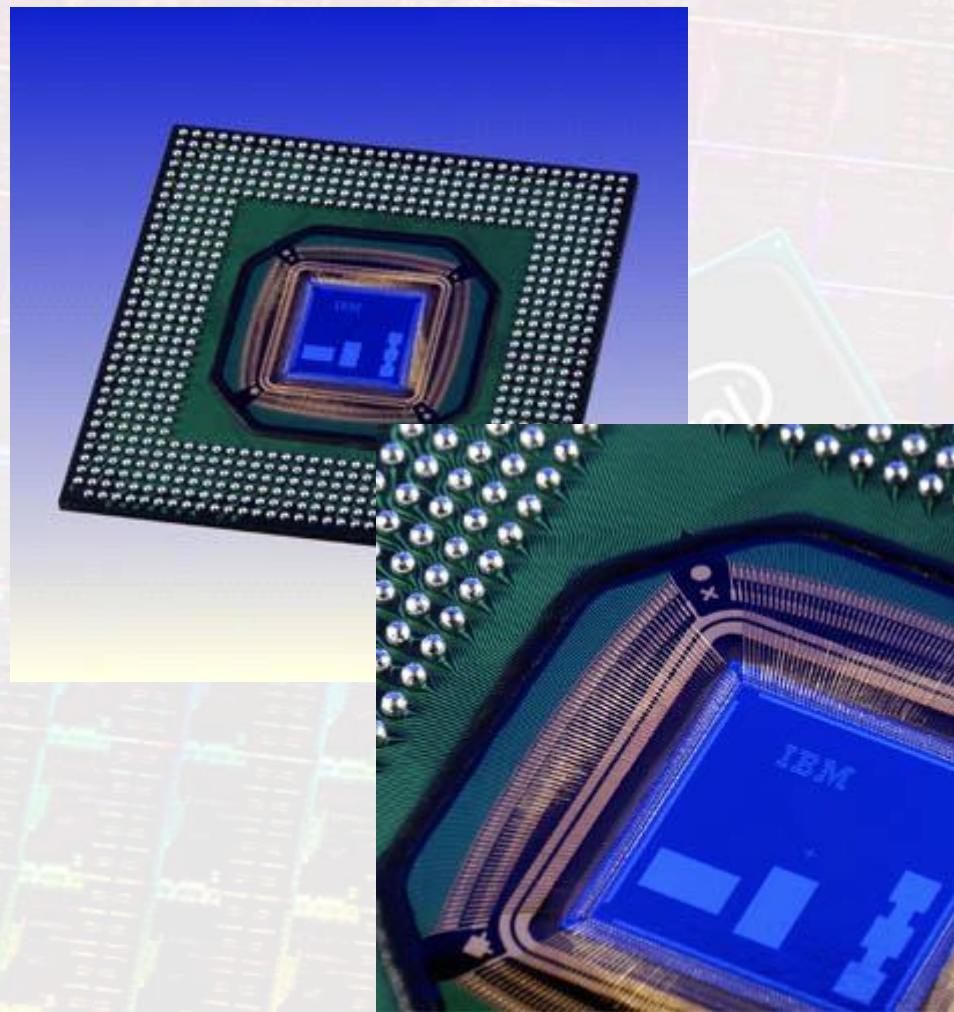
Basics

- **Layout:** collection of all features on all material layers (and in-depth doped ones) which form the IC





Packaging: a great challenge today



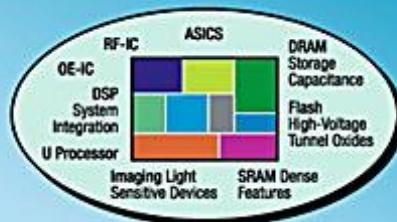
Many connections – fine *pitch*

- *high frequency properties*
- *thermal properties*

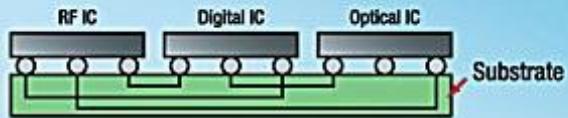


Modern packaging – 3D integration

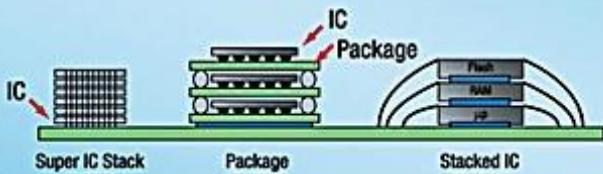
SOC
Complete system on one chip



MCM
(Multi-Chip Module)
Interconnected components

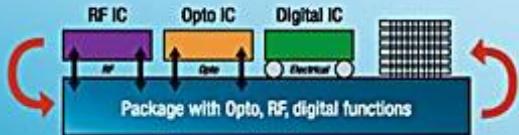


SIP
Stacked chip/package
for reduced form factors



SOP

- Optimizes functions between ICs and package
- Miniaturizes systems



ASIC

DRAM

Si interposer

50µm pitch

Microbump

Top Chip

Cu Wired

Bottom Chip

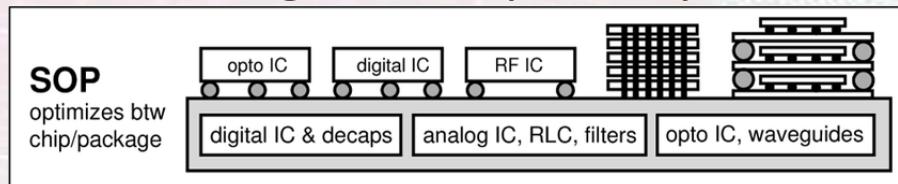
Substrate



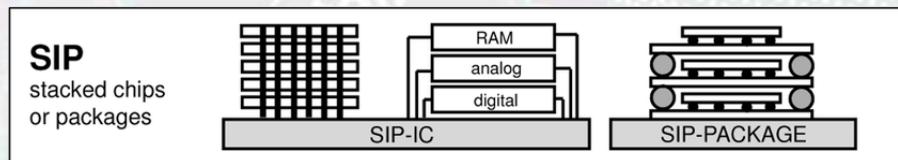


Packaging – one of the big challenges

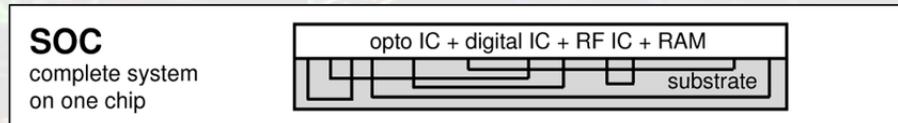
- SoP - System on Package - Complete system integration



- SiP - System in Package - is a number of integrated circuits enclosed in a single module (package)



- SoC - System on Chip - integrates all components of a computer or other electronic system into a single chip.



- LoC - is a device that integrates one or several laboratory functions





The silicon fab

- Cleanroom
- Some process steps





The cleanroom



Tiny structures are created on the surface of the wafers: high level of cleanliness is required!

Special suit – like for astronauts

Special room: **cleanroom**; much cleaner than the operation room of a hospital



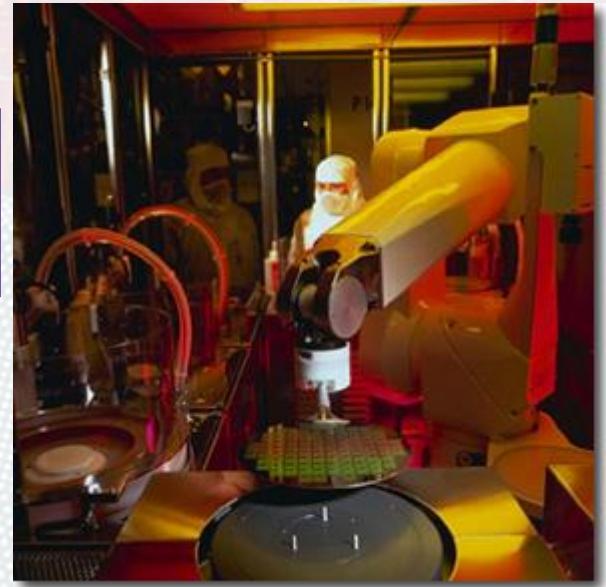


The cleanroom

Tiny structures are created on the surface of the wafers: high level of cleanliness is required!



EET cleanroom



IBM

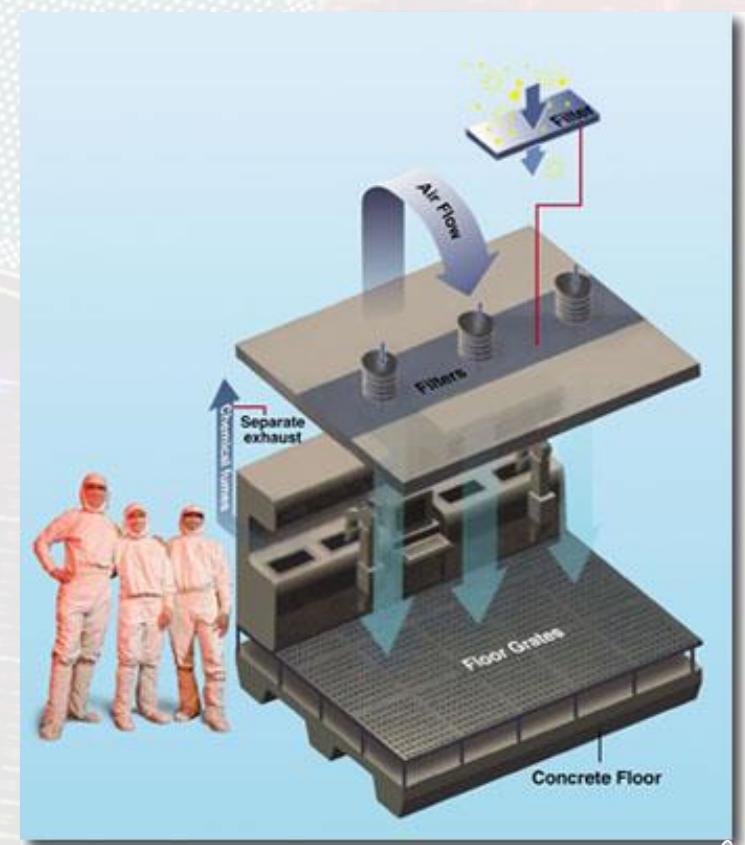
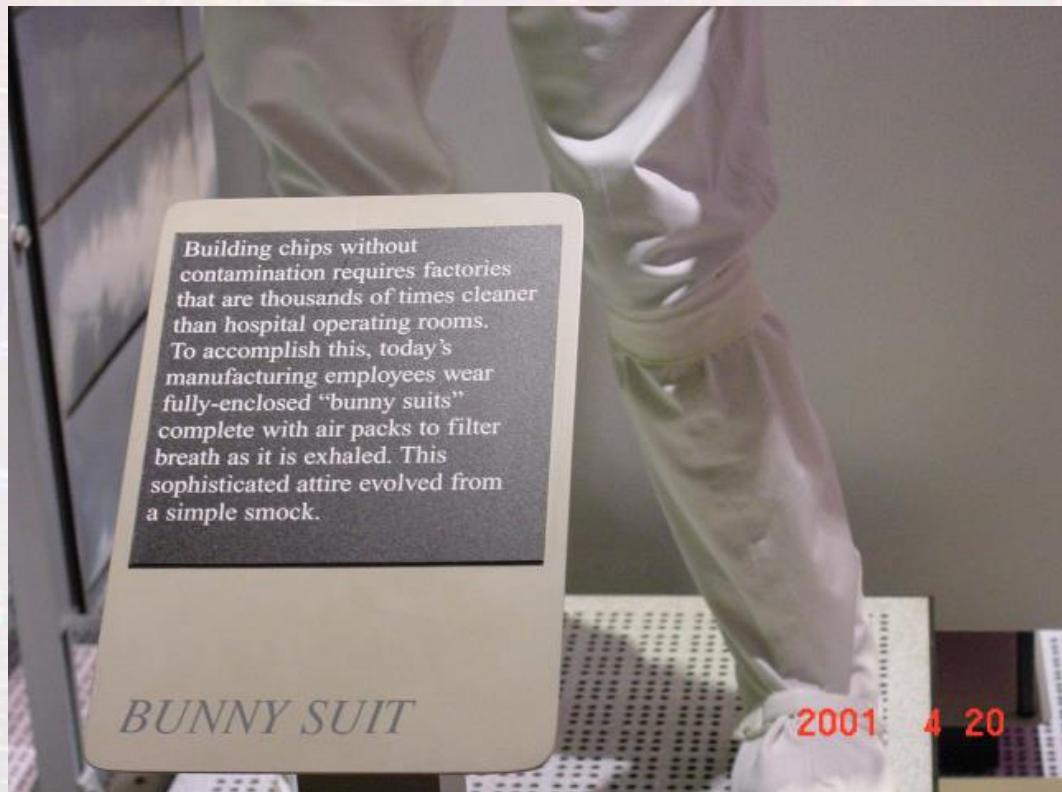


Intel Museum





The cleanroom





The cleanroom



Suiting Up

This is a typical sequence of steps that everybody who plans to enter a fab must follow:

1. Store personal items.
2. Discard any gum, candy, etc.
3. Remove any makeup with clean room soap and water.
4. Take a drink of water to wash away throat particles.
5. Cover any facial hair with a surgical mask or beard/mustache cover.
6. Put on a lint-free head cover.
7. Clean shoes with shoe cleaners.
8. Place shoe covers over street footwear.
9. Enter an air shower designed to blow off loose particles.
10. Exit air shower and enter shoe change room.
11. Clean any small, pre-approved items to be taken inside.
12. Pick up booties.
13. Sit on "dirty" side of bench.
14. Put on one bootie.
15. Swing booted foot to "clean" side of bench.
16. Put on other bootie on "dirty" side.
17. Swing booted foot to "clean" side.
18. Enter main gowning room.
19. Set aside badge, pager, and any other items to be taken inside.
20. Put on nylon gowning gloves.
21. Obtain bunny suit and belt from hanger.
22. Put on bunny suit without letting it touch the floor.
23. Put on belt.
24. Tuck bunny suit pant legs into booties.
25. Fasten snaps at top of booties.
26. Attach filter unit to belt.
27. Attach battery pack to belt.
28. Plug filter unit into battery pack.
29. Obtain helmet, safety glasses.





Cleanroom at the DED

- New semiconductor lab @ Building Q



*Semiconductor Technology
Laboratory,*

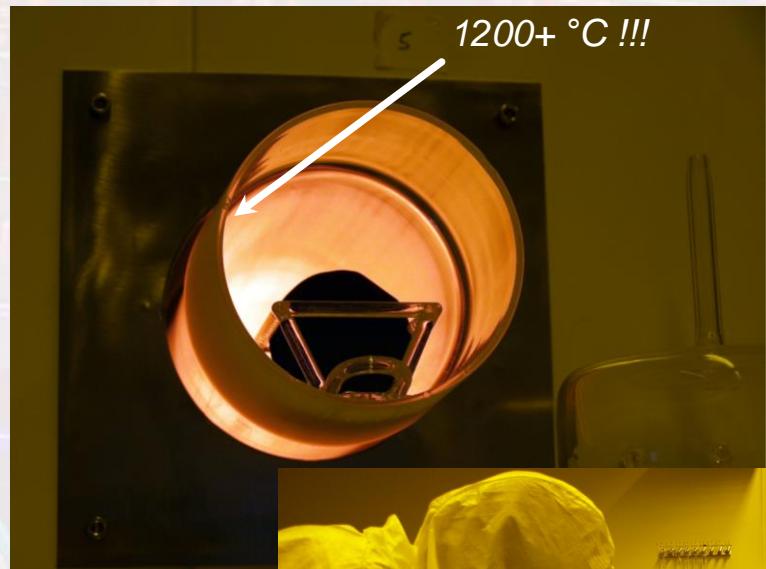
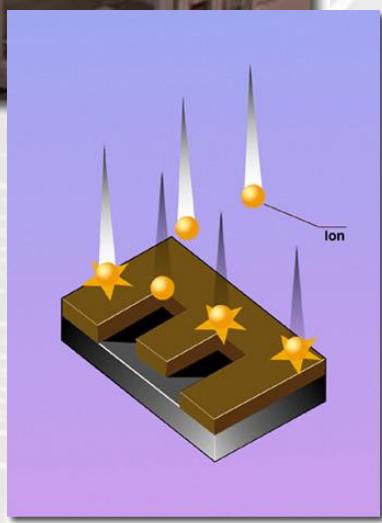
*Smart System Integration
MSc minor spec.*

Solar cells manufacturing





Ion-implanter, diffusion furnace



Semiconductor Technology Lab,
Microelectronics branch at DED

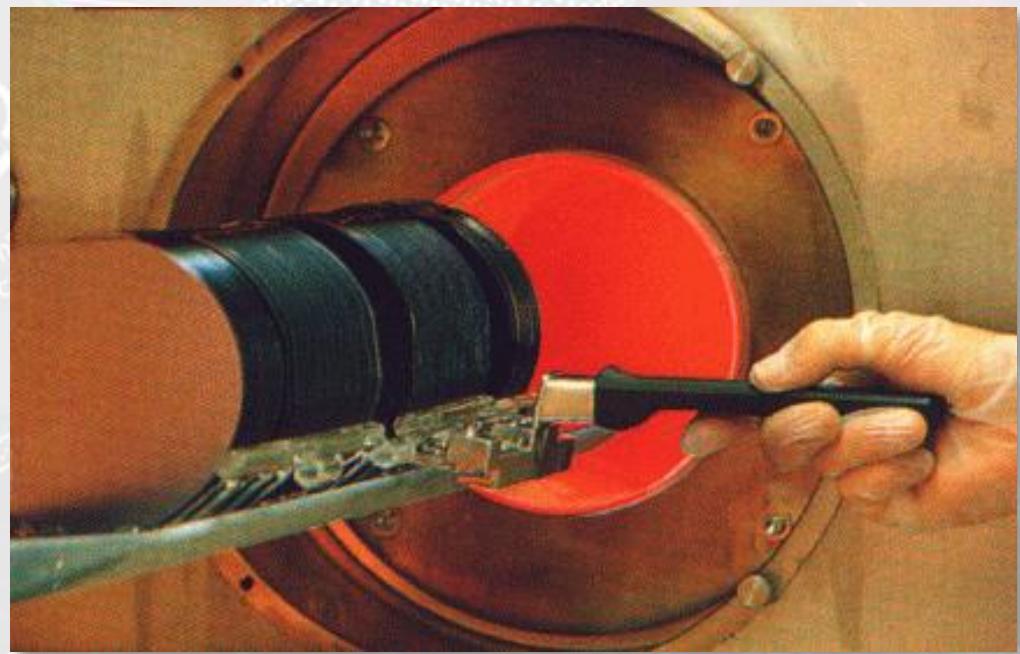




A batch

- Si wafers are processed in so called batches
- 40..100 wafers/batch, 10 000 – 50 000 chips/batch
- **Batch fabrication**

Inserting a batch into a diffusion furnace





Wafer sizes

- Today at around 30cm in diameter (8") or even 12"





An Intel *fab*





Intel fab sites - 2022

Fab name	City	Production start year	Process (wafer, node)
D1B	USA, Oregon, Hillsboro	1996	300mm, Development
RB1	USA, Oregon, Hillsboro	2001	300mm, Development
D1C	USA, Oregon, Hillsboro	2001	300mm, Development
RP1	USA, Oregon, Hillsboro	2001	300mm, Research
D1D	USA, Oregon, Hillsboro	2003	300mm, Development
D1X	USA, Oregon, Hillsboro	2013	300mm, Development
Fab 11X	USA, New Mexico, Rio Rancho	1995 upgrade 2020/2021 with 22/14	300mm, 45nm/32nm, Packaging
Fab 12	USA, Arizona, Chandler	2006	300mm, 22nm/14nm/10nm
Fab 22	USA, Arizona, Chandler	2002	300mm, 22nm/14nm/10nm
Fab 24	Ireland, Leixlip	2006	300mm, 14nm ^[2]
Fab 28a	Israel, Kiryat Gat	1996	300mm, 22nm
Fab 28	Israel, Kiryat Gat	2008	300mm, 22nm/10nm ^{[3][4]}
Fab 32	USA, Arizona, Chandler	2007	300mm, 22nm/14nm/10nm
Fab 42	USA, Arizona, Chandler	2020	300mm, 10nm/7nm (2024)
SC2	USA, California, Santa Clara		Reticle/Masks , Intel Mask Operations ^[8]





Intel fab sites - 2022

Fab name	City	Production start year	Process (wafer, node)
Fab 52	USA, Arizona, Chandler	(2024) ^[6]	300mm, 7nm
Fab 62	USA, Arizona, Chandler	(2024) ^[7]	300mm, 7nm
Fab 34	Ireland, Leixlip	2023	300mm, 7nm ^[5]
	Malaysia, Kedah, Kulim	(2024)	300mm, Packaging ^[9]





Intel fabs 2006-2008

- 65nm fabs, ~2006



- 45nm: "Fab32", 2007-



An Intel manufacturing technician uses a scanner to start the very first 45nm production lot of 300mm wafers inside of Fab 32, Intel's first high-volume 45nm chip factory in Chandler, Ariz.



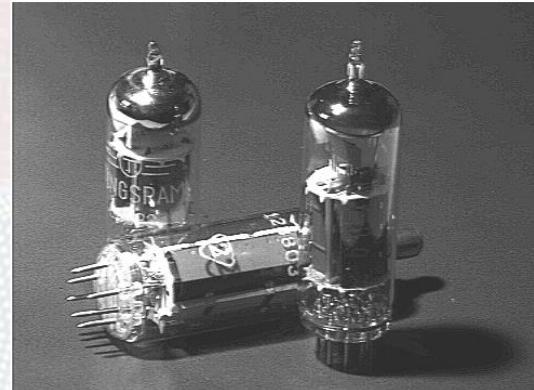
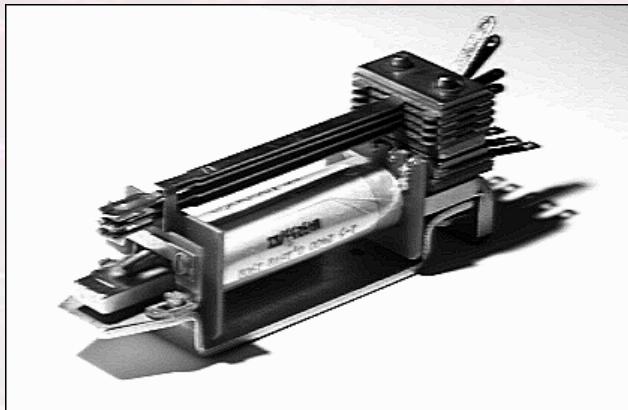


History, trends

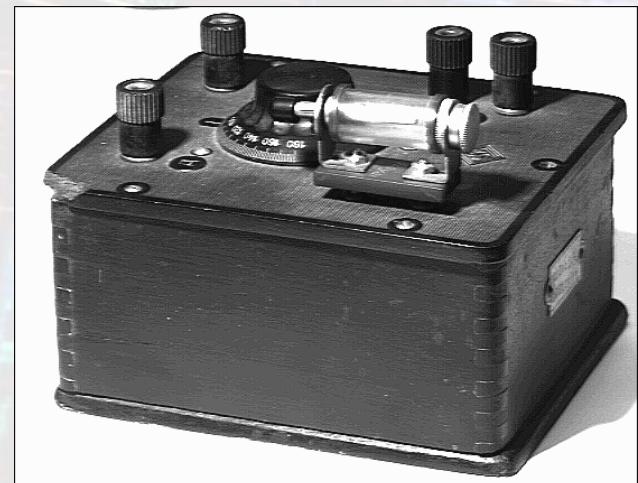
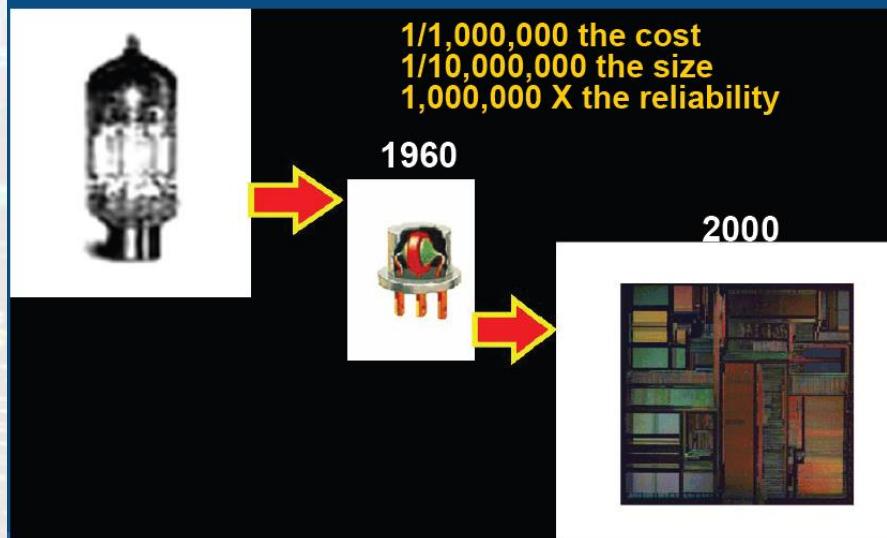


Roots

1837 Morse, telegraph



1920



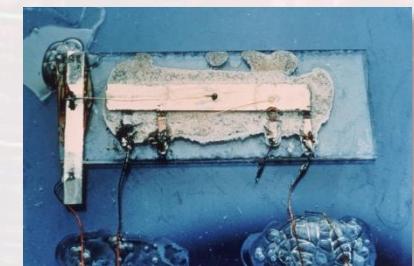
~1920 radio receiver



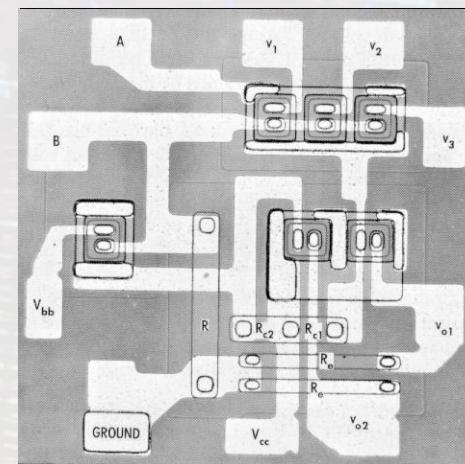
Microelectronics: fastest growing industry

Revolution of the transistors

- Transistor – Bardeen (Bell Labs), 1947
- Bipolar transistor – Schockley, 1949
- The 1st bipolar logic gate – Harris, 1956
- The 1st monolithic IC – Jack Kilby, 1959
- The 1st commercial IC with logic gates – Fairchild, 1960
- TTL – 1962..1990-ies
- ECL – 1974..1980-ies



You can make an IC of such a "complexity" in the cleanroom of the Department





Microelectronics: fastest growing industry

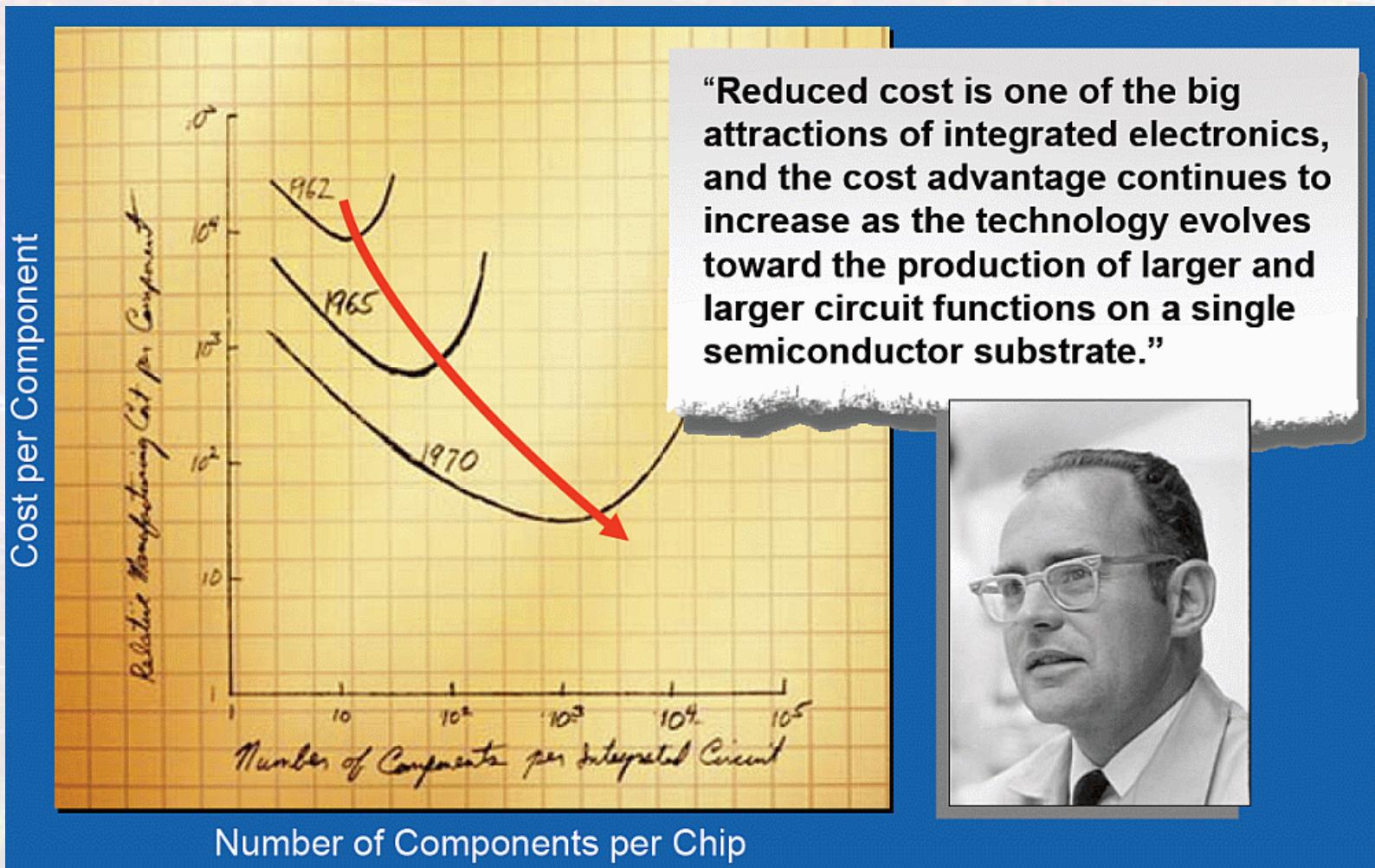
Development of the MOSFETs:

- MOSFET transistor – Lilienfeld (Canada, 1925) and Heil (England, 1935)
- CMOS – 1960s, stack due to manufacturing problems
- pMOS processes in the 1960s
- nMOS processes in the 1970s (4004, 8080)
- CMOS processes from the 1980s – preferred MOSFET process due to low power consumption
- Nowadays e.g.:
 - BiCMOS, GaAs, SiGe – for very high frequency circuits
 - SOI, Cu metallization, *low-K* dielectrics





Gordon Moore, 1965:





Microelectronics: fastest growing industry

Moore's law

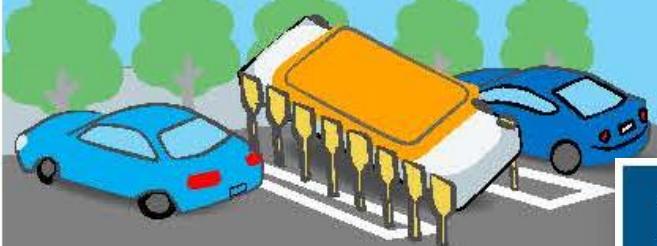
- In 1965 Gordon Moore predicted that in every 14..18 months the number of transistors integrated in a chip will double (exponential growth)
- This prediction is valid even today.
- The 1 million transistors/chip threshold was reached in the 1980s
 - 2300 transistors, 1 MHz clock frequency (Intel 4040) - 1971
 - 16 million transistors (UltraSPARC III)
 - 42 million transistors, 2 GHz clock frequency (Intel P4) - 2001
 - 140 million transistors, (HP PA-8500) – 1998
 - ~3 billion transistors, (i7-9900K) - 2018
- *More than Moore*: further increase of integration density, e.g. 3D stacking of chips (RAM-s, pen drives)





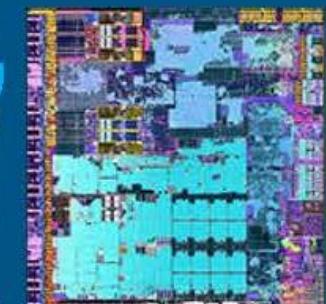
Moore's law

IF AN INTEL-BASED ANDROID** PHONE WERE BUILT USING 1971 TECHNOLOGY, THE PHONE'S MICROPROCESSOR ALONE WOULD BE THE SIZE OF A PARKING SPACE.



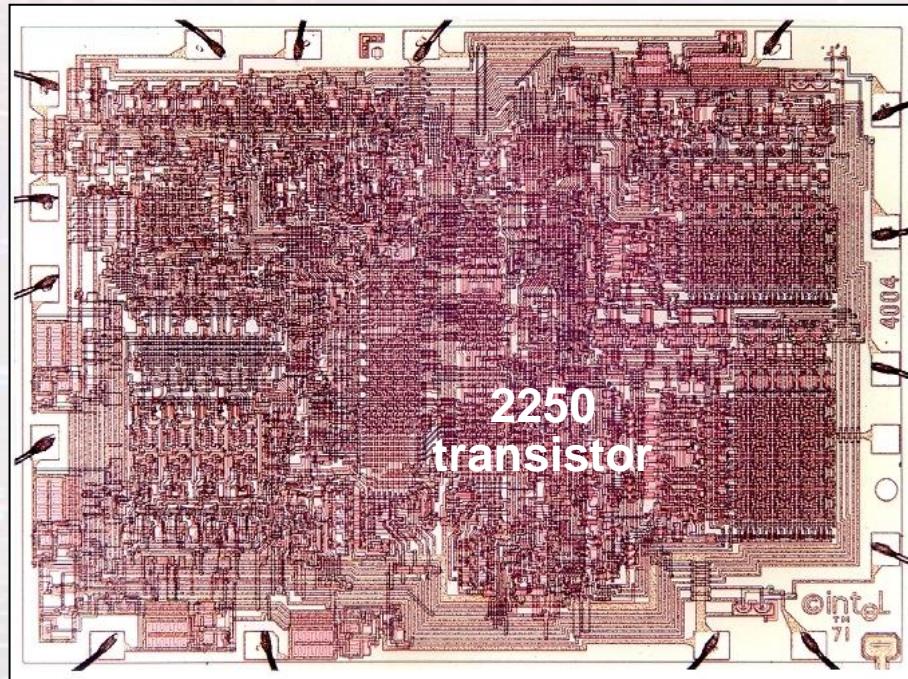
- ▶ Result of 50 years of progress presented through a modern CPU and a processor of a modern smart phone

COMPARED TO INTEL'S FIRST MICROPROCESSOR, THE INTEL® 4004, TODAY'S 14NM PROCESSORS DELIVER 3,500 TIMES THE PERFORMANCE, AT 90,000 TIMES THE EFFICIENCY AND AT 1/ 60,000TH THE COST.





Microelectronics: fastest growing industry



1971 -Intel 4040

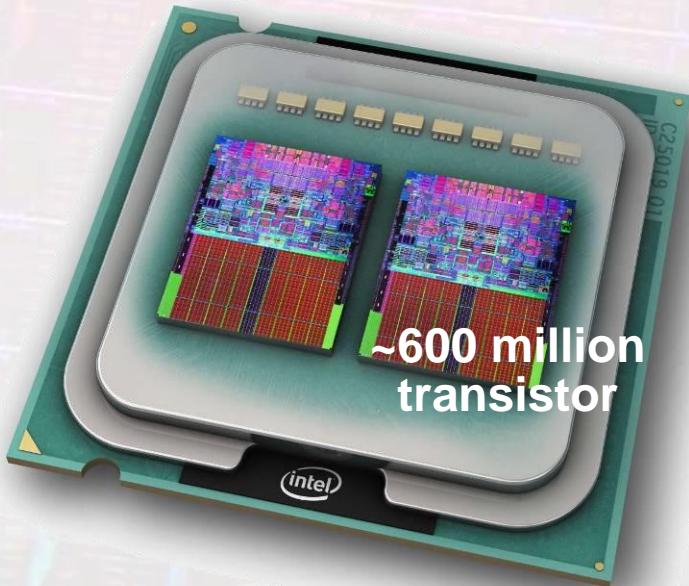


2002 - Intel Pentium IV

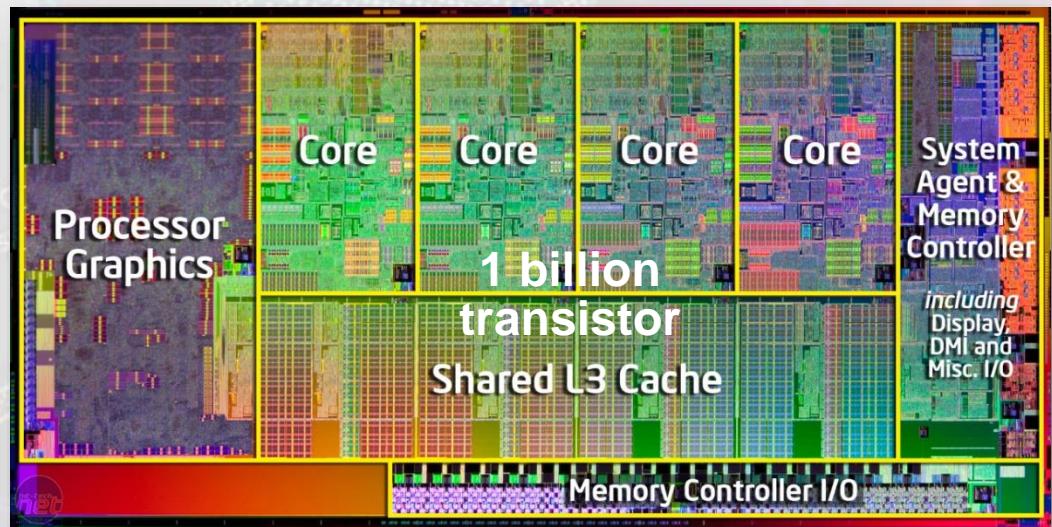




Microelectronics: fastest growing industry



2007 - Intel Core2 EE



2012 - Core i7





Microelectronics: fastest growing industry

Moore's Law: circa 2008

Intel® Atom™ - dual-core

47 Millions Transistors
45nm node
Hi-k Metal Gate
193 dry Litho

Rice – single grain

In 2014, on 14nm technology, the above chip would be 1/8 the size
- Much smaller than the grain of rice!





Prognosis: roadmap

The screenshot shows a web browser window for the International Technology Roadmap for Semiconductors (ITRS) website at itrs2.net. The page features a navigation bar with links to HOME, ITRS NEWS, ITRS REPORTS, ITRS SCHEDULE, ITRS MODELS AND PAPERS, and IRC/FOCUS TEAMS/ITWG INFORMATION. The main content area displays the ITRS 2.0 logo, followed by a section titled "ITRS SPONSORS" listing five industry associations:

- ESIA** European Semiconductor Industry Association
- JEITA** Japan Electronics and Information Technology Industries Association
- KSIA KOREA SEMICONDUCTOR INDUSTRY ASSOCIATION** Korean Semiconductor Industry Association
- SIA SEMICONDUCTOR INDUSTRY ASSOCIATION** Semiconductor Industry Association
- TSIA** Taiwan Semiconductor Industry Association





Recent – company webpages

2016-17 CCG Mobile Product Roadmap

Schedule represents front-end of [RTS](#)

BDW = Broadwell	BSW = Braswell
SKL = Skylake	APL = Apollo Lake
KBL = Kaby Lake	GLK = Gemini Lake
CNL = Cannon Lake	CHT = Cherry Trail
CFL = Coffee Lake	

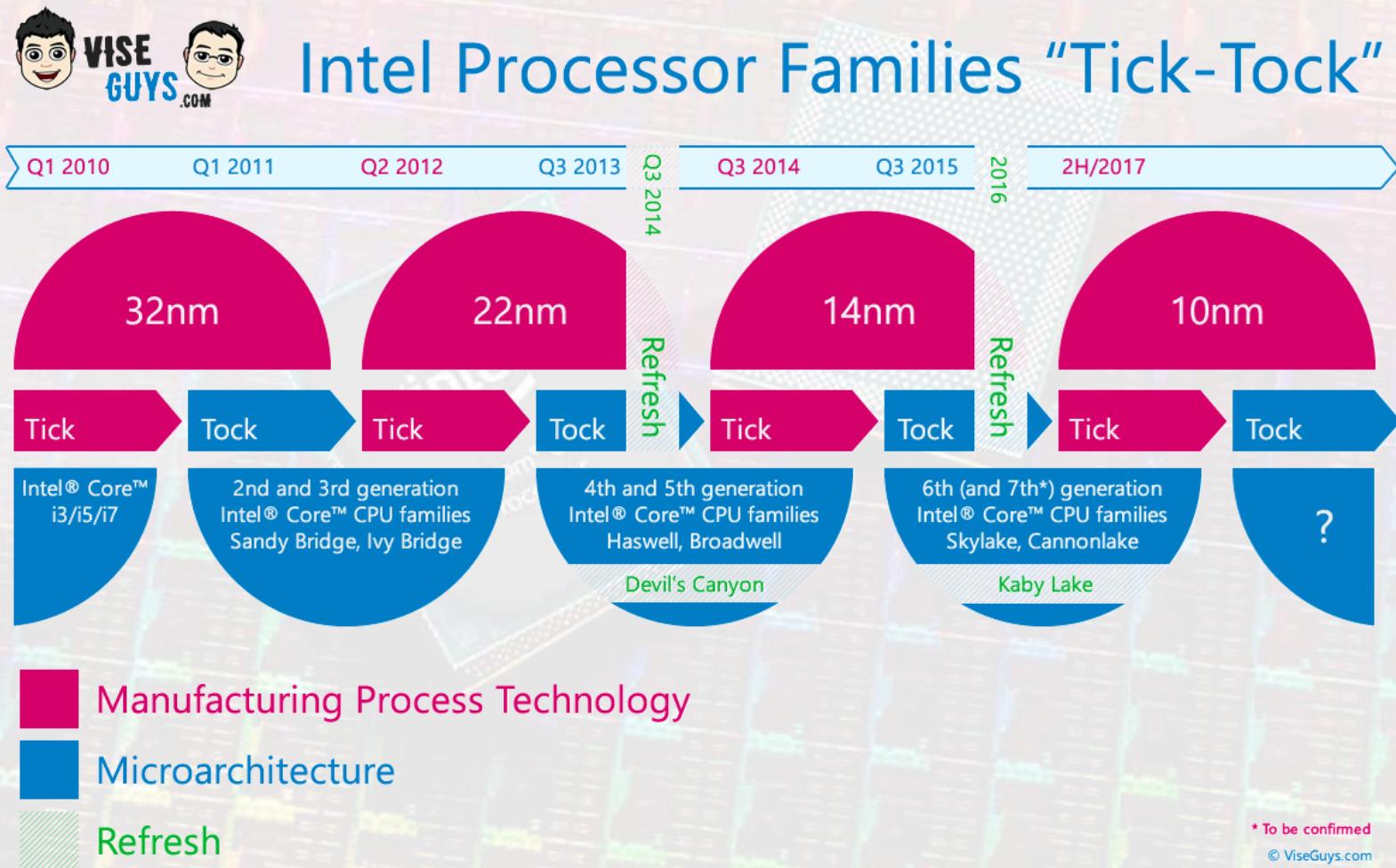


	Q2'16	Q3'16	Q4'16	Q1'17	Q2'17	Q3'17	Q4'17	Q1'18	Q2'18
H Processor	Skylake 45W, 2-chip BGA								QC GT4e CFL 6C GT2
	Skylake 45W, 2-chip BGA	QC GT2		Kaby Lake 45W, 2-chip BGA				QC GT2	45W
U Processor	Skylake 15W/28W, SoC BGA		GT3e	Kaby Lake 15W/28W, SoC BGA				GT3e CFL 15W/28W	QC GT3e
	Skylake 15W SOC BGA GT2			Kaby Lake 15W, SoC BGA			GT2	CNL 15W, SoC BGA	GT2
Y Processor	Skylake 4.5W, SoC BGA	GT2		Kaby Lake 4.5W, SoC BGA			GT2	CNL 5.2W, SoC BGA	GT2
N Processor	BSW 4W/6W, SoC BGA	QC	Apollo Lake 4W/6W, SoC BGA				QC	GLK 4W/6W,Soc BGA	QC



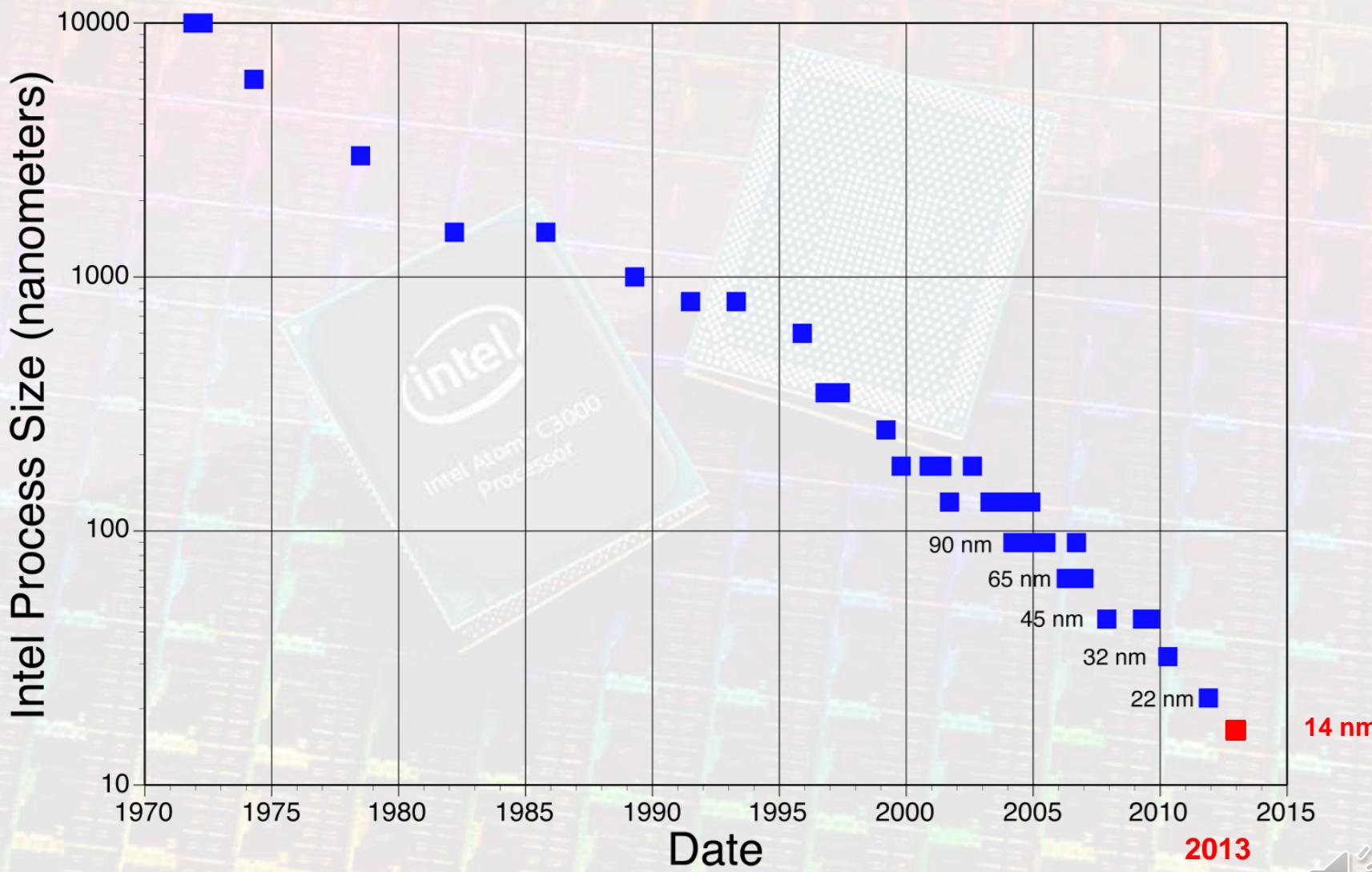


Recent – company webpages



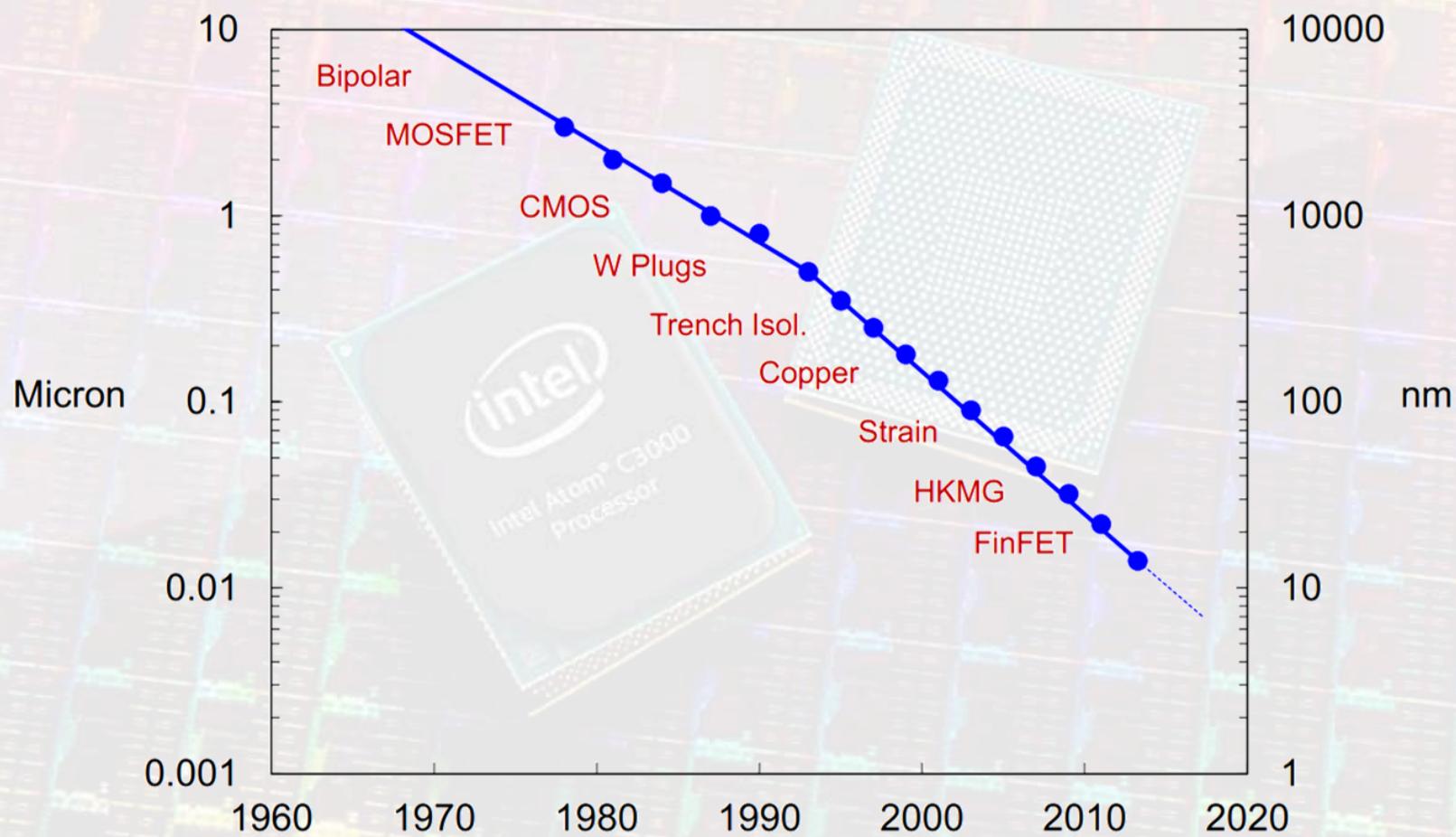


Minimal feature size – trend (Intel)





Minimal feature size – trend (Intel)



Process/device innovation has always been an indispensable part of scaling





Toplist of processors in 2015

Intel HEDT Flagship Processors (Gen vs Gen Specifications Comparison):

Intel HEDT Family	Gulftown	Sandy Bridge-E	Ivy Bridge-E	Haswell-E	Broadwell-E	Skylake-E
Process Node	32nm	32nm	22nm	22nm	14nm	14nm
Flagship SKU	Core i7-980X	Core i7-3960X	Core i7-4960X	Core i7-5960X	Core i7-6950X	Core i7-7970X (TBA)
Max Cores/Threads	6/12	6/12	6/12	8/16	10/20	TBD
Clock Speeds	3.33/3.60 GHz	3.30/3.90 GHz	3.60/4.00 GHz	3.00/3.50 GHz	3.00 GHz / TBD	TBD
Max Cache	12 MB L3	15 MB L3	15 MB L3	20 MB L3	25 MB L3	TBD
Max PCI-Express Lanes	32 Gen2	40 Gen2	40 Gen3	40 Gen3	TBD	TBD
Chipset Compatibility	X58 Chipset	X79 Chipset	X79 Chipset	X99 Chipset	X99 Chipset	New HEDT Chipset (TBA)
Socket Compatibility	LGA 1366	LGA 2011	LGA 2011	LGA 2011-3	LGA 2011-3	New HEDT Socket (TBA)
Memory Compatibility	DDR3-1066	DDR3-1600	DDR3-1866	DDR4-2133	DDR4-2400	DDR4-2400+
Max TDP	130W	130W	130W	140W	TDP	TBD
Launch	Q1 2010	Q4 2011	Q3 2013	Q3 2014	1H 2016	2017
Launch Price	\$999 US	\$999 US	\$999 US	\$999 US	~\$999 US	TBD

Main features:

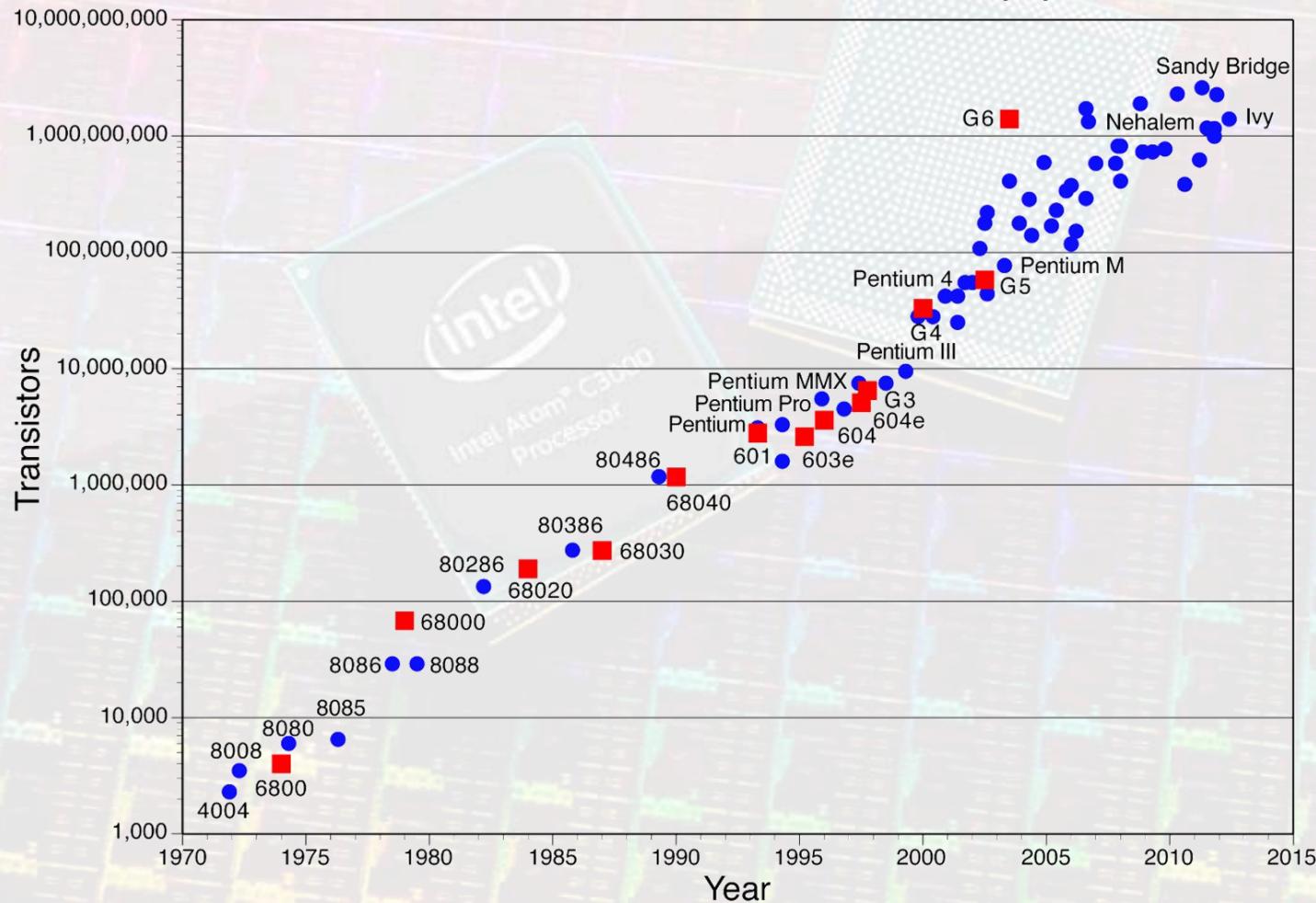
- clock frequency,
- CACHE size, organization
- packaging
- die size,
- number of transistors
- power consumption





A Moore's law for processors

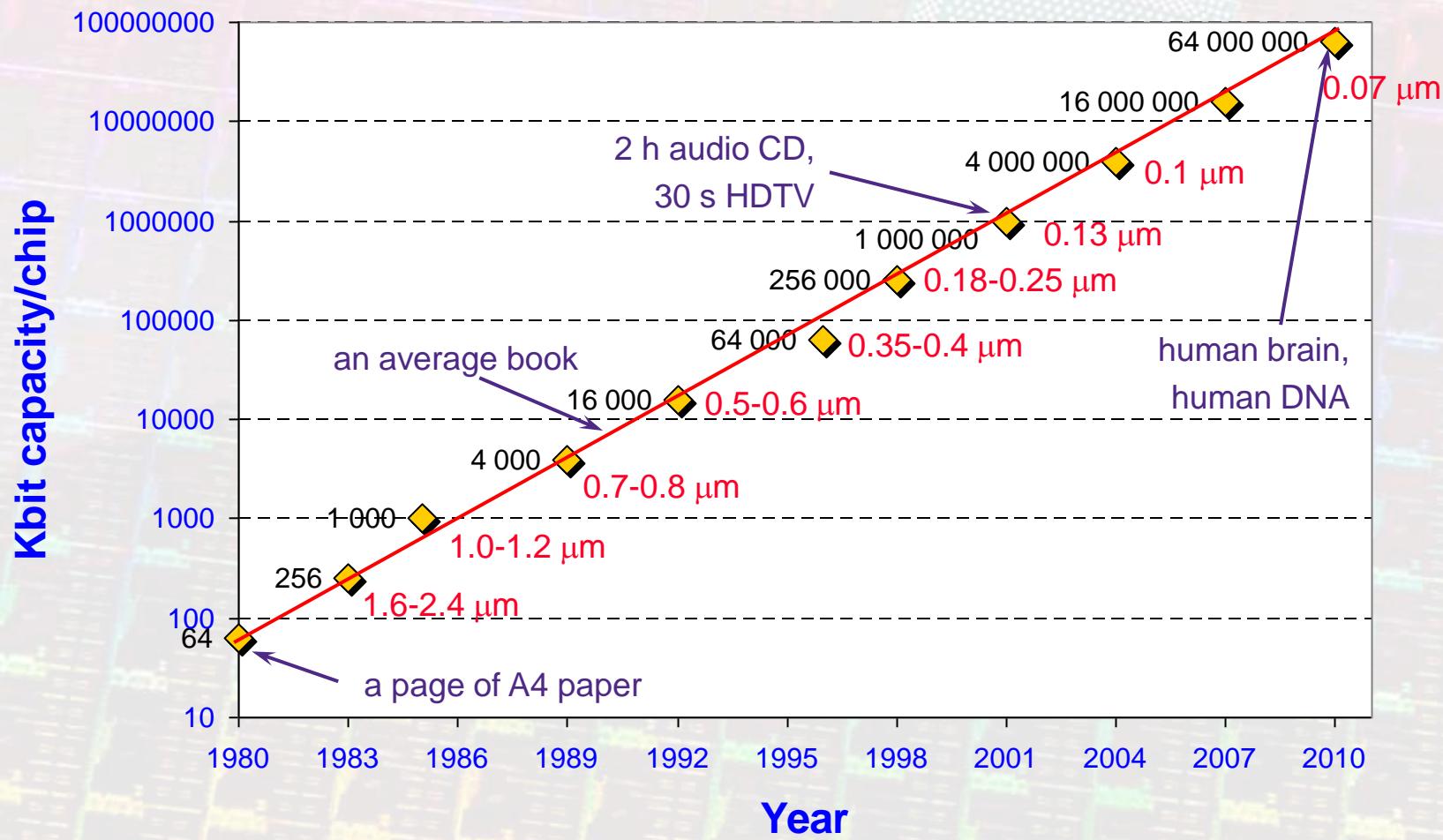
- Number of transistors doubles in almost every year





Development of DRAMs

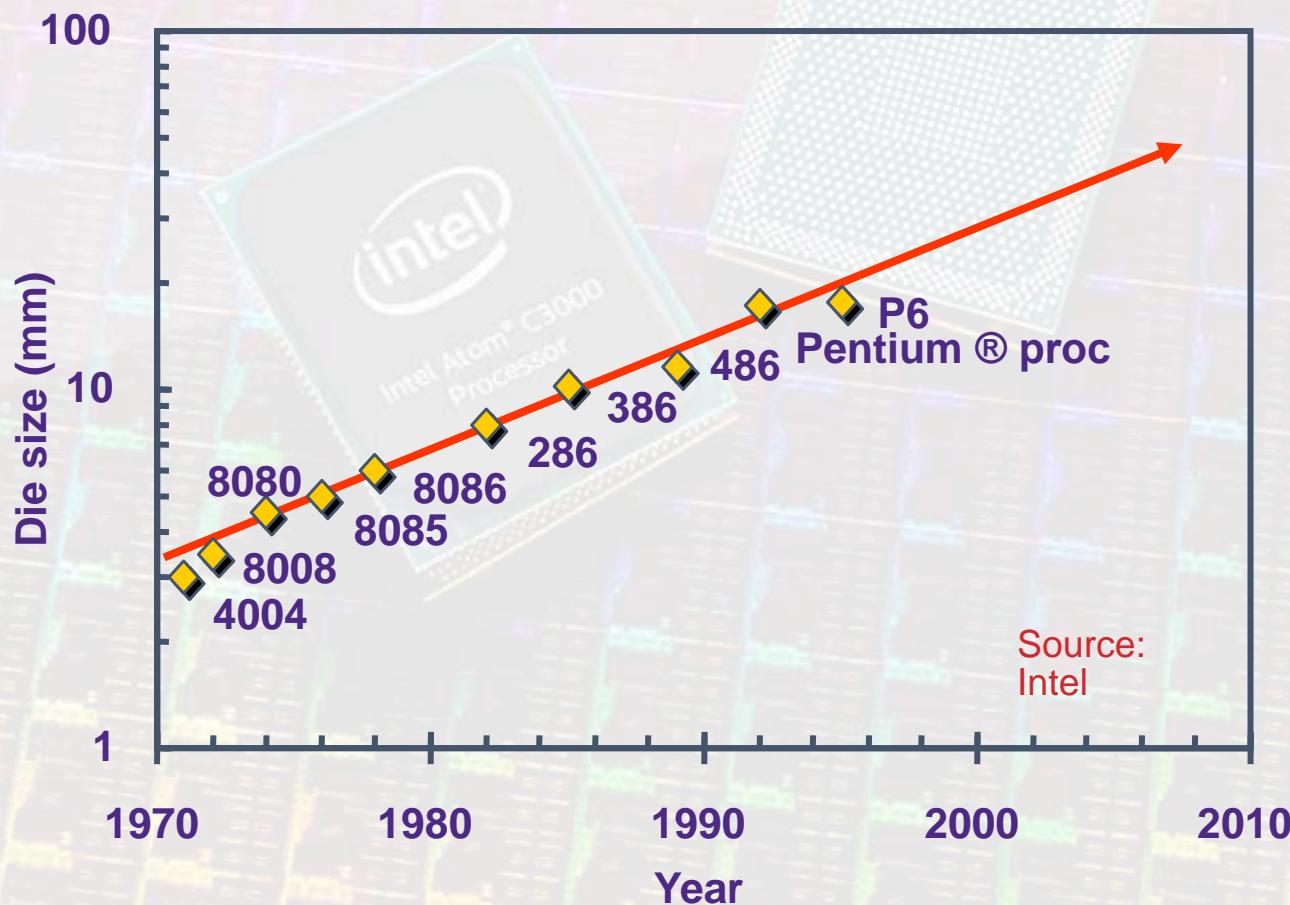
- 4-times growth in every 3 years





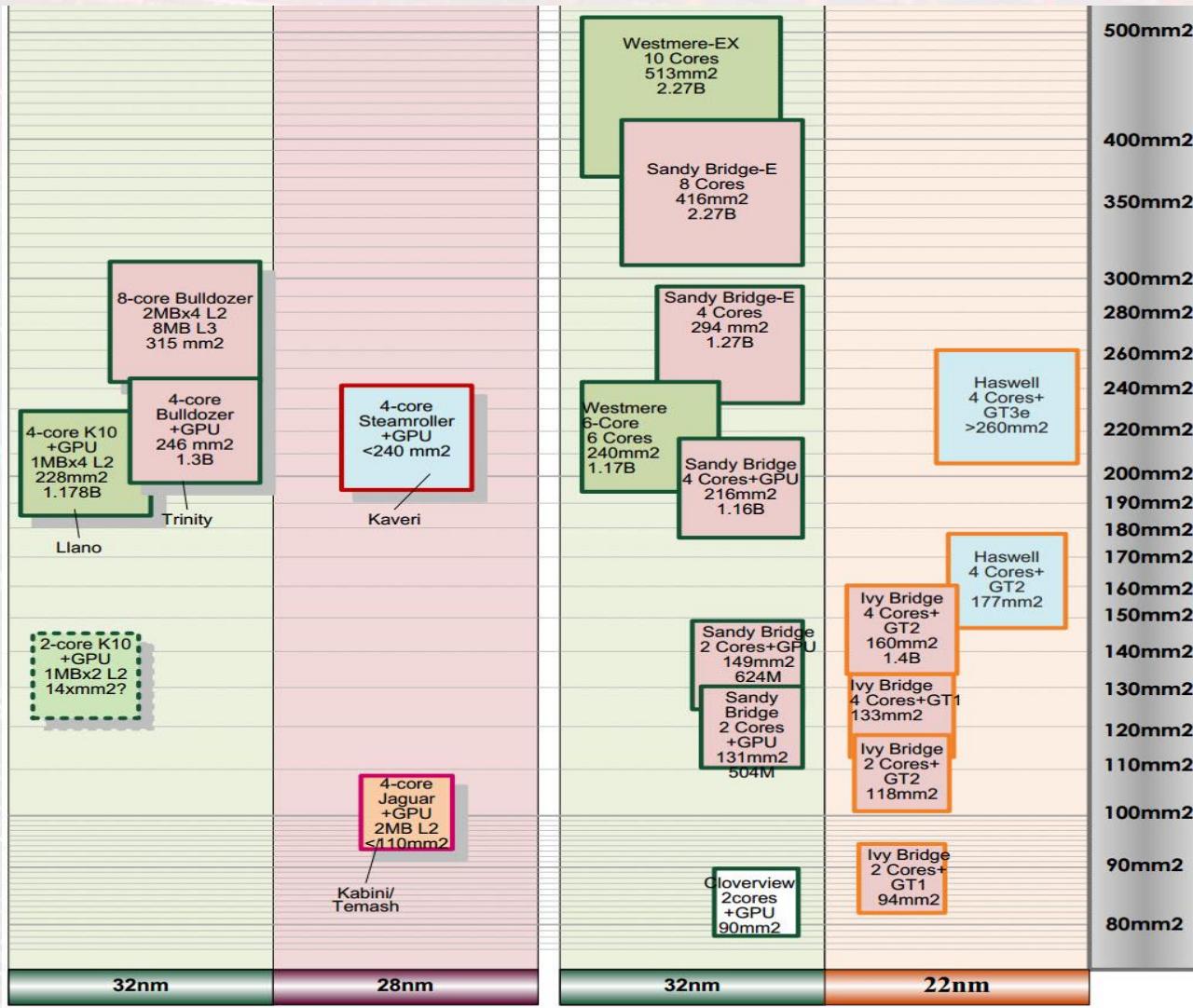
Increase of the *die size*

- 2-fold growth in 10 years – 7% annual growth, corresponds to Moore's law





Increase of the *die size*

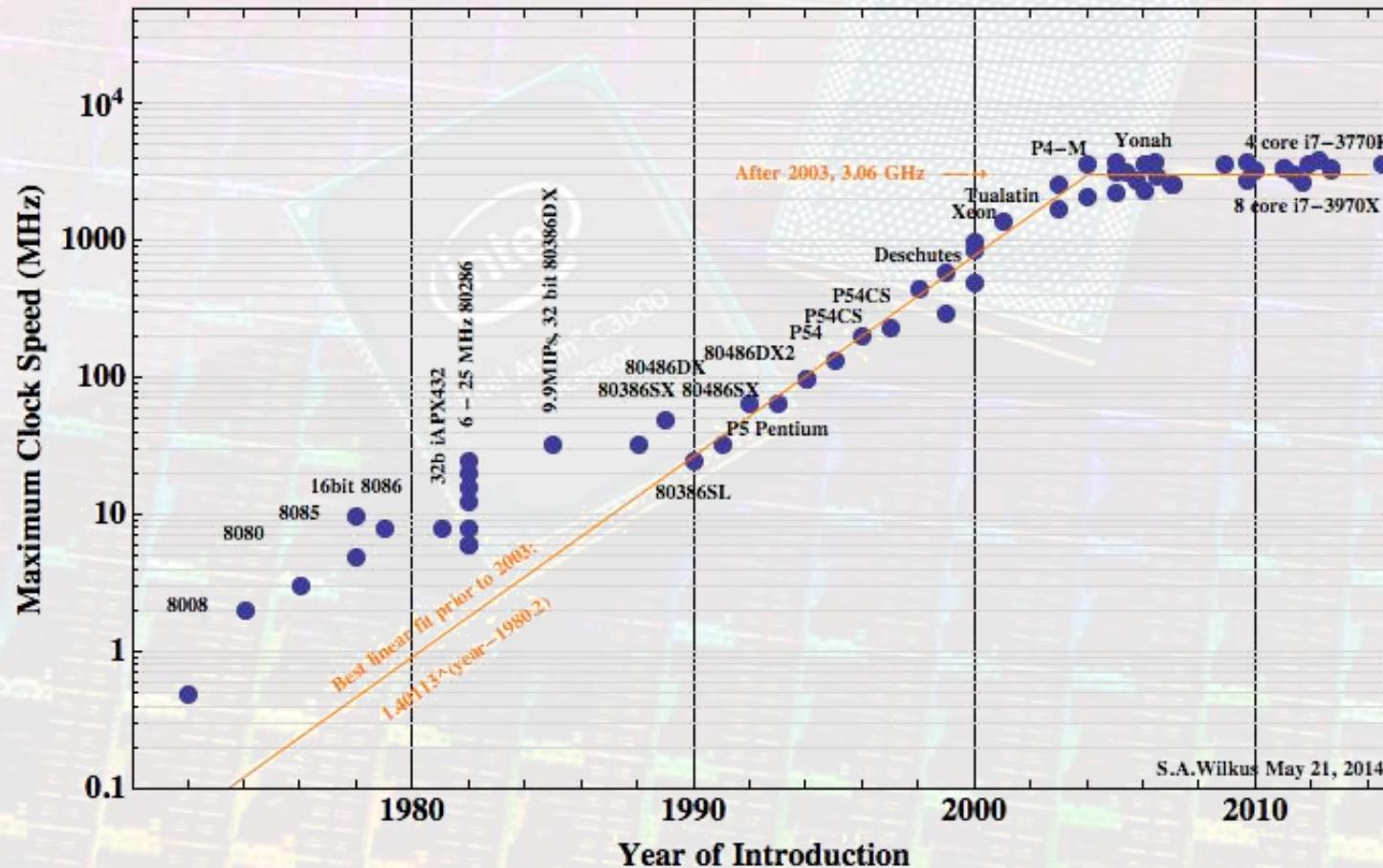




Clock frequency trend

2× every 2 years

μProcessor Clock Speed Trends



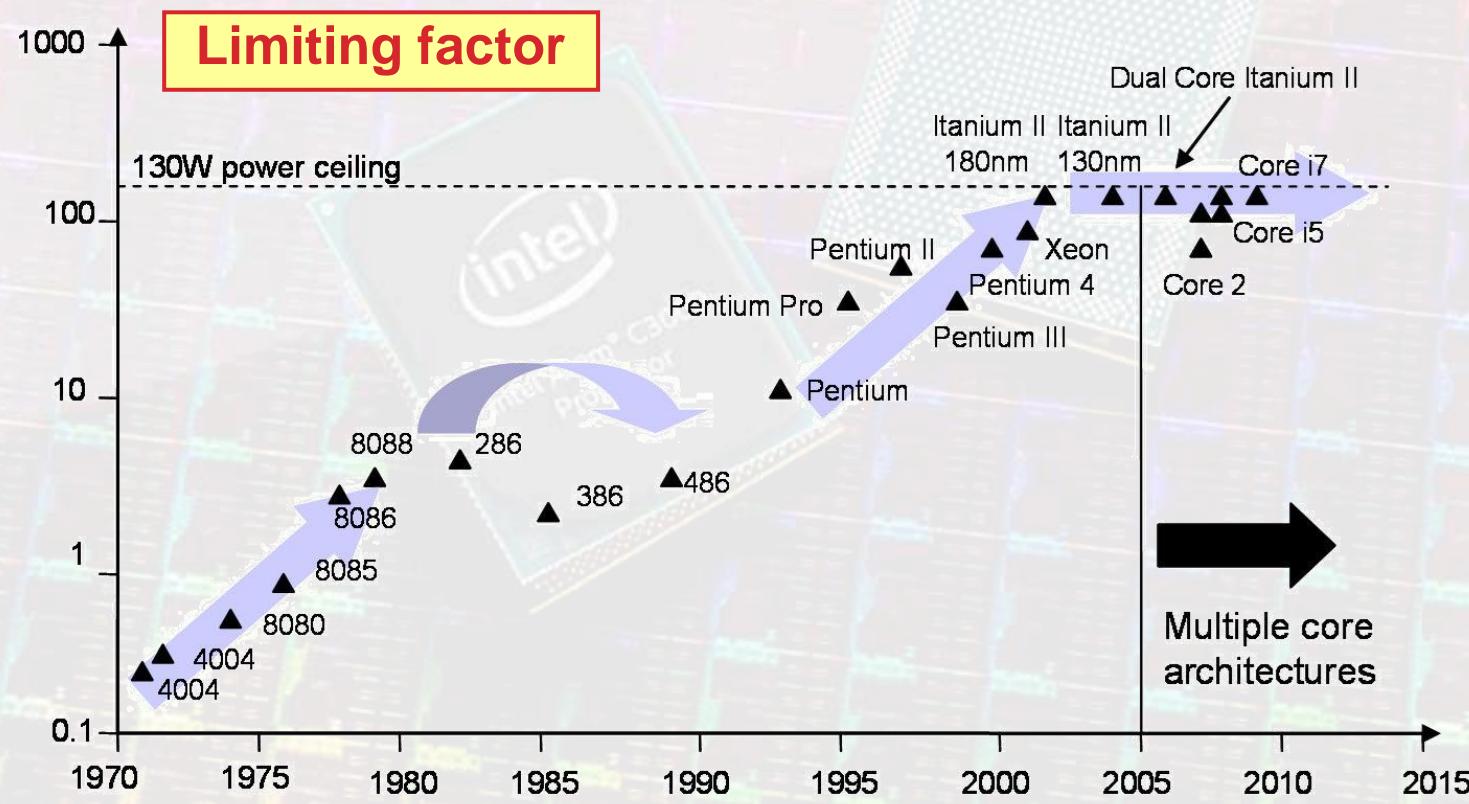
S.A.Wilkus May 21, 2014





Power consumption trend

- Continuous increase in case of processors



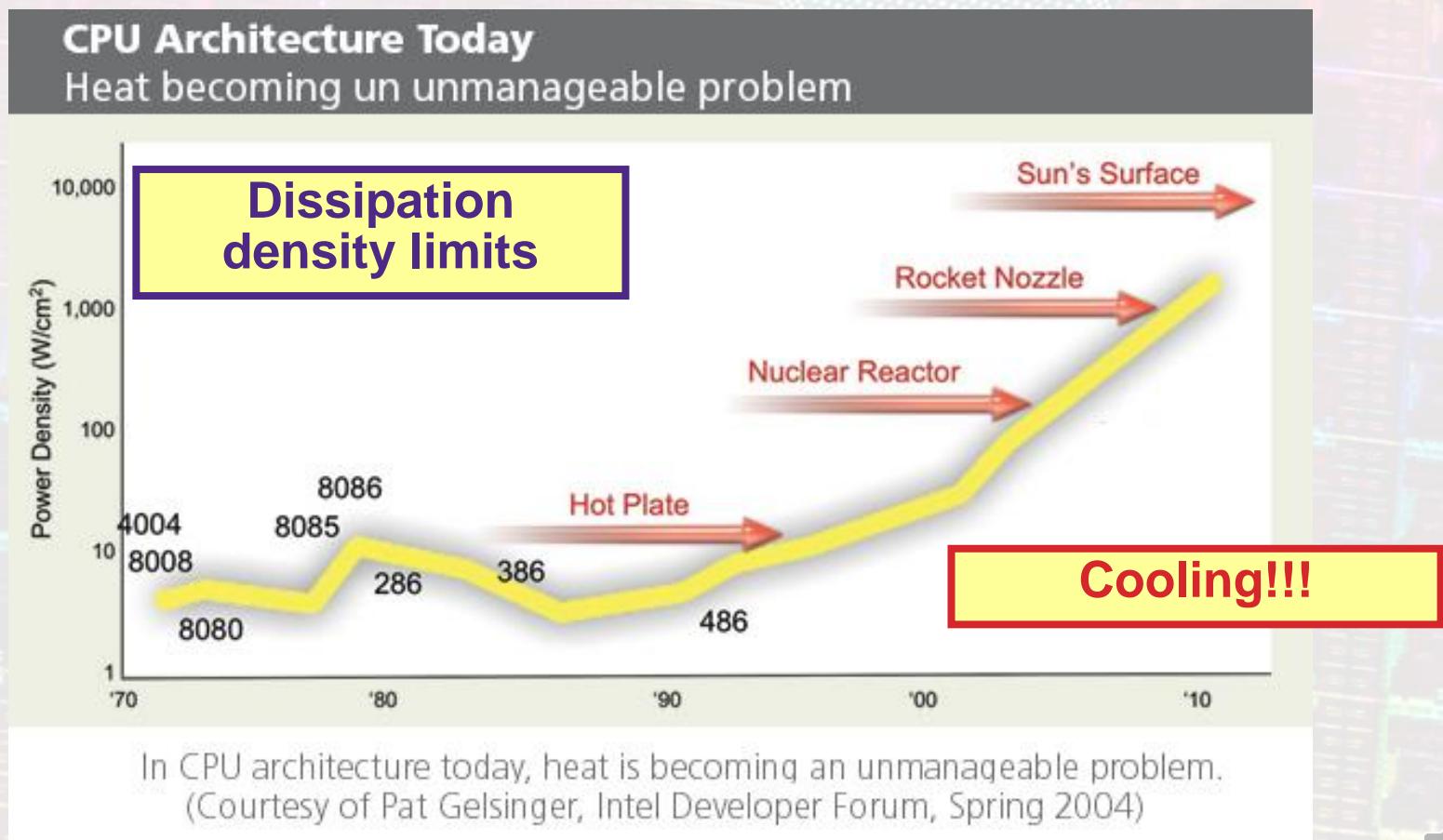
Adapted from ARC 2010 presentation by Dr. Ram Krishnamurthy, Intel Research





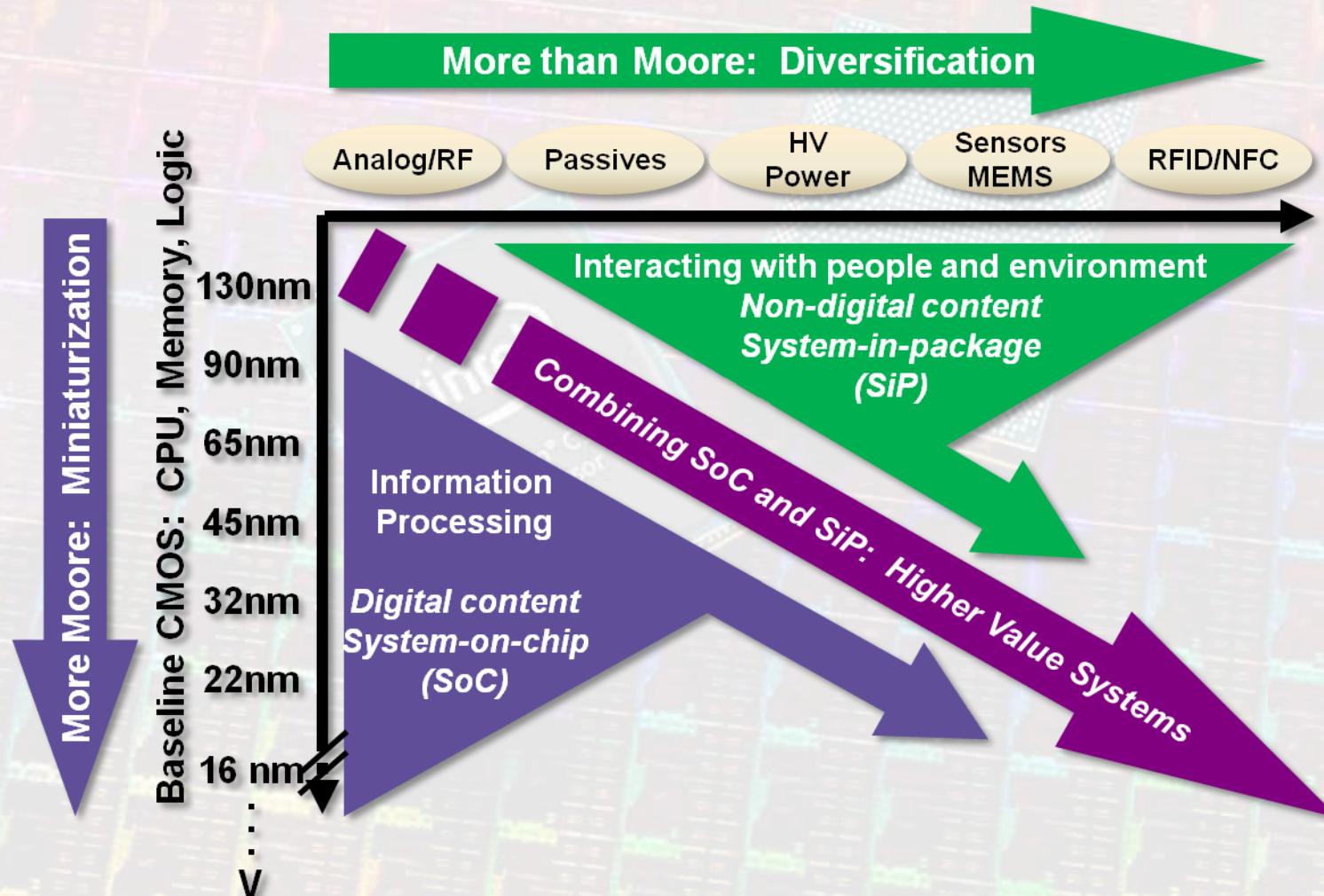
Increase in dissipation density

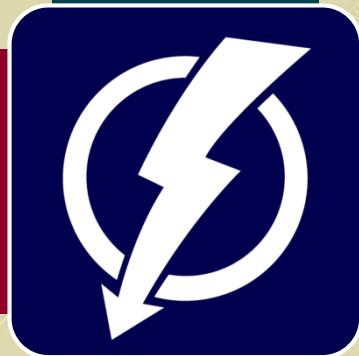
- Power consumption growth faster than the die size





More than Moore integration ITRS





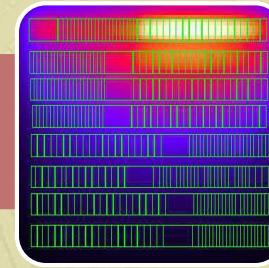
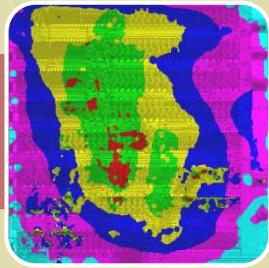
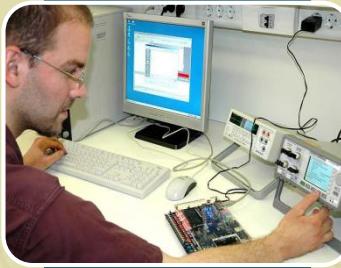
Budapest University of
Technology and
Economics



Microelectronics, BSc course

Manufacturing technology

Department of Electron Devices



The process: manufacturing technology

Overview of the steps and equipment



Production of polysilicon

20% of the Earth's crust is made up by silica glass (SiO_2) or silicat

From silica glass Si with 2...3% impurity can be produced in arc furnace:



Reacted with hydrochloric acid gas, trichlorosilane with boiling point of 32°C



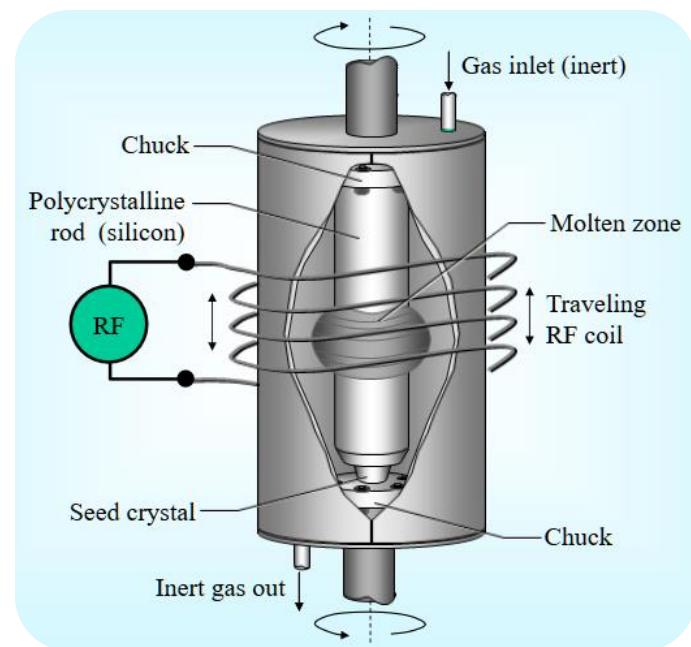
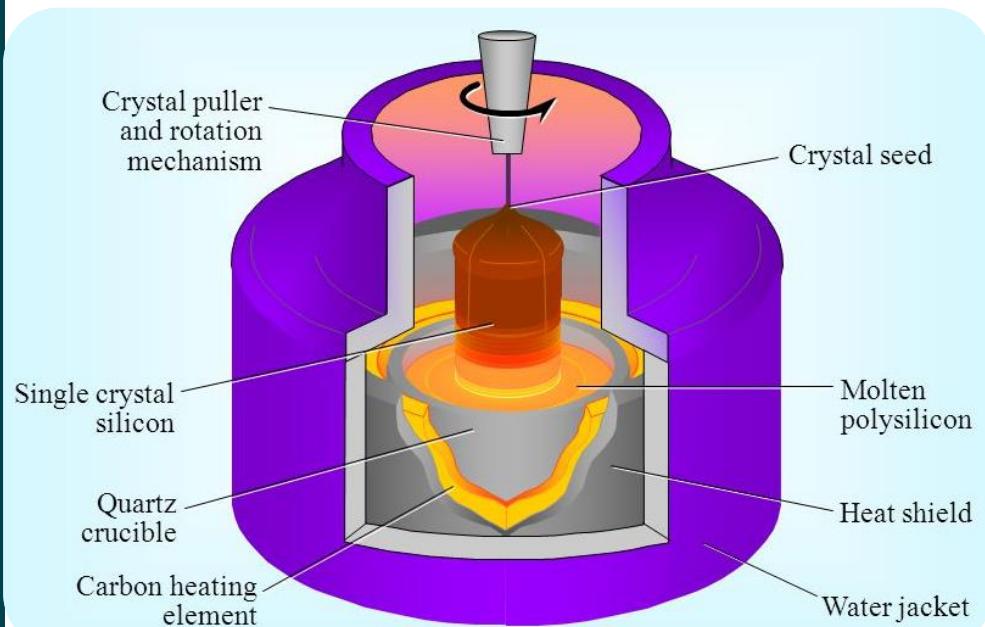
Impurity concentration can be decreased below 10^{13} db/cm^3

Poly Si (rod) manufacture:



Creating single crystal rods

Czochralsky (CZ) or FloatingZone (FZ) procedure



Properties of the silicon

14 atomic number in the periodic table

Melting point at 1415°C

Diamond crystal (face centered cubic)

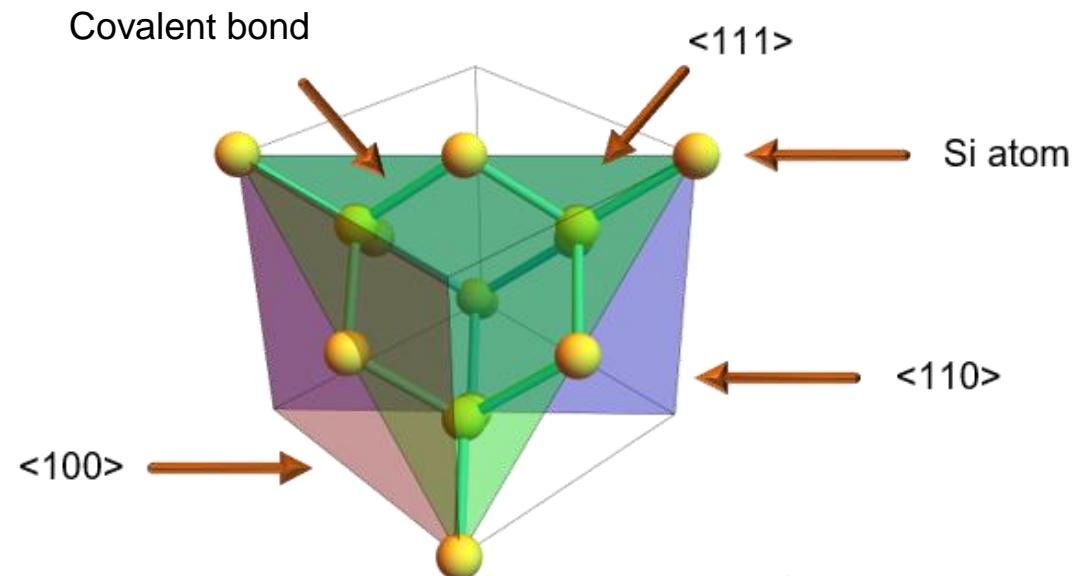
Good heat conductance property $\alpha = 156 \frac{W}{mK}$

Excellent mechanical properties

High degree of hardness

High tensile stress

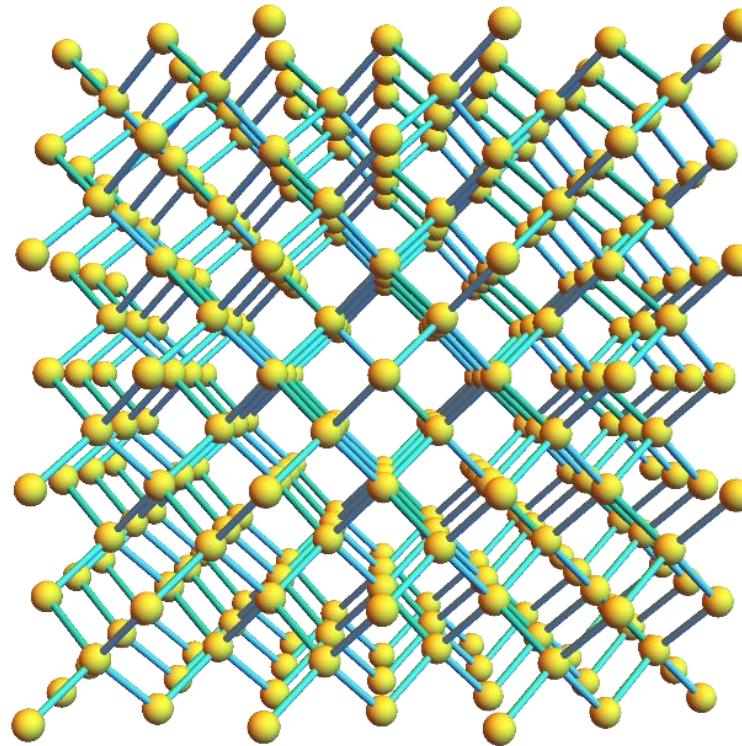
But rigid



Properties of the silicon

Different orientation (*viewing the structure from different views*) different usage

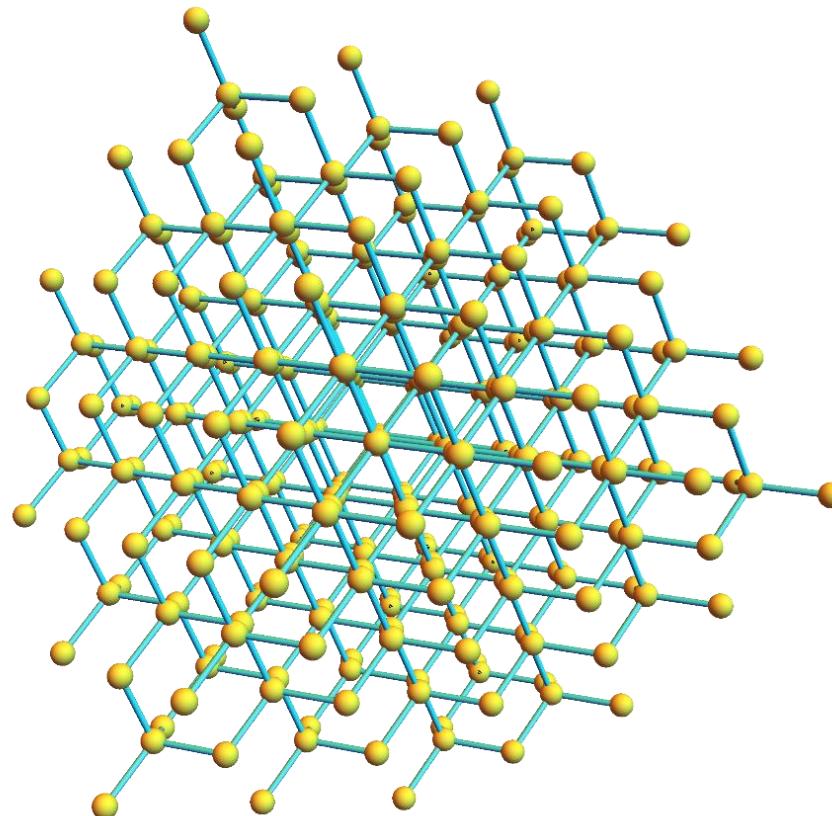
CMOS circuits: substrate with $<100>$ orientation



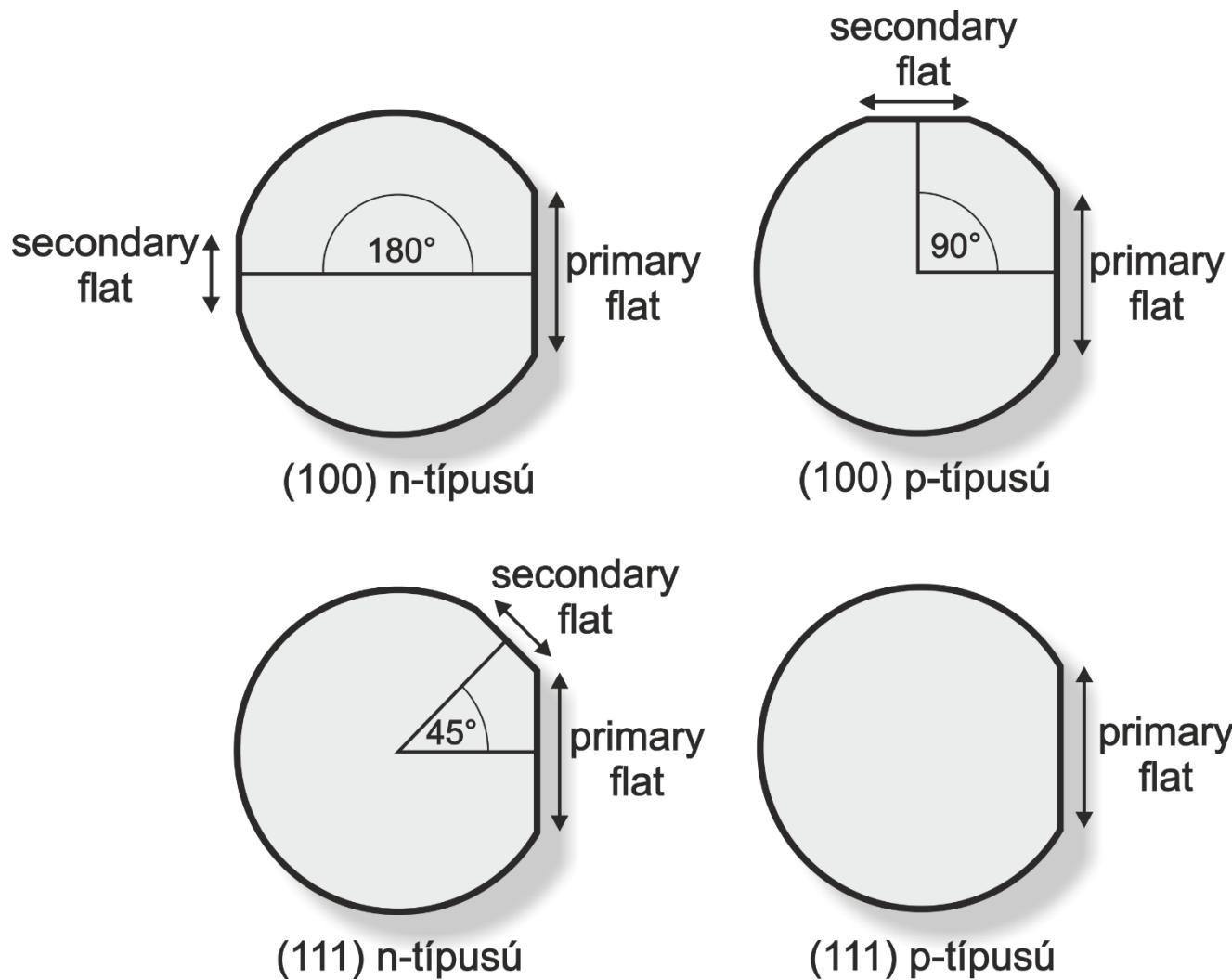
Properties of the silicon

Different orientation (*viewing the structure from different views*) different usage

Bipolar circuits: substrate with $<111>$ orientation

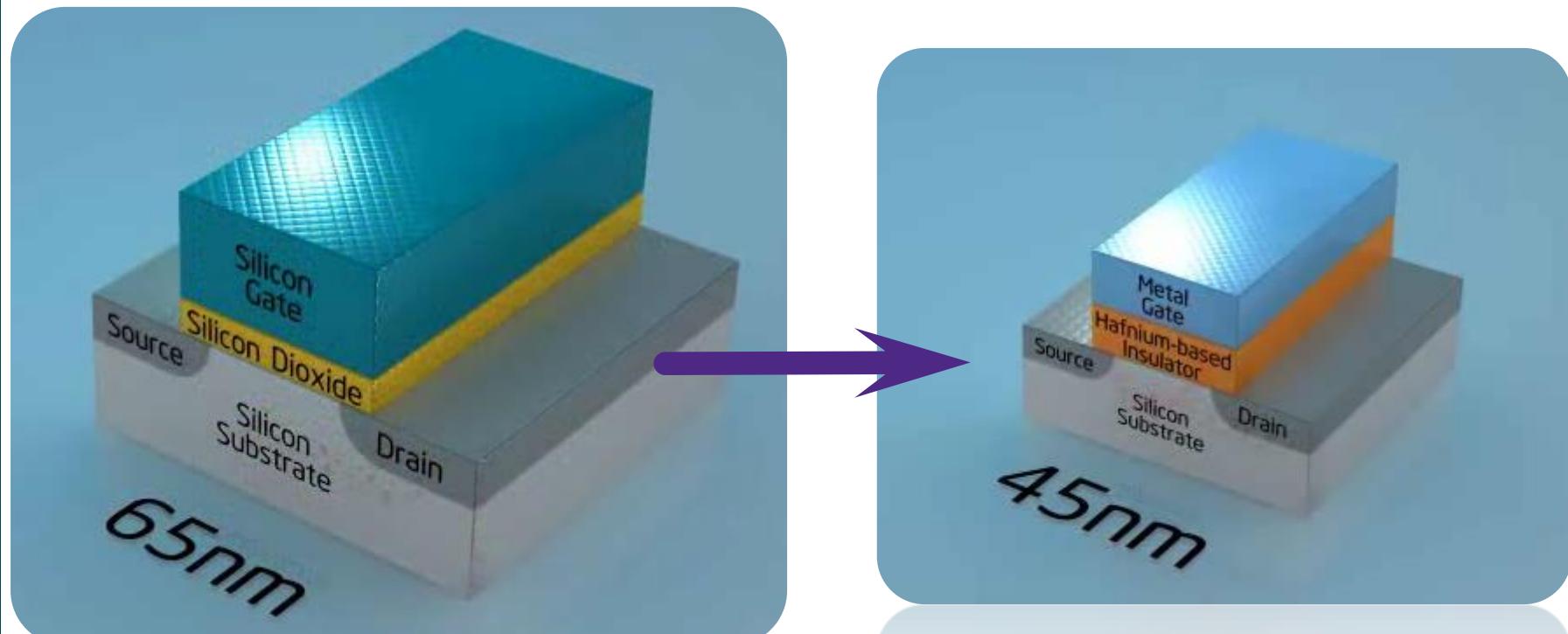


Substrate orientations



Devices to manufacture - MOSFETs

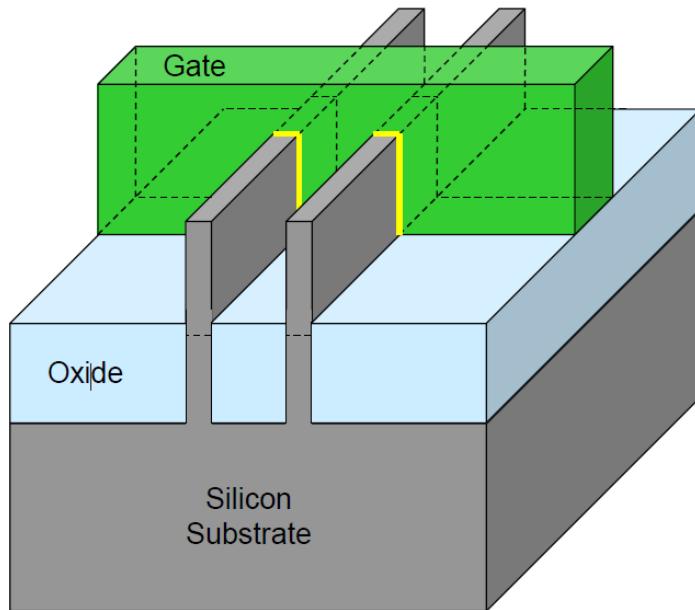
Minimal Feature Size (MFS) 2007/2008, Intel



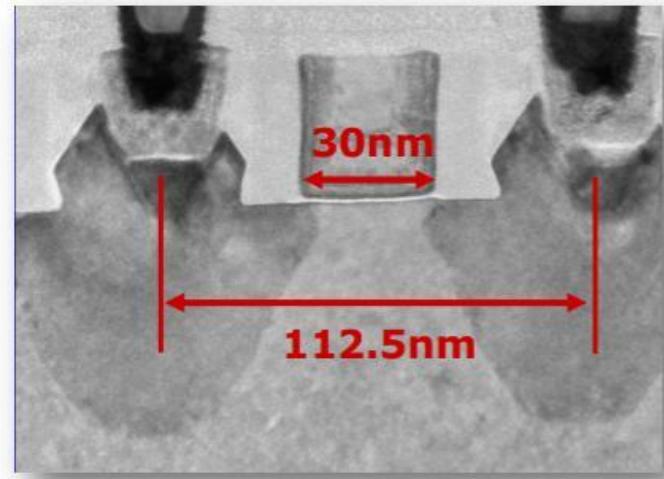
Devices to manufacture - FinFETs

Minimal feature size (MFS)

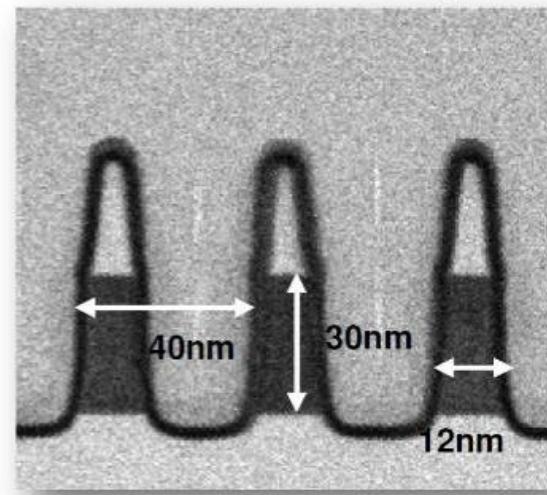
Intel 2012, 2014, 2018 (10nm)



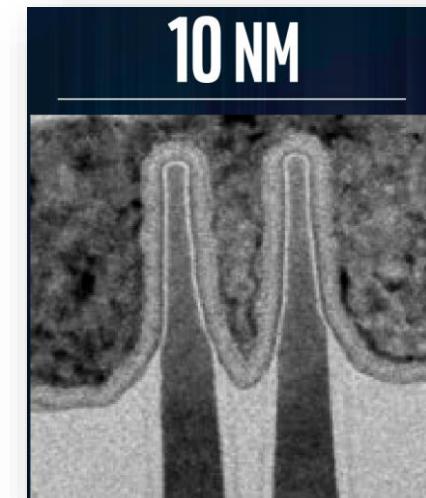
32nm Planar Example



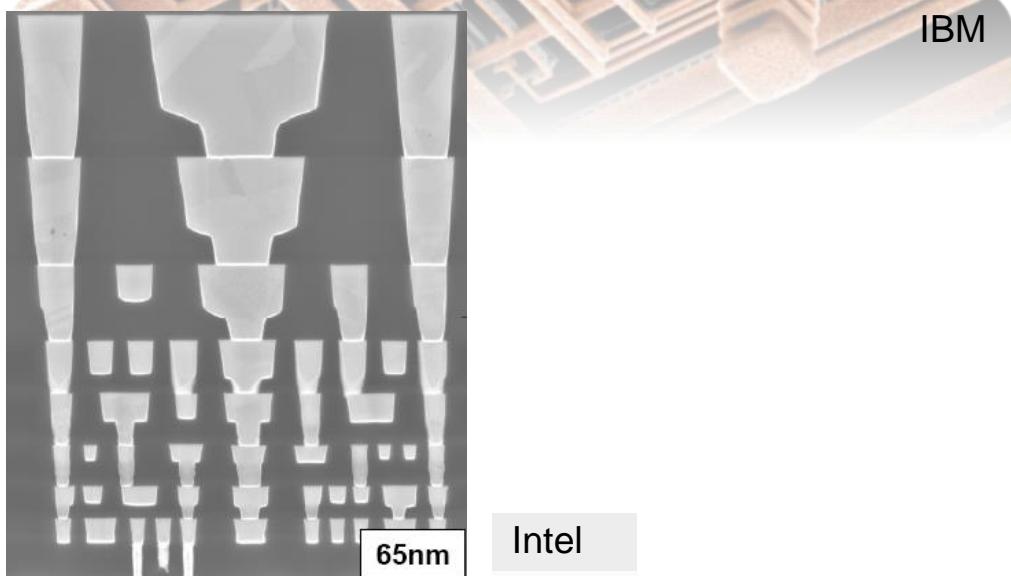
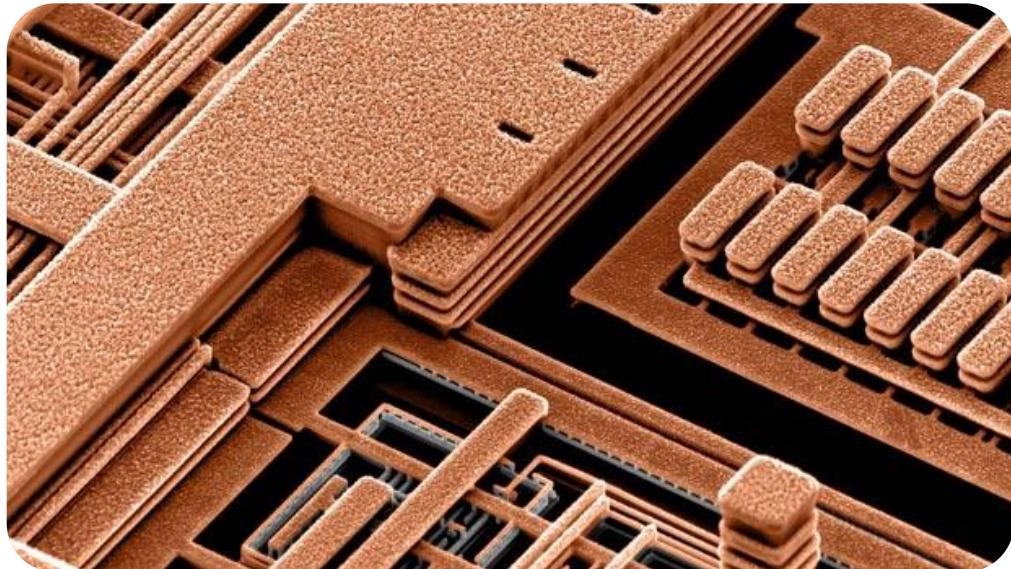
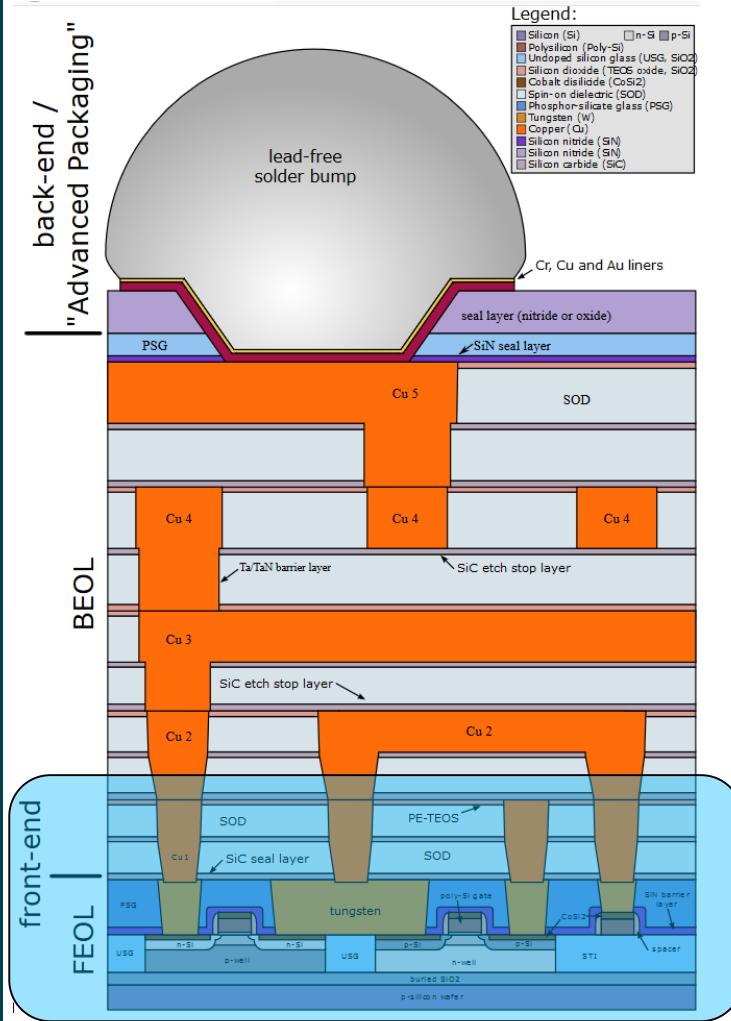
22nm FinFET Example



10 NM

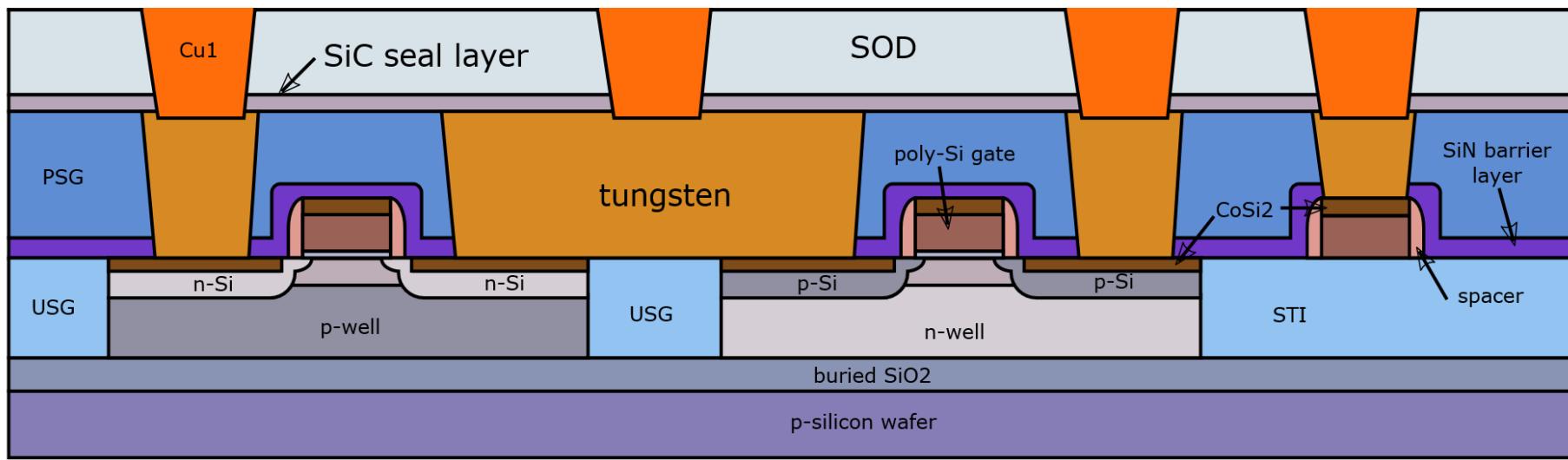


Devices to manufacture - Metalization



Intel

Devices to manufacture - Metalization



Basic processing principles

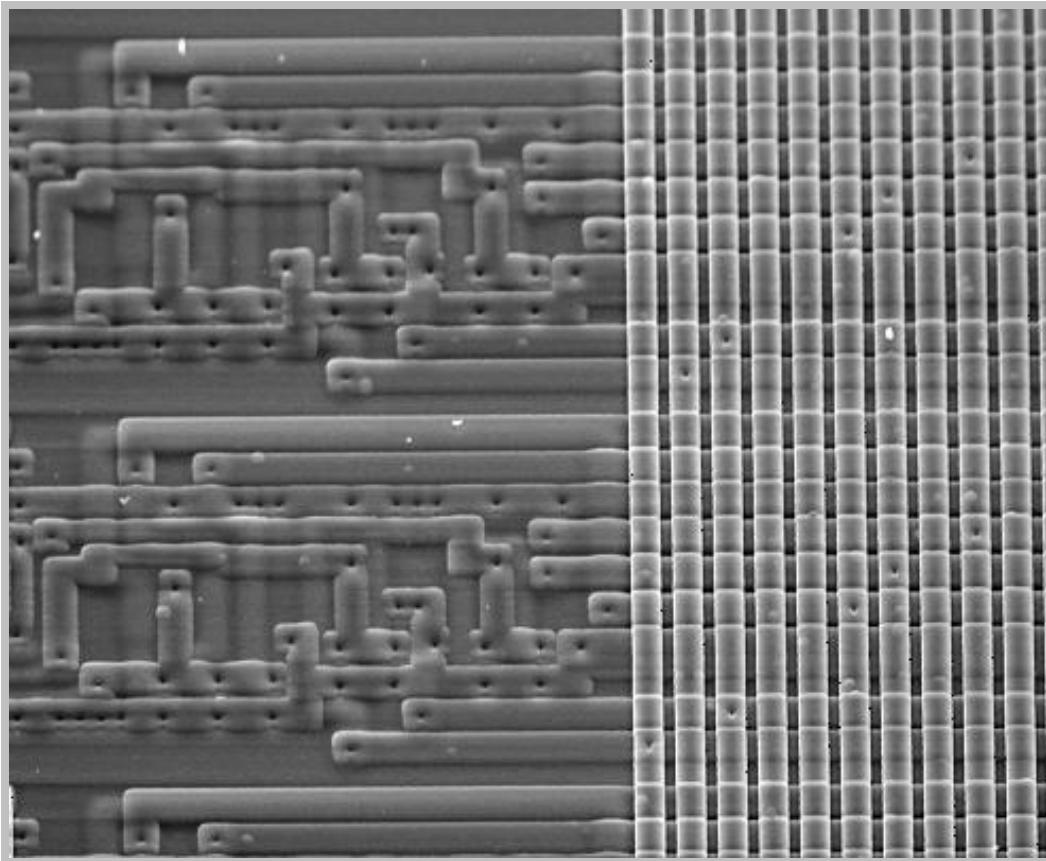
Layer growth or deposition: new material layer is formed over the entire surface of the wafer

In-depth deposition of external material: diffusion, ion implantation

Patterning: some patterns are formed in the deposited layer

- deposition of a photo-sensitive lack (**photoresist**)
- photographing the pattern onto the lack
- developing the photoresist: pattern formed in the resist layer
- transferring the pattern from the resist to the material layer underneath by some kind of **etching**
- removal of the resist

Monolithic IC-s



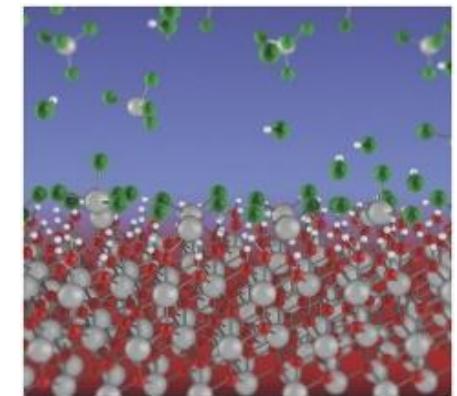
Mono lit = single stone

- In-depth structure
 - *Layer growth, deposition*
- Surface structure
 - *Patterning*

MFS – the major property of a process

$15 \mu\text{m} \rightarrow 7 \text{ nm}$

Layer growth or deposition



In-depth structure

Layer growth / deposition:

Growth of epitaxial layer (continue the Si-lattice but doped)

today e.g.: IBE – ion-beam epitaxy: atomic layers are grown

LPE: *liquid phase epitaxy*

VPE: *vapor phase epitaxy*

CVD: *chemical vapor deposition* – continuous carrier gas (H_2) flow

MBE – *molecular-beam epitaxy*: atomic layers are grown in 10^{-8} Pa vacuum
(examination of quantum effects, possible way to create quantum devices)

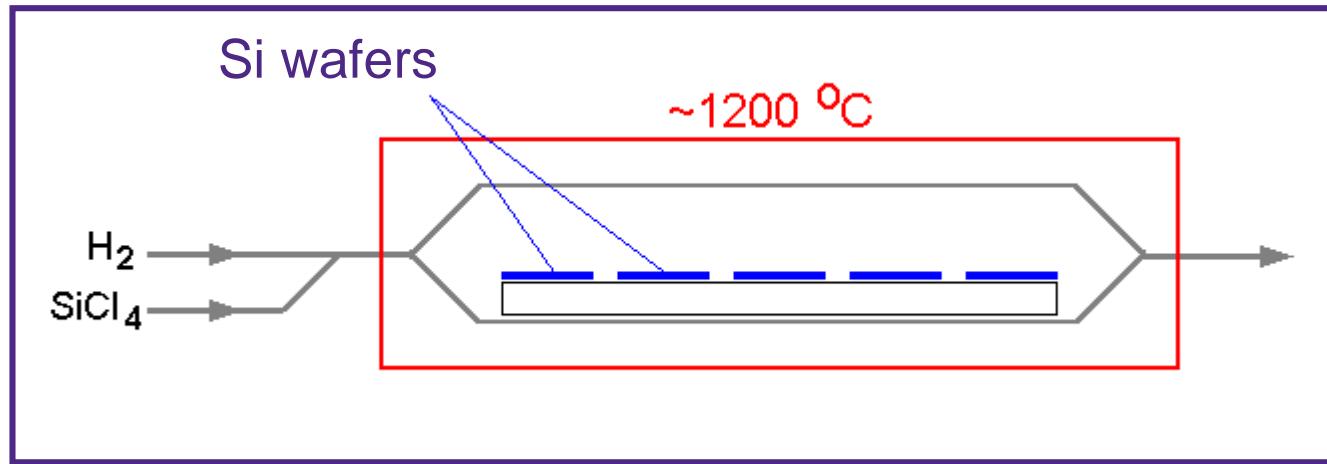
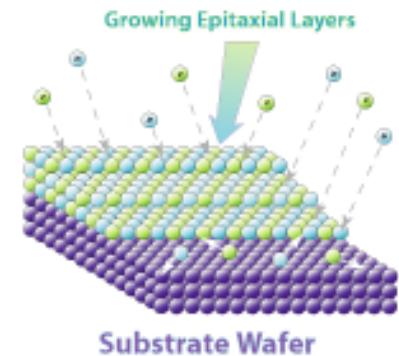
Oxidation (deposit/grow SiO_2)

Evaporation (e.g. deposit metal such as Al)

Sputtering

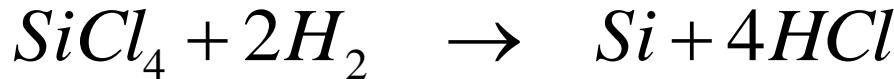
Growth of epitaxial layers

The classical epitaxial growth
either from gas or from liquid phase



The crystalline structure of the Si wafer is perfectly continued by the layer grown

$\sim 1200 \text{ }^{\circ}\text{C}$



Growth of epitaxial layers

Depending on the SiCl_4/H_2 ratio

- Growth of a single crystalline layer
- Growth of poly-crystalline silicon – called poly-Si
- Etching off Si
- Doping!



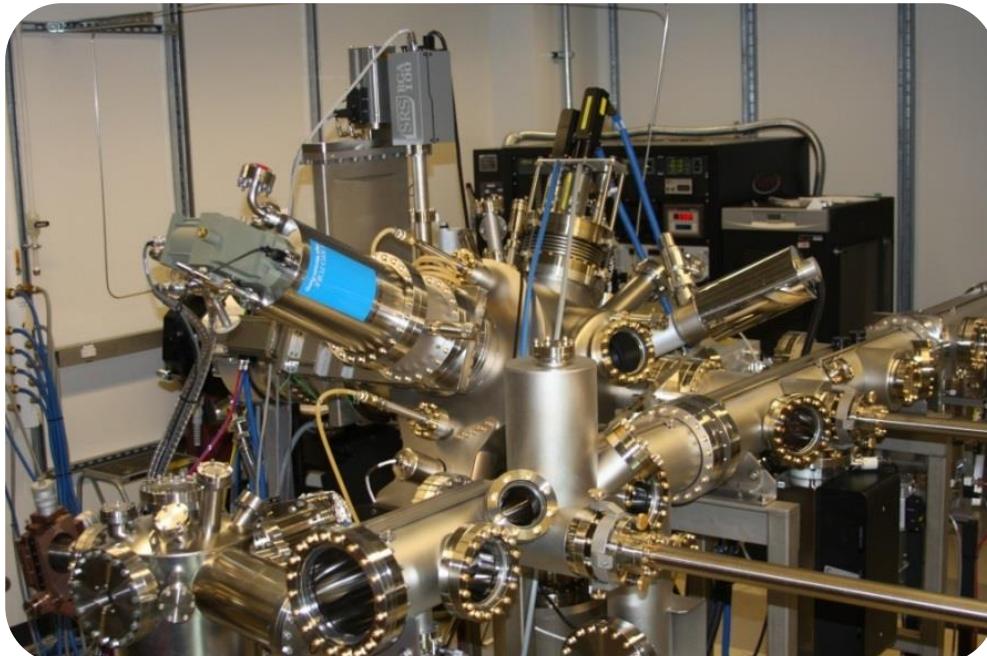
Growth of epitaxial layers

Molecular-beam epitaxy:

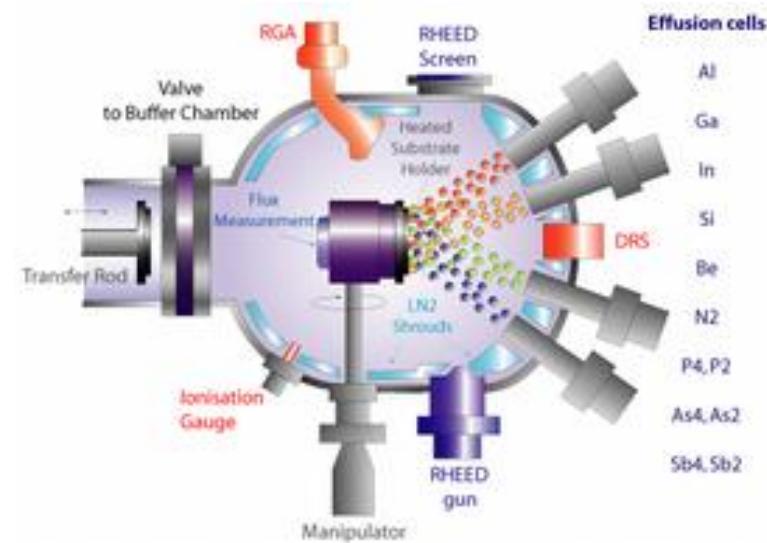
Multi-layer, varying composition, compound semiconductors

Quantum devices

Cc. 100 nm/h grow speed



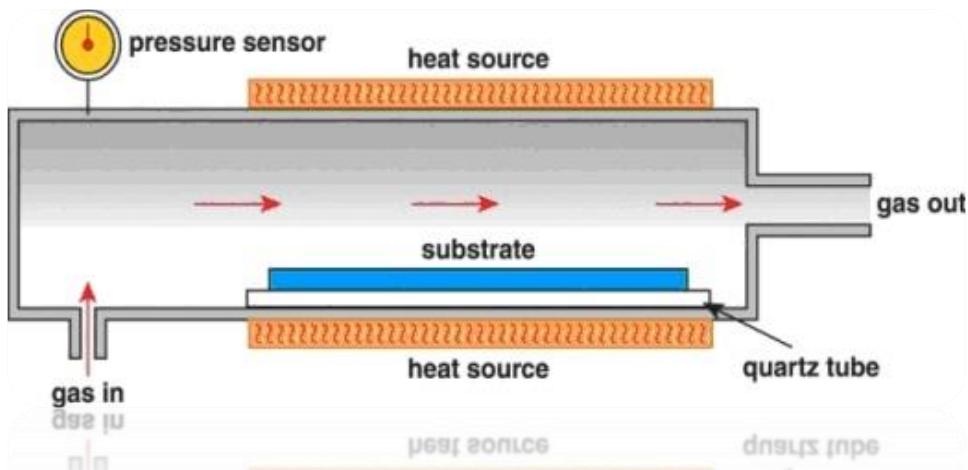
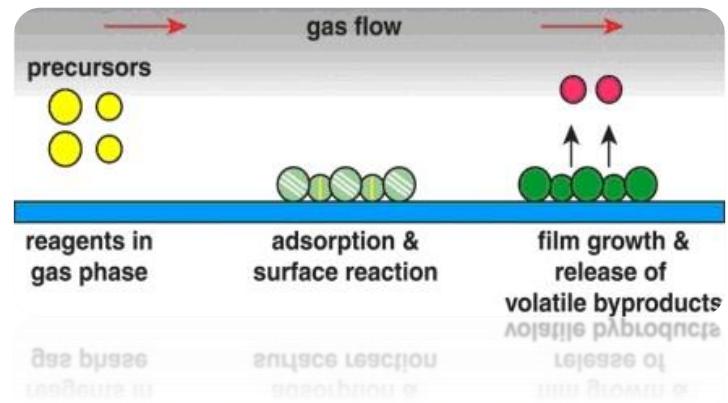
MBE: molecular beam epitaxy



Growth of oxide layers

- ▶ Thermal oxidation (900-1200 °C)
- ▶ Chemical Vapor Deposition (CVD)

$$d_{SiO_2} \sim \sqrt{t}$$

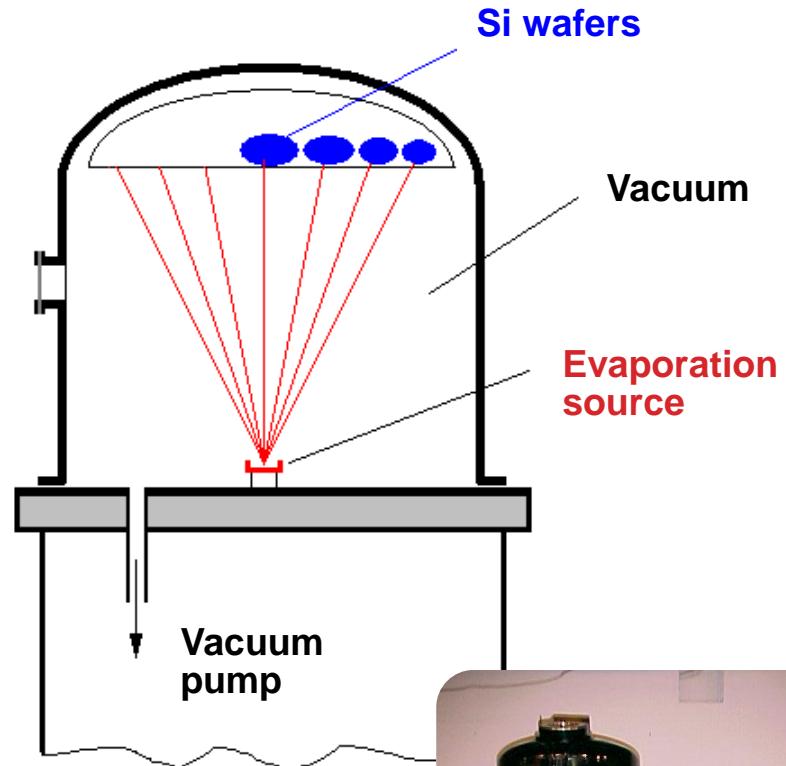
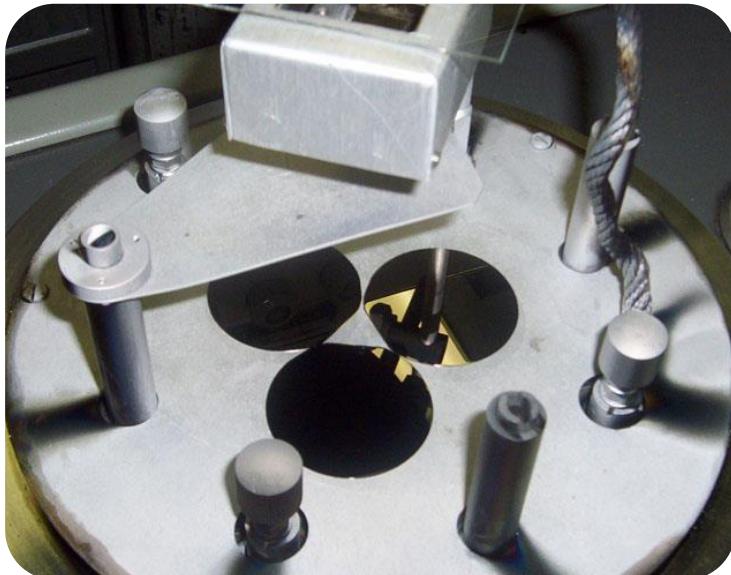


Vapor deposition

Free mean path > size of the chamber

Metallization

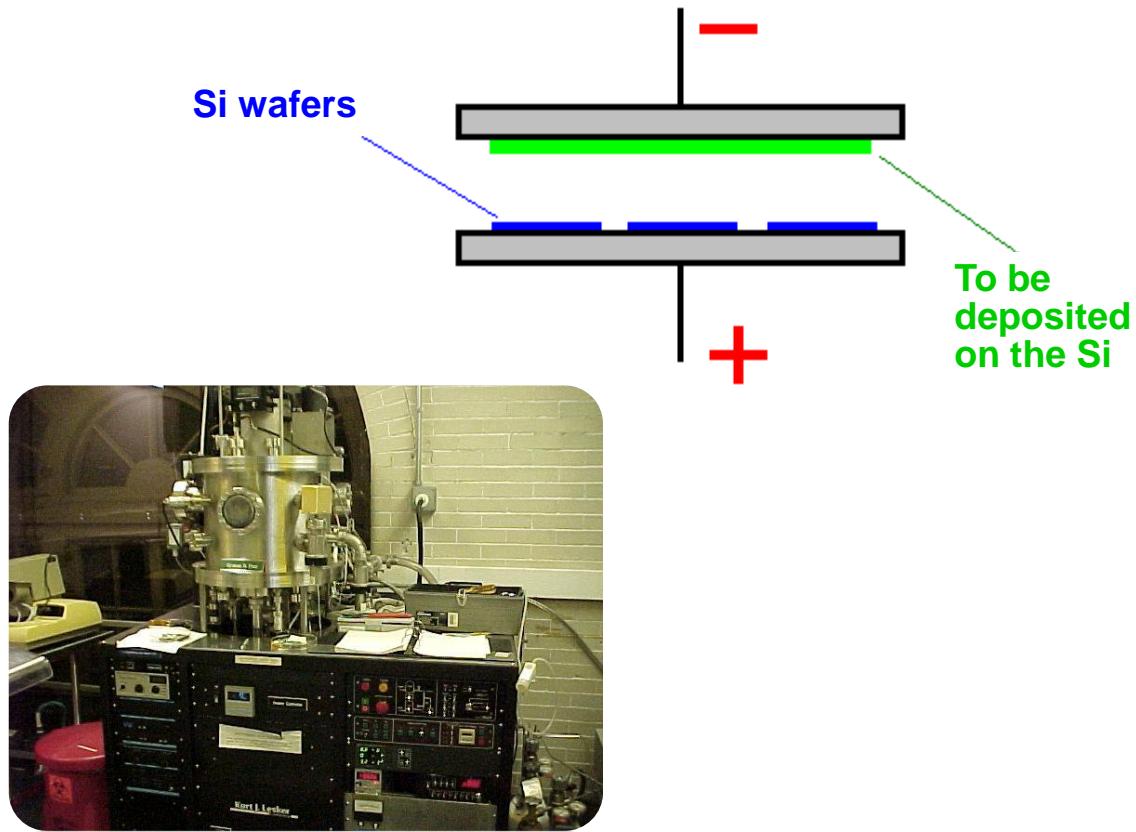
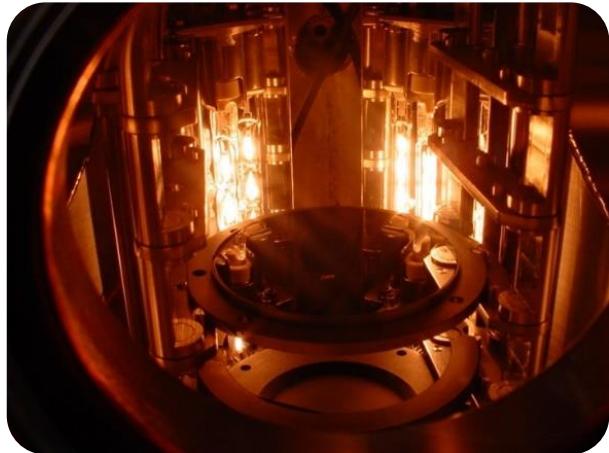
~0.1-0.5 µm



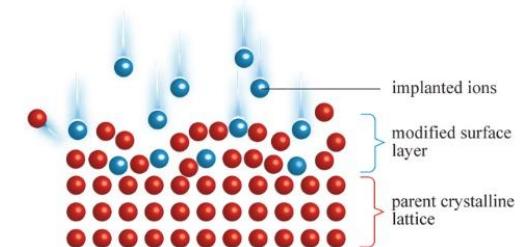
Sputtering

Gas discharge is used to carry the material to be deposited from a cathode (e.g. Ar atmosphere)

Using high frequency **dielectrics** can also be sputtered

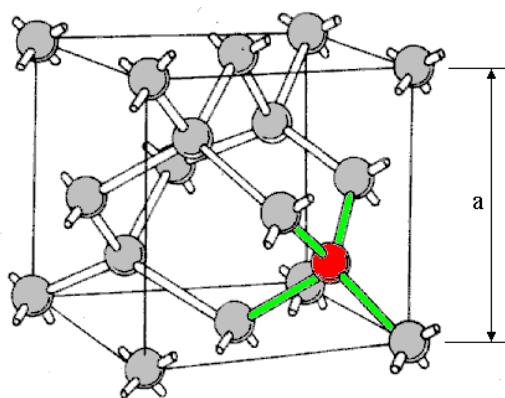


In-depth deposition of external material

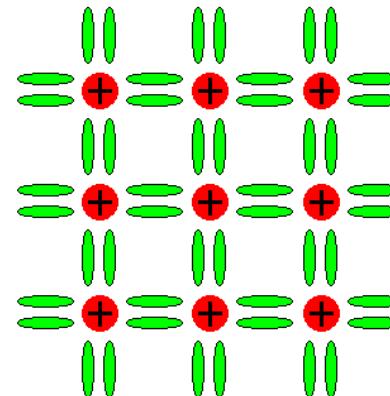


In-depth structure

Deposition of dopants (foreign atoms) in the silicon crystal to modify its properties



Simplified view in 2D

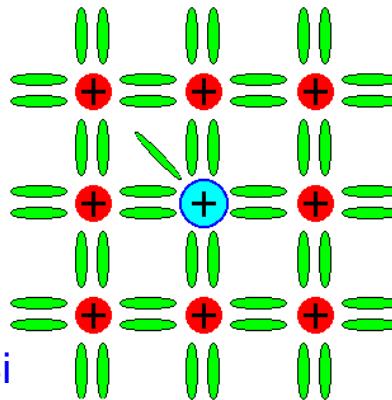


Diamond lattice in 3D

Dopant from
column V
(Phosphorus):

extra electron
DONOR

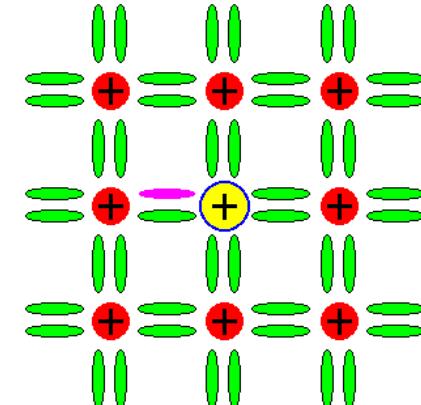
n-type Si



Dopant from
column III
(Boron):

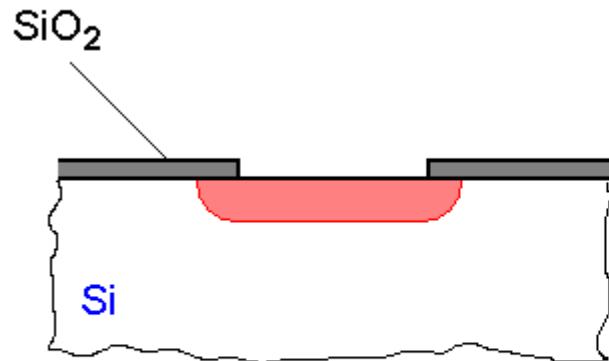
1 less electron
ACCEPTOR

p-type Si

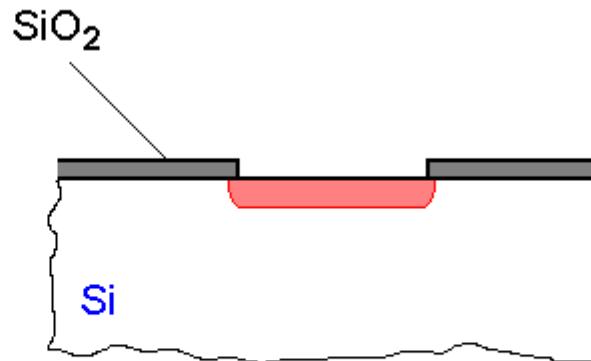


How to select where to dope?

SiO_2 is an excellent mask against the flux of dopants

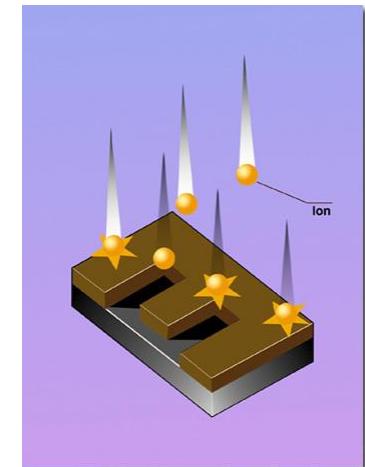


Diffusion
deep profile



Ion implantation
shallow profile

Masked by a SiO_2 pattern



In-depth structure

Deposition of dopants by diffusion

Dopants diffuse in the high temperature Si-lattice

The energy of the Si atoms helps the dopants move

Movement mechanisms:

interstitial movement: movement by changing place with a Si atom

movement along crystalline defects

In-depth distribution of dopants is determined by Fick's laws:

$$J = -D \frac{\partial c}{\partial x} \quad D = D(T) \quad !$$

$$\frac{\partial c}{\partial t} = - \frac{\partial J}{\partial x}$$

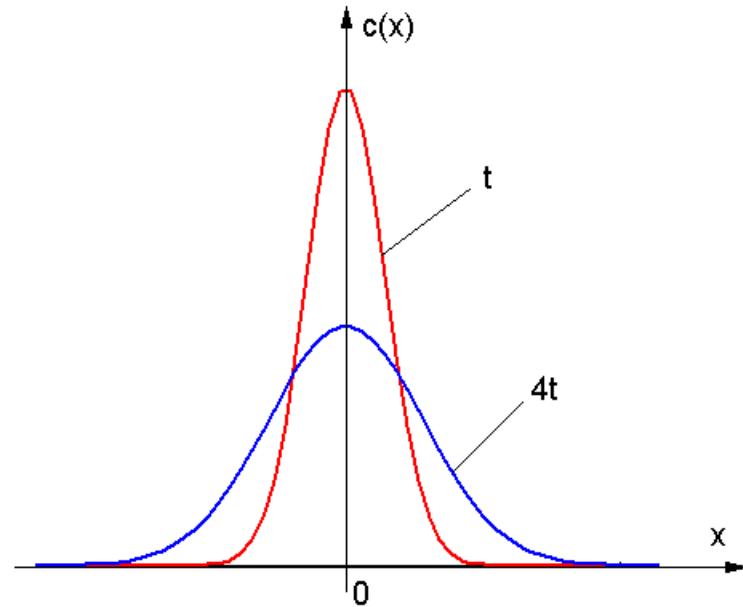
$$\boxed{\frac{\partial c}{\partial t} = D \frac{\partial^2 c}{\partial x^2}}$$

Diffusion

The solution is:

$$c(x,0) = M_0 \cdot \delta(x)$$

$$c(x,t) = \frac{M_0}{\sqrt{4\pi Dt}} \exp\left(-x^2 / 4Dt\right)$$

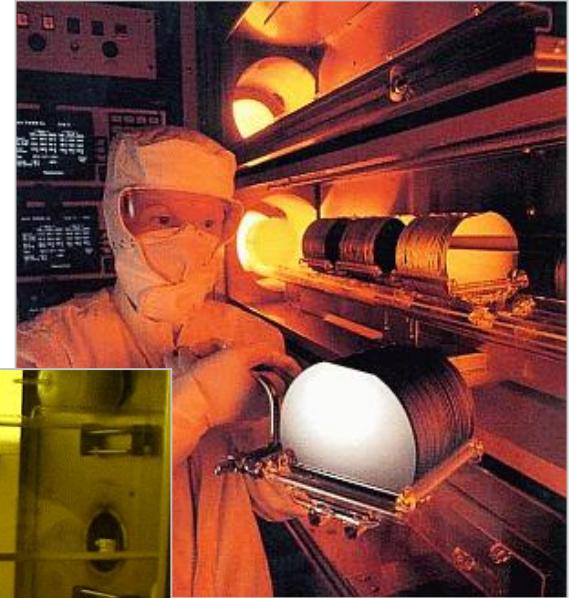


Two steps

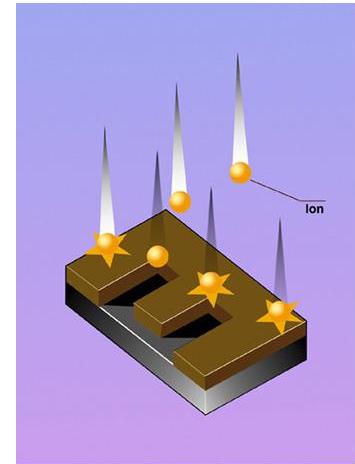
- initial deposition / pre-diffusion (e.g. 1100°C, 3 hours)
- drive-in (e.g. 1240°C, 1 hours)

Diffusion

The diffusion furnace



Masked by a SiO_2 pattern



Industry scale diffusion furnace

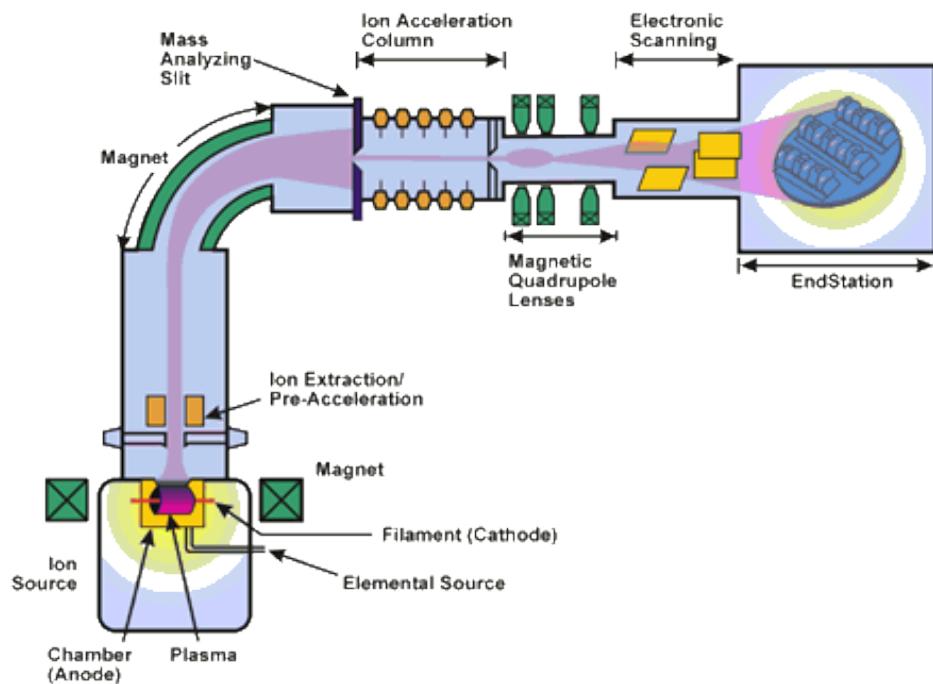


Ion implantation

Procedure:

1. Doping atoms are inserted into the ion source then they are ionized
2. Ions are accelerated by the electrical field
3. The surface of the wafer is bombarded to get the ions into the wafer

Ions fired upon the targeted Si wafer chosen from an ion beam with a mass spectrometer



Ion implantation

From an ion beam one selects the ions that target the Si and penetrate the lattice

Initial distribution of deposited dopants depends on the energy and the dose of the ion beam

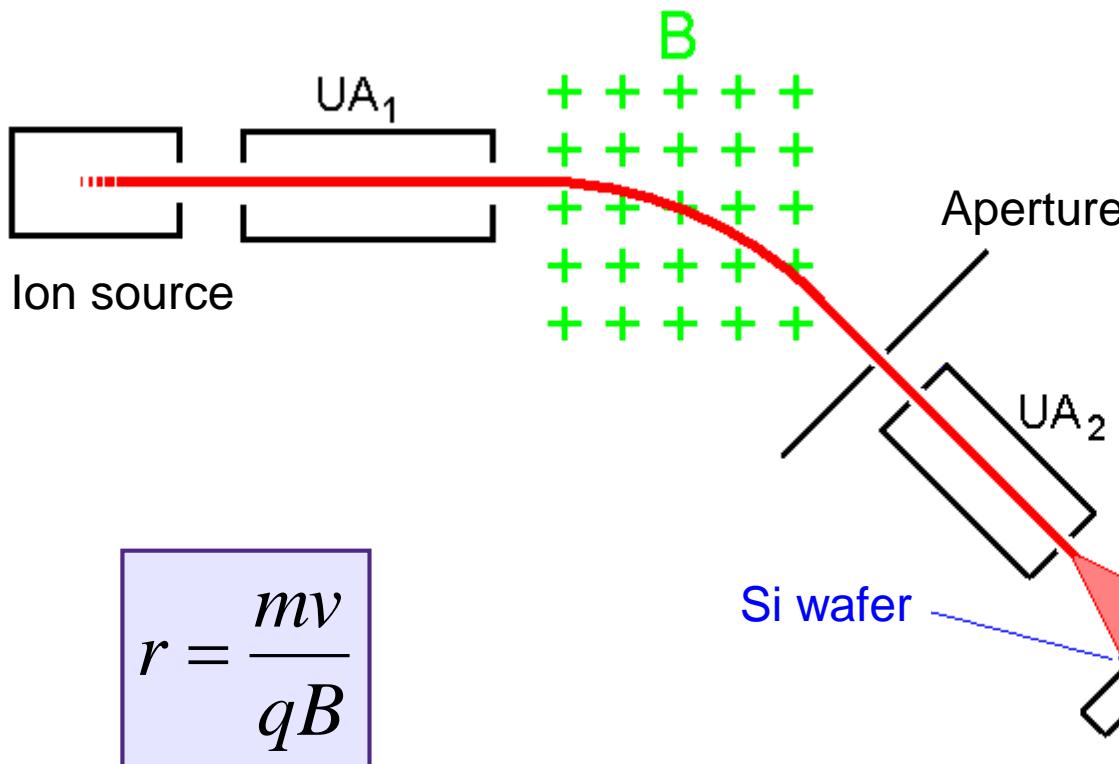
Thermal treatment follows the implantation

restore the Si-lattice

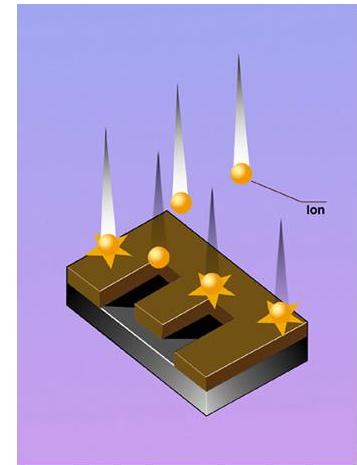
drive-in the dopants (form final doping profile)

~100 kV voltage is used

Ion implantation



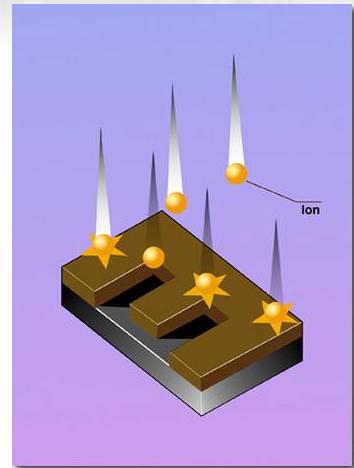
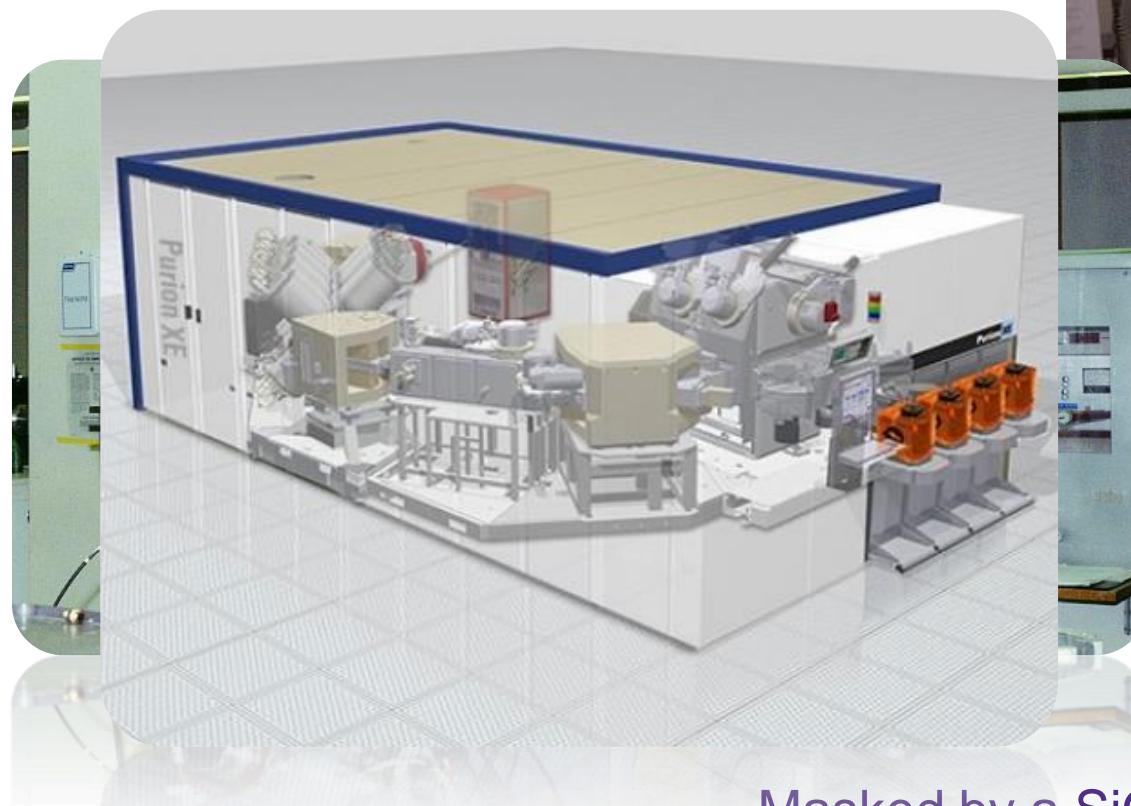
Masked by a SiO₂ pattern



Ion implantation

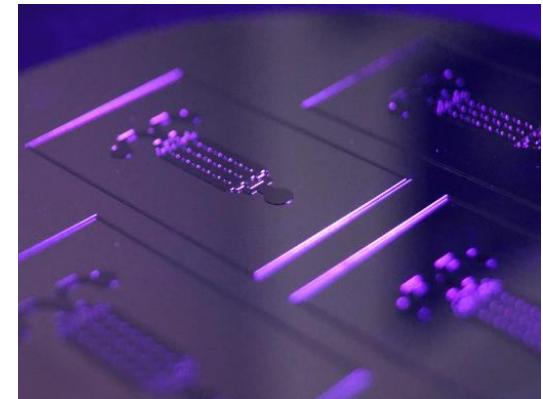
It is a **low temperature** process.

Advantage: existing profiles are less effected



Masked by a SiO_2 pattern

Patterning



Window opening

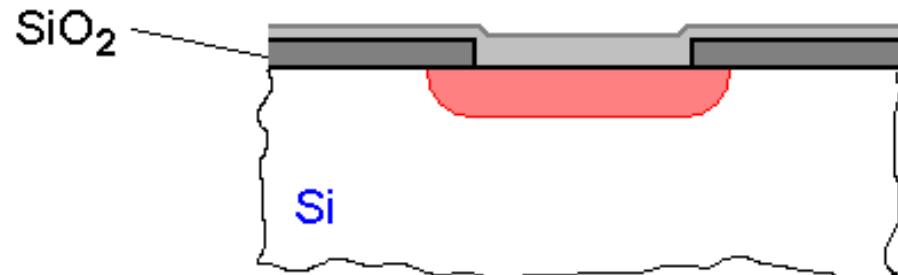
With photolithography – always the first step of any patterning

Problem of oxide steps: step coverage

Alignment problems: wafer-mask, mask-mask

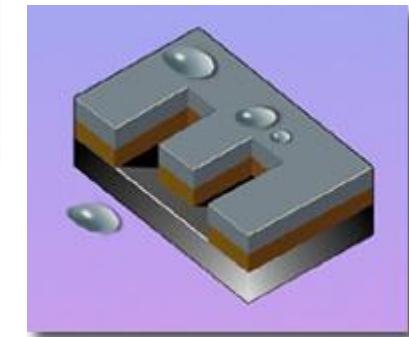
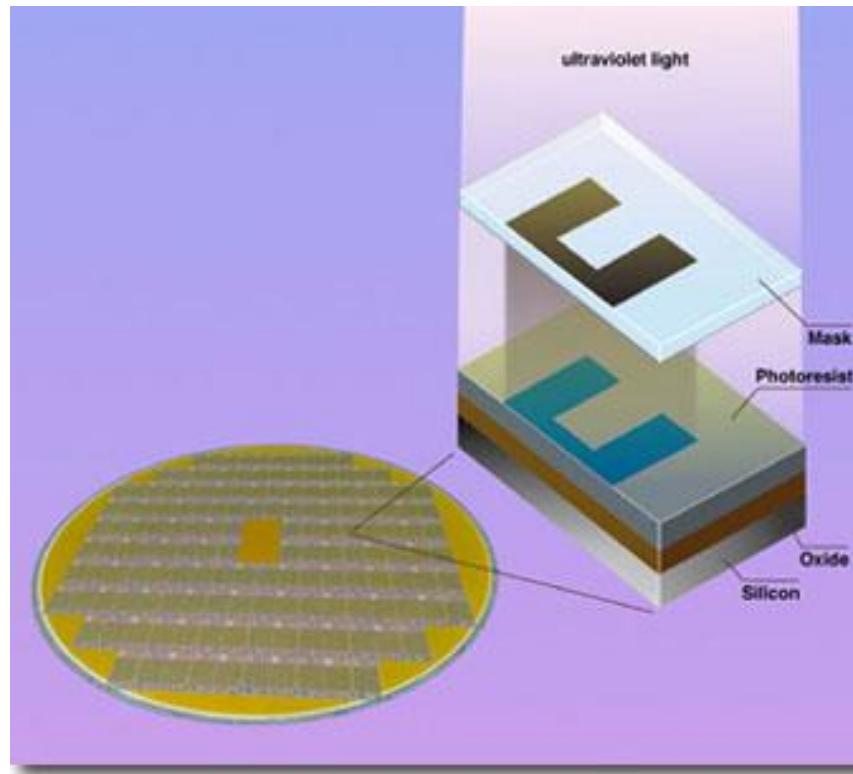
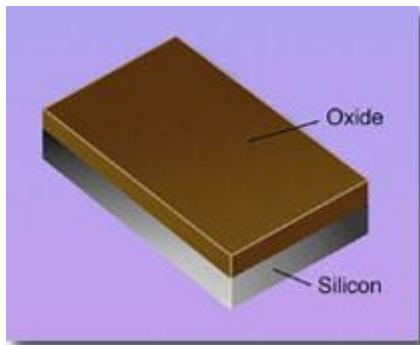
Diffraction

Standing wave effect



Window opening on the
oxide with
photolithography

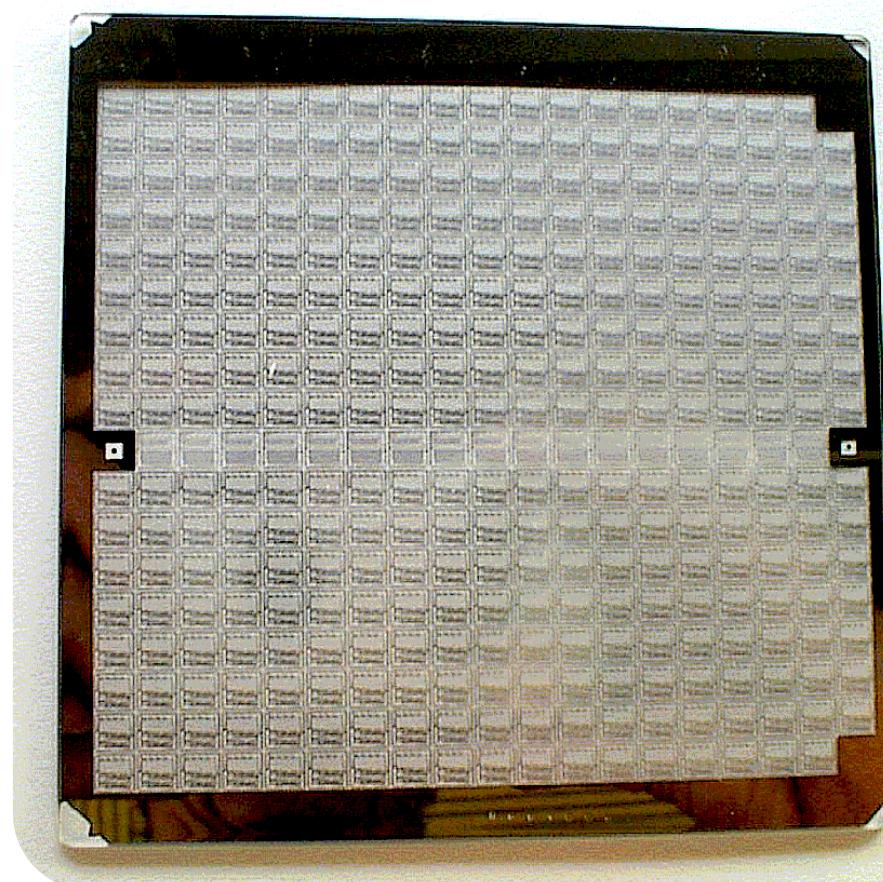
Patterning: photolithography



Patterning

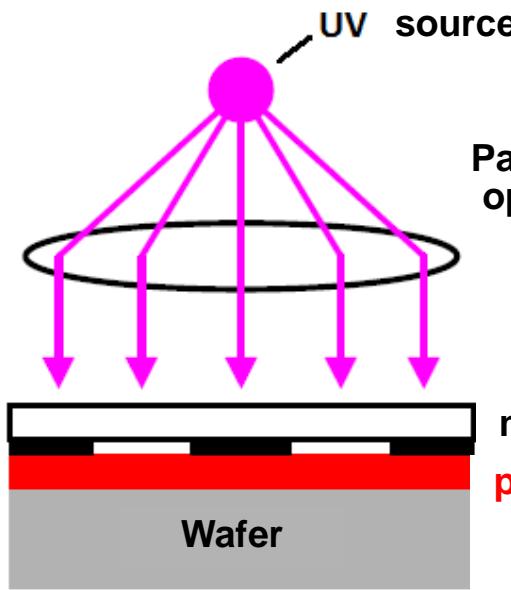
The original pattern is on a so called photo-mask

- made of chromium on glass substrate
 - many times larger than a chip
- Need for high level of accuracy:
- $0.03\mu\text{m}$ over 30cm!
 - 10^{-7}
- Visible light:
- $\lambda=0.3\text{-}0.6 \mu\text{m}$
 - deep UV needed!

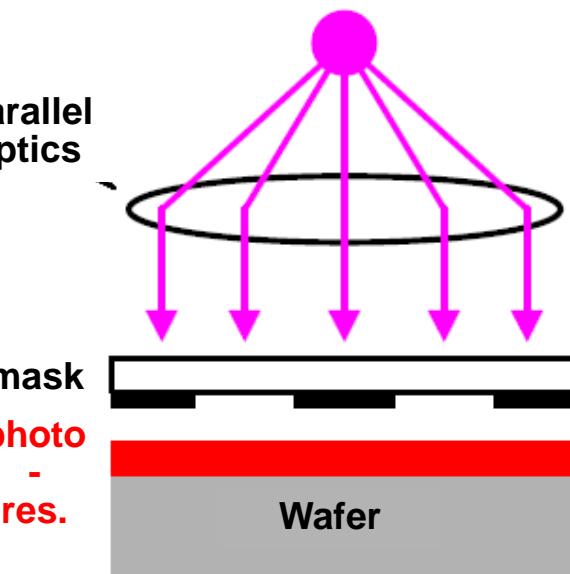


Mask alignment

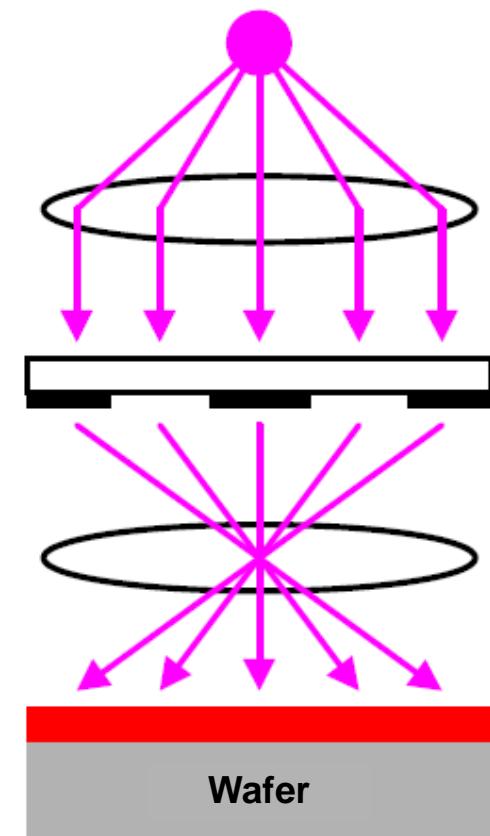
CONTACT



PROXIMITY



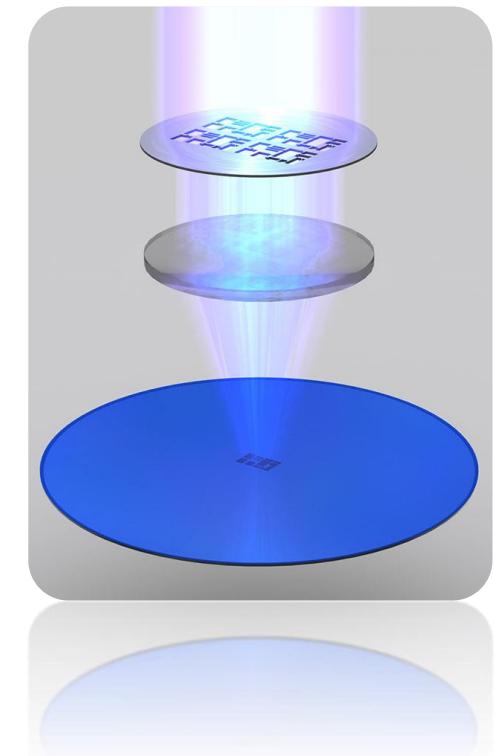
PROJECTION



Patterning: photolithography

Major steps:

- ▶ Thermal annealing (evaporation of the absorbed humidity)
- ▶ Adhesion enhancement (chemical procedure)
- ▶ **Photoresist** coating
- ▶ Drying (evaporation of the solvents from the photoresist) – soft bake
- ▶ Mask allignment and exposure
- ▶ Development
- ▶ Hard bake (curing the photoresist, further improvement of adhesion)



Photoresist

- ▶ 1..2um thickness (10um if it needs to be resistant)
- ▶ Polymer + photoactive component + solvent
- ▶ Solvent determines the viscosity of the mixture
 - Important at the spinning
- ▶ Negative or positive resist: If the photoactive component makes the polymer **easier** or **harder** to desolve during the exposition, then we call the photoresist **positive** or **negative**



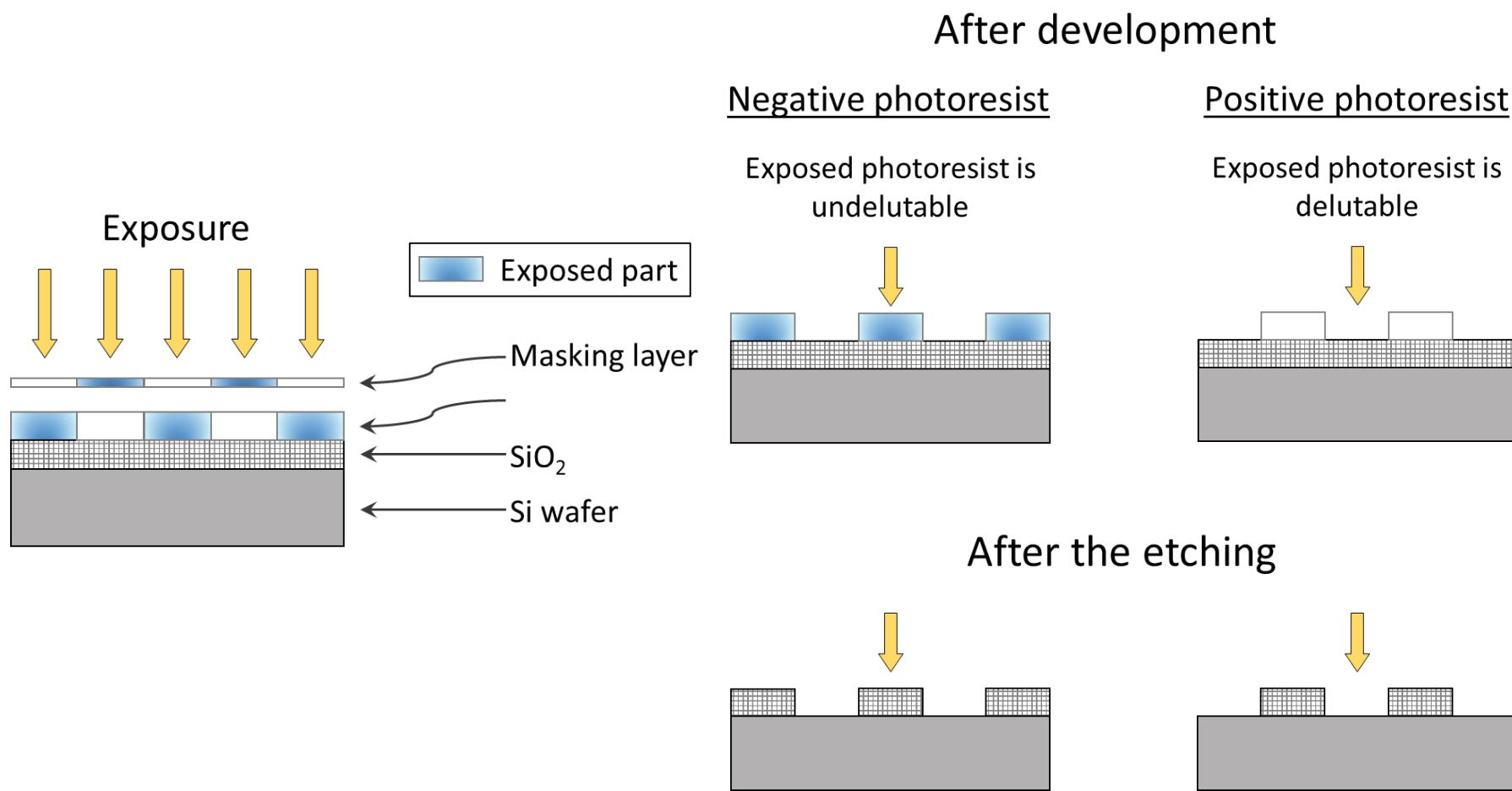
Photoresist

Properties:

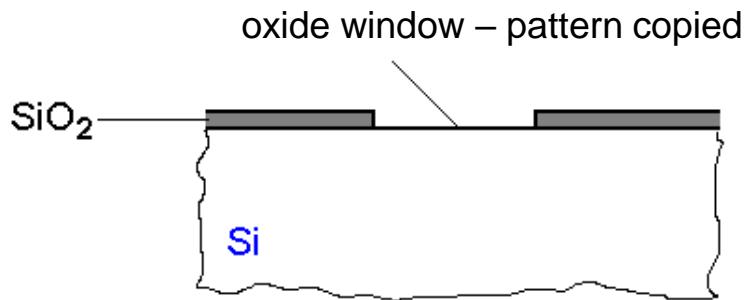
- spectral sensitivity,
- viscosity,
- lightsensitivity (relation of the absorbed photons and the transformed molecules)
- contrast (difference in dissolution velocity between the exposed and the shaded areas)
- resolution



Photolithography



The photolithography



E.g. metallization pattern:

1. deposit metal over the entire surface
2. coat with resist
3. UV photography through mask, develop
4. etch off unnecessary metal
5. remove resist

Modern photolithography

MFS=7nm
(2019)

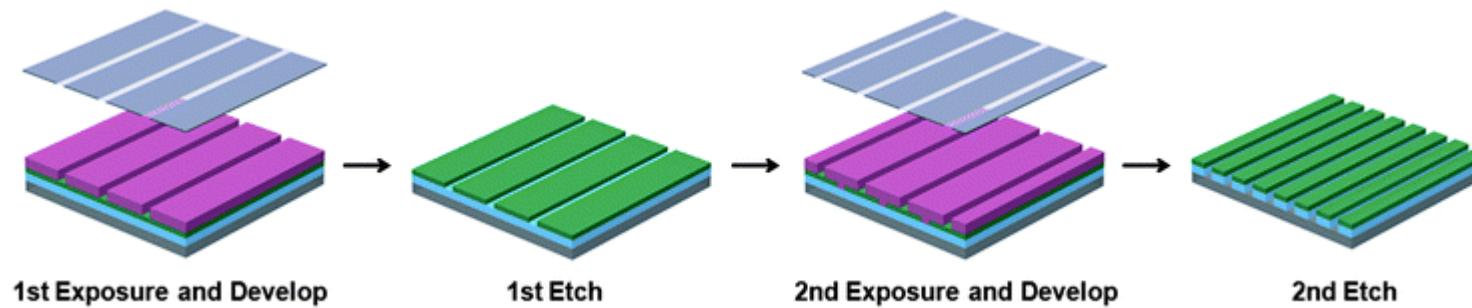
How to go below 193nm resolution?

► Immersion lithography

Liquid, usually purified multiple times, distilled water (NA increase)

► Multiple patterning

Exposure repeated several times



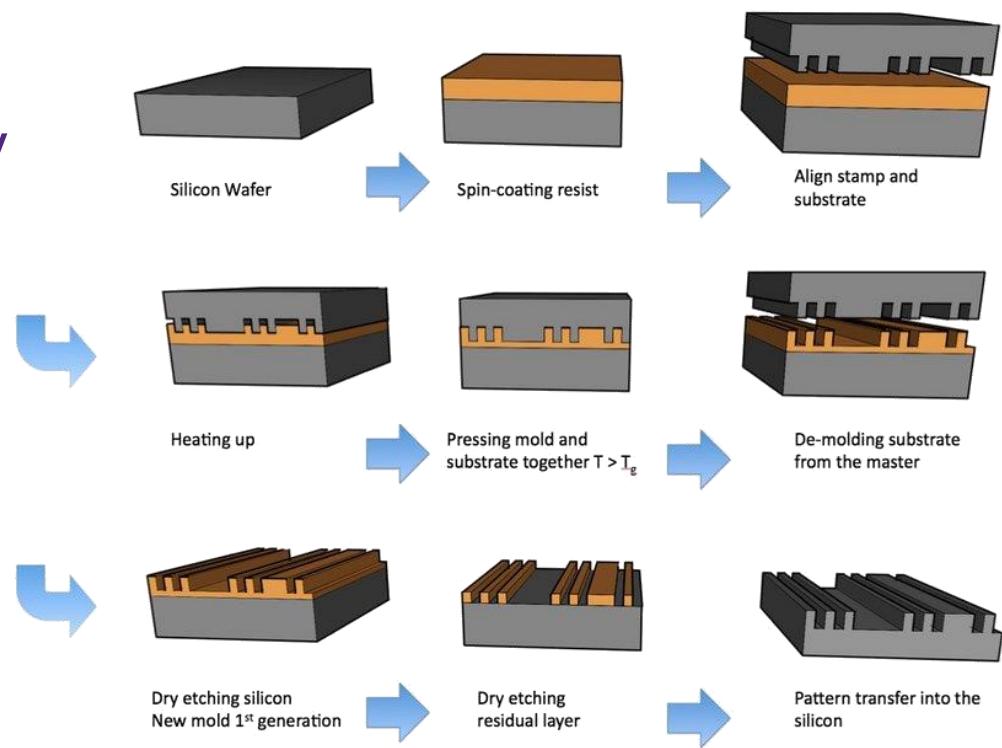
Modern photolithography

MFS=7nm
(2019)

How to go below 193nm resolution?

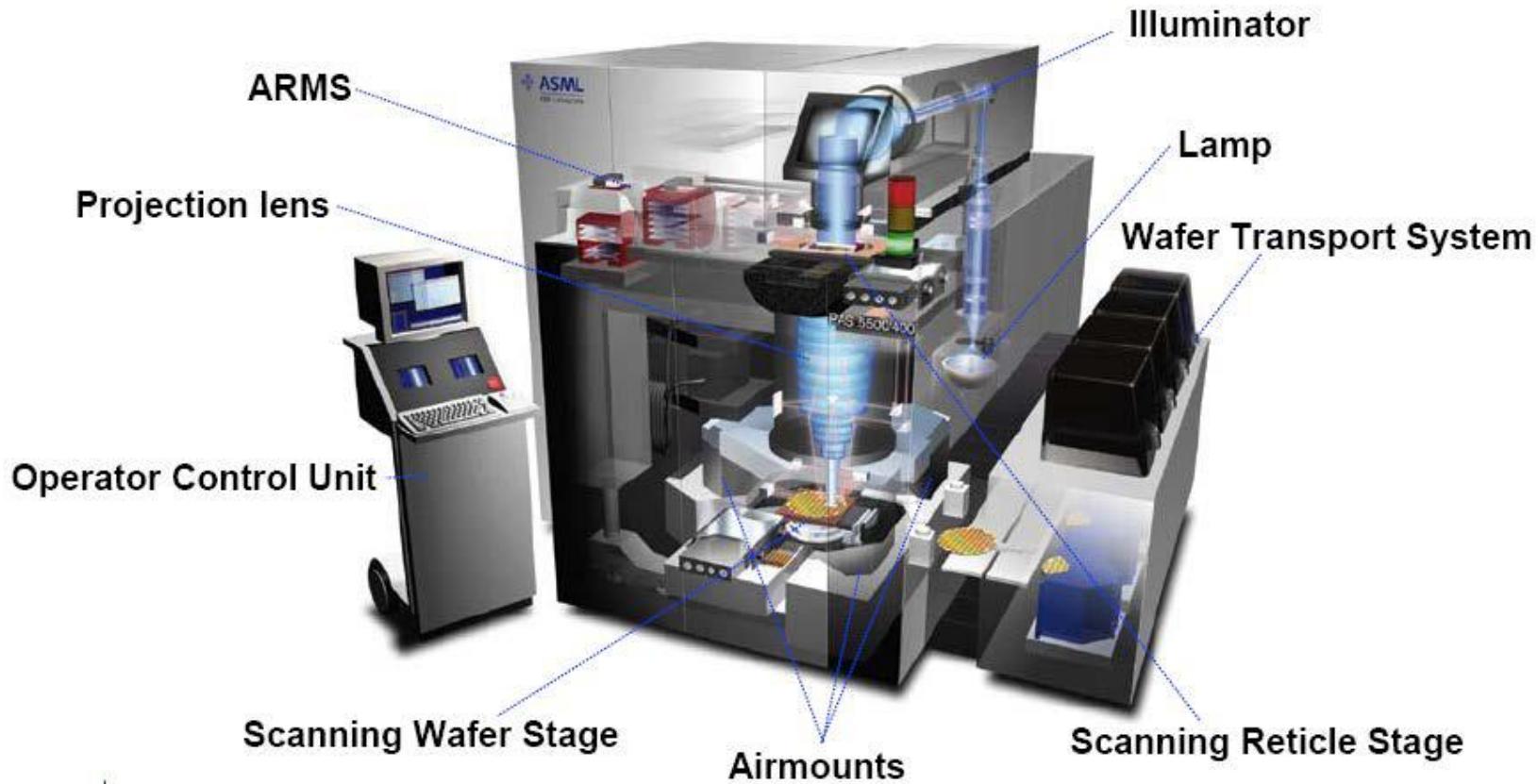
- ▶ Electron-Beam Direct-Write Lithography
 - resolution below 10nm but not on the whole wafer at the same time!
 - Slow!

- ▶ Nanoimprint lithography
 - contact lithography, printing template pressed and dried into a soft polymer (act as a „photoresist”)



Modern photolithography

Extreme Ultraviolet (EUV) lithography



ASML TWINSCAN NXE:3300B

Intel making of a chip



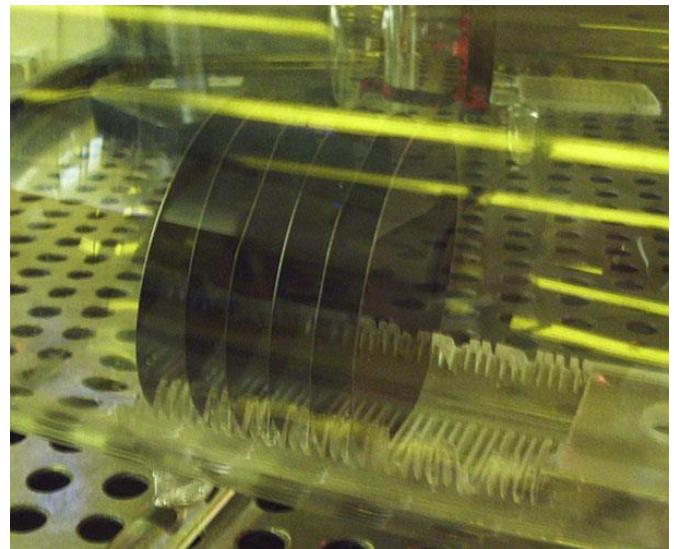
A simple pMOS process

Process at our cleanroom facility

The process steps



Steps of a simple pMOS process



Wafer cleaning

Steps of a simple pMOS process



Growth of thick SiO₂ (field oxide)

Steps of a simple MOS process



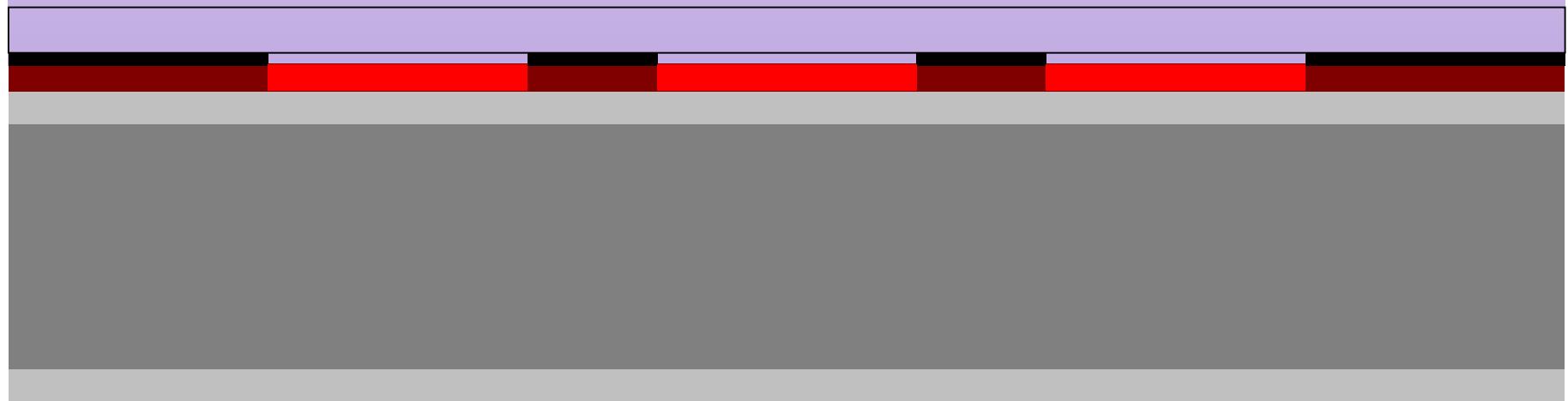
Photolithography: spin-coating with resist

Steps of a simple pMOS process



Photolithography: mask alignment

Steps of a simple pMOS process



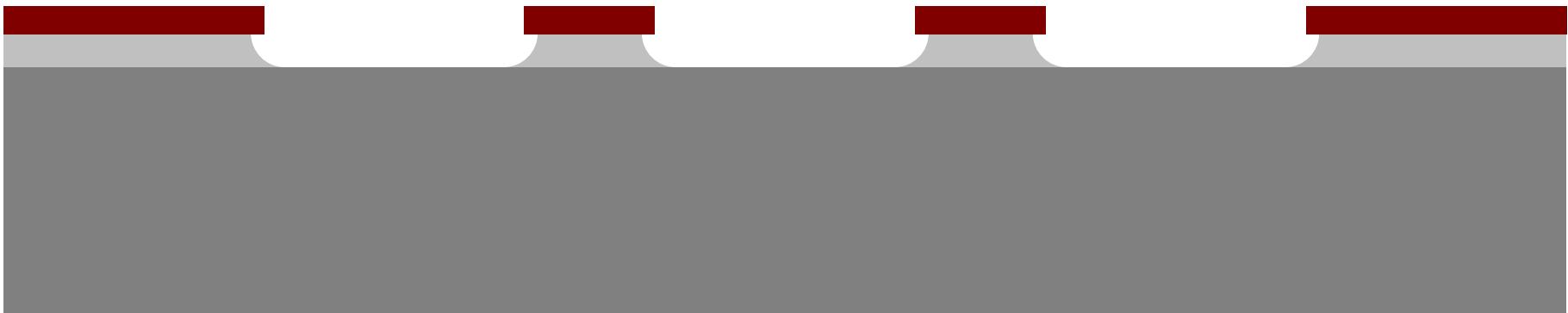
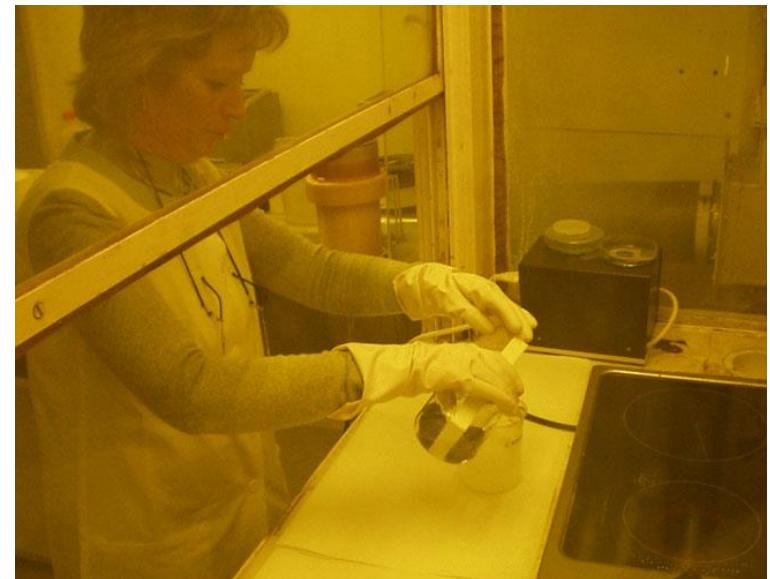
Photolithography: UV exposure

Steps of a simple pMOS process



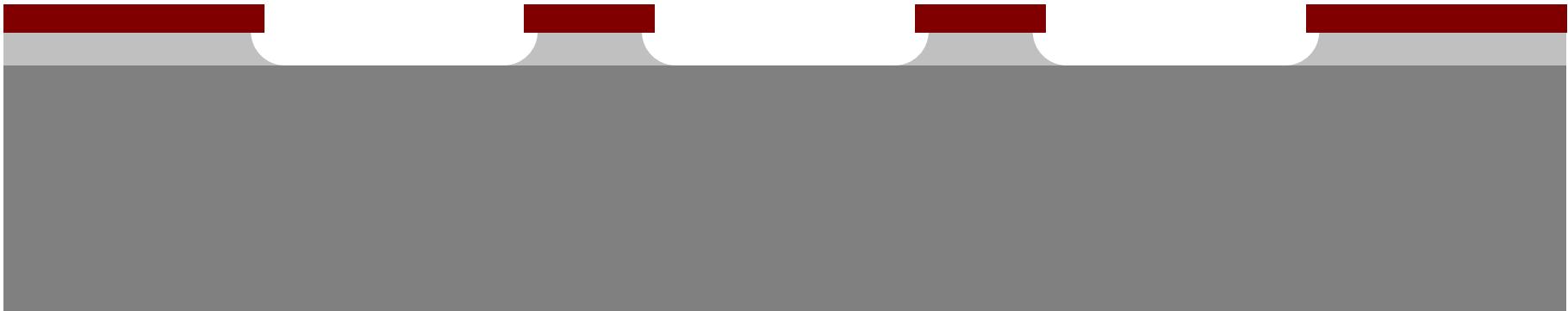
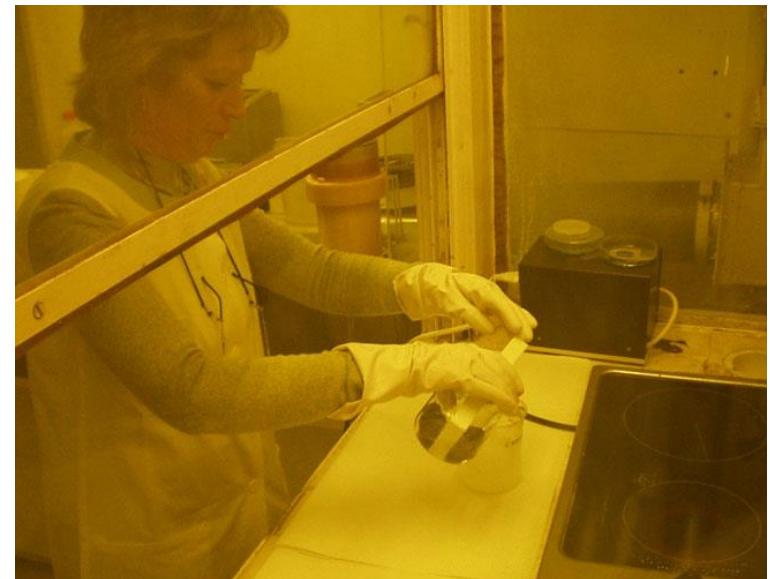
Photolithography: development

Steps of a simple pMOS process



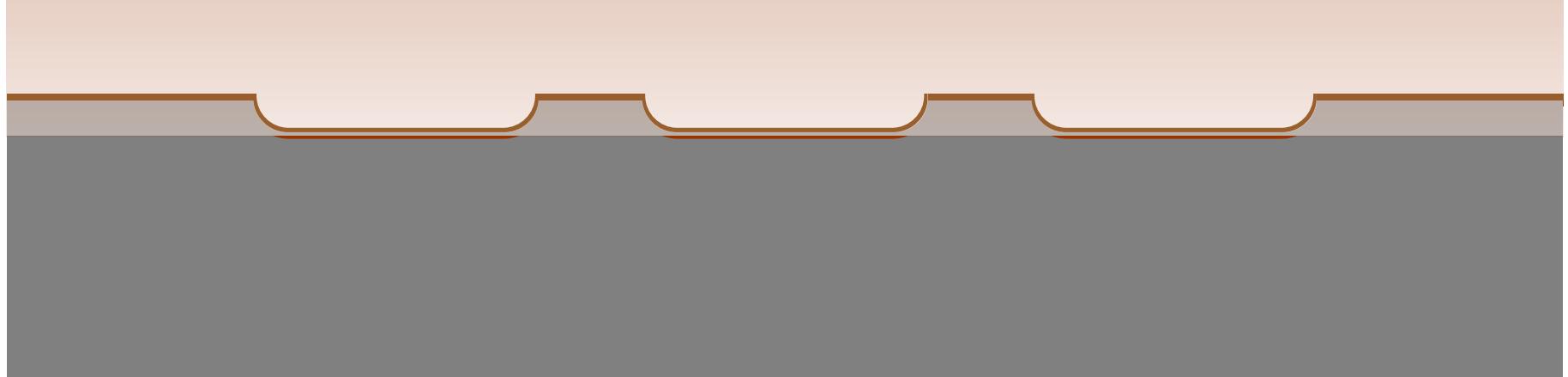
Patterning: oxide etching

Steps of a simple pMOS process



Patterning: oxide etching, resist removal

Steps of a simple pMOS process



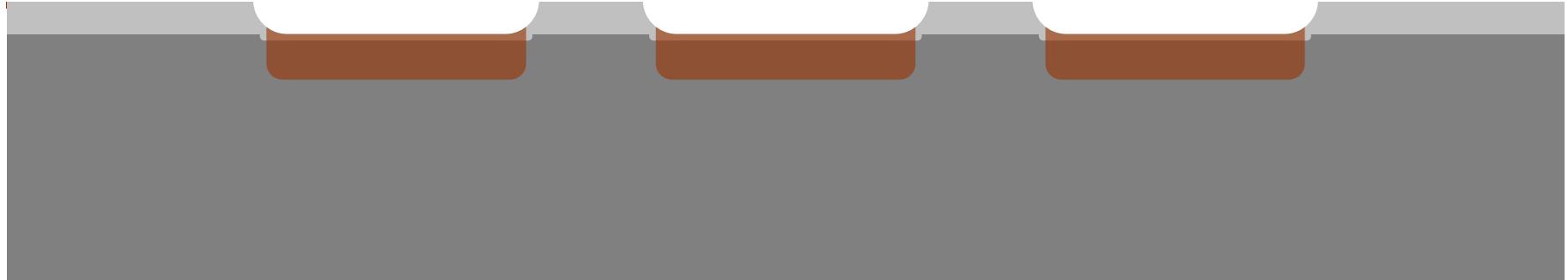
Diffusion from solid boron (pre-diffusion)

Steps of a simple pMOS process



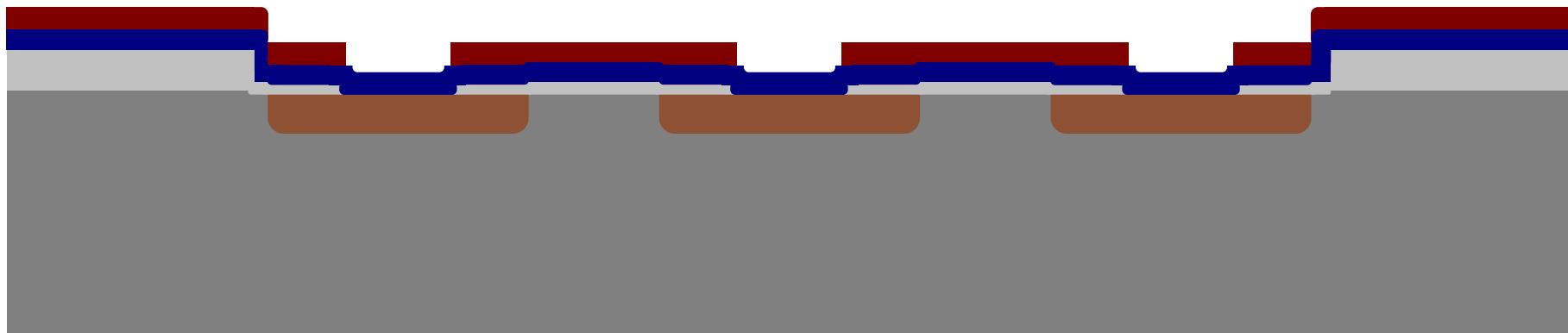
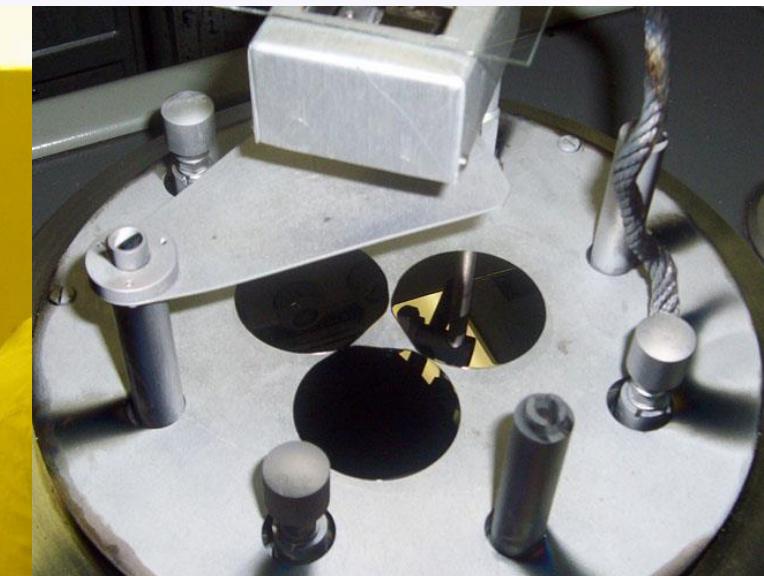
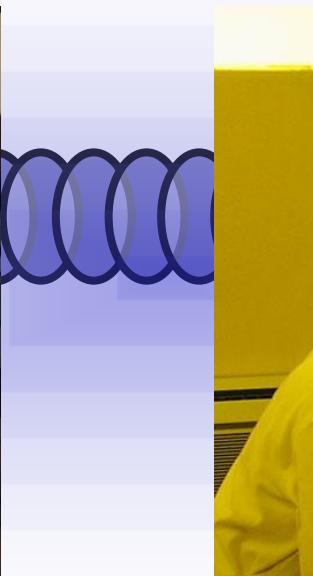
Removal of boron glass

Steps of a simple pMOS process



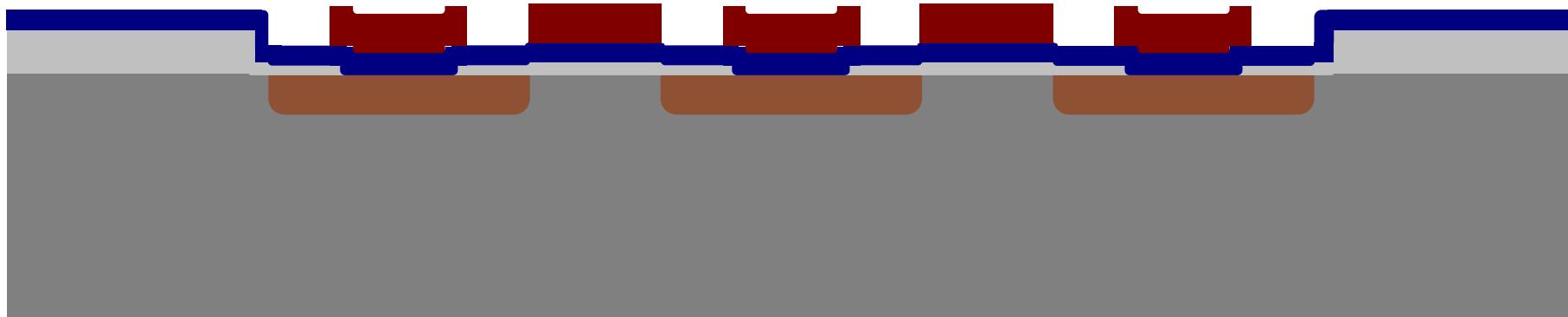
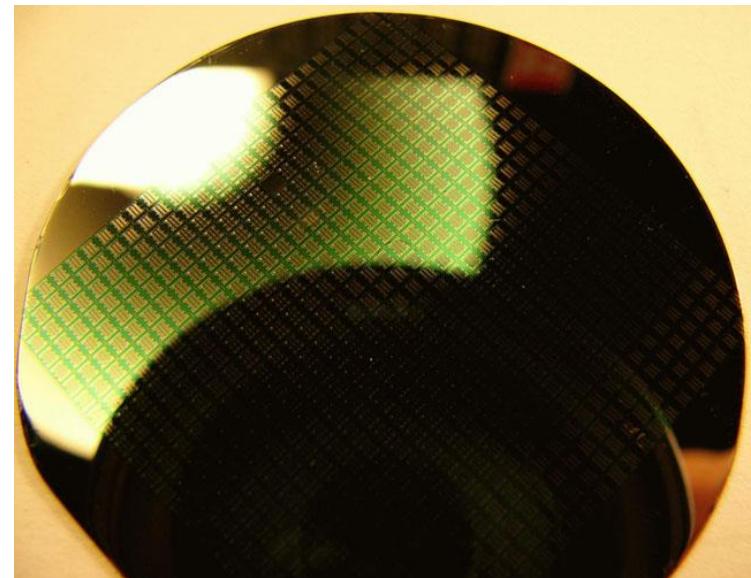
Boron diffusion, 2nd step: driving in (in oxygen)

Steps of a simple pMOS process



Electrostatically controlled plasma polymerization (ECCP) of silicon gate oxide

Steps of a simple pMOS process



Wafer lithography - patterning and development

Steps of a simple pMOS process



Dicing, bonding

Pulvis et umbra sumus.

We are but dust and shadow.
Horace, 65-8 BC, Roman poet





Budapest University of Technology and Economics
Department of Electron Devices

Microelectronics, BSc course

Basic semiconductor physics

Summary of the essential semiconductor physics

- Charge carriers in semiconductors
- Currents in semiconductors
- Generation, recombination; continuity equation

Energy bands in the lattice

- Basic consequences of quantum mechanics:
 - Positive nucleus (p^+ , n^0) – Negative electrons
 - **Pauli theorem:** two e^- cannot be on the same state; in one electron orbital max two e^- with opposite spin
 - Energy minimum: lowest shells, orbitals are filled
 - Switching between shells: energy gained (equals the energy between the actual state and the unfilled shells)
 - It is most likely that the e^- with upper most state will gain more energy



Discrete
energy
levels

Energy bands

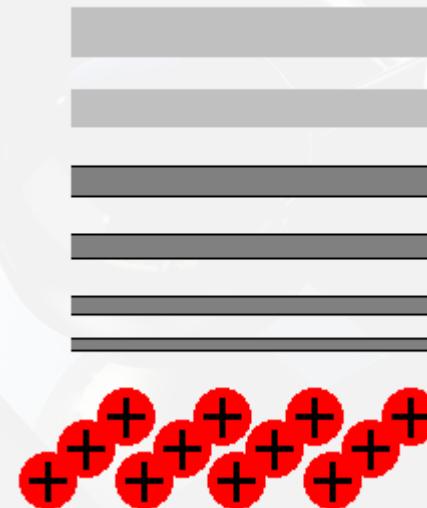
- Resulting from principles of quantum physics

Discrete energy levels:



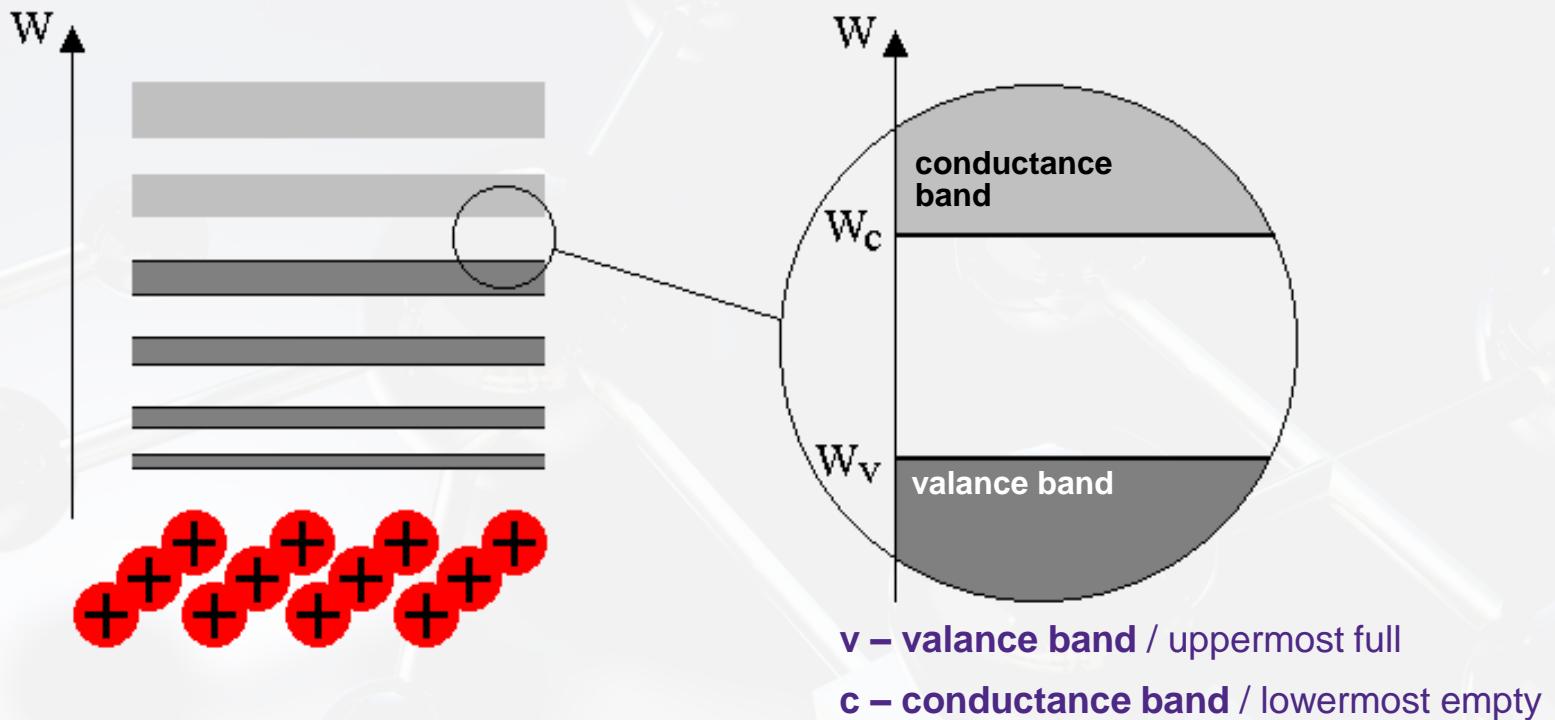
More atoms – more energy levels of electrons:

There are so many of them that they form energy bands:



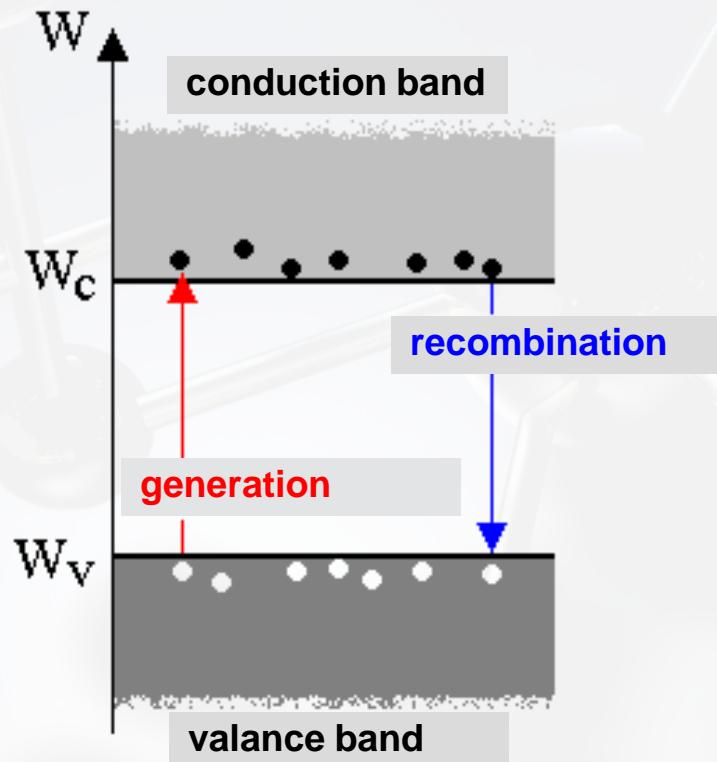
The discrete (allowed) energy levels of an atom become energy bands in a crystal lattice

Valence band, conductance band



- Valence band – these electrons form the chemical bonds
 - almost full
- Conductance band – electrons here can move freely
 - almost empty

Electrons and holes

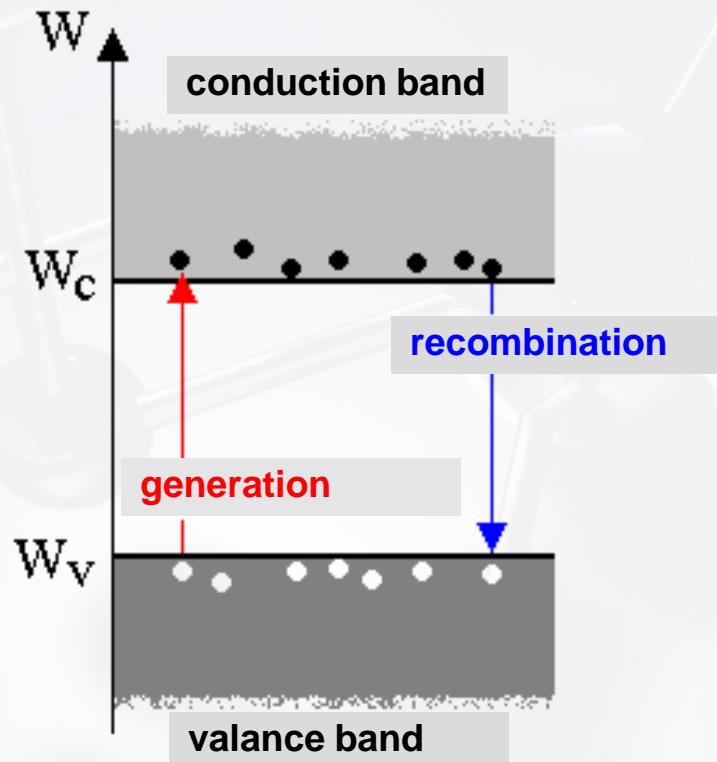


- Generation: using the average thermal energy
- **Electrons:** in the bottom of the conduction band
- **Holes:** in the top of the valence band
- Both the electrons and the holes form the electrical current

Electron: negative charge, positive eff. mass

Hole: positive charge, positive eff. mass

Electrons and holes

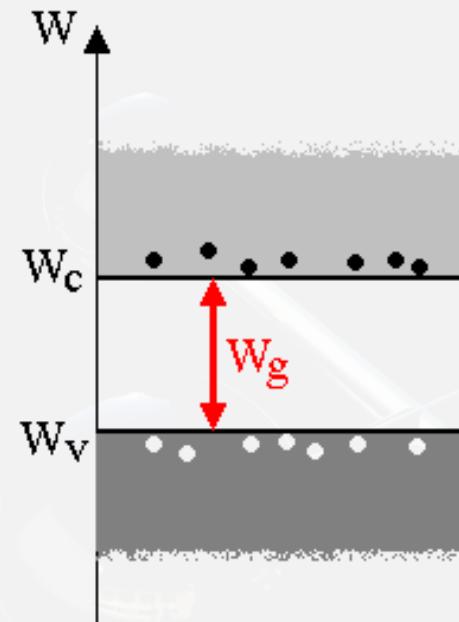


How does the electrons gain energy?

- ▶ Thermal excitation
 - Energy of lattice resonance transferred to the e^-
- ▶ Photon excitation
 - Incident light with energy greater than W_g
 - $E = h \cdot v$

The electron density of the conductance band is determined by the dynamic balance of the generation and the recombination

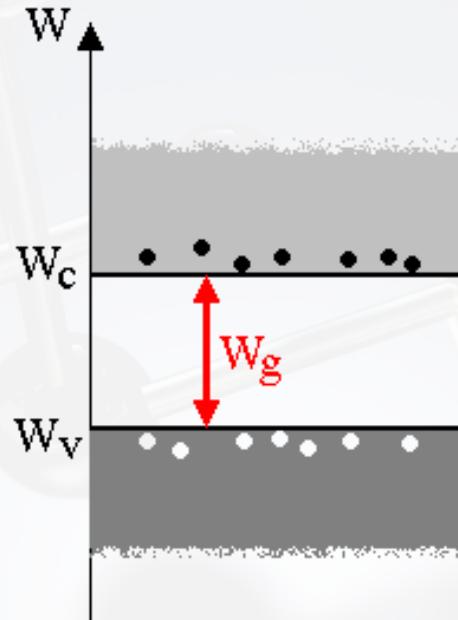
Conductors and insulators



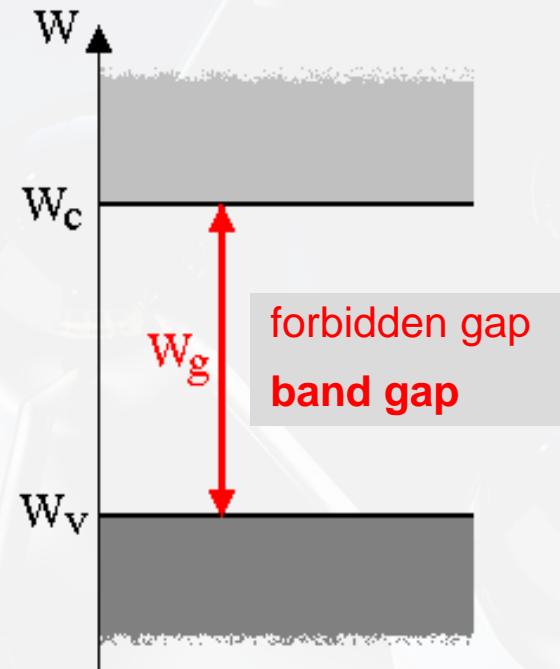
semiconductor

Electron cannot be located in the forbidden gap!

Conductors and insulators



semiconductor



insulator



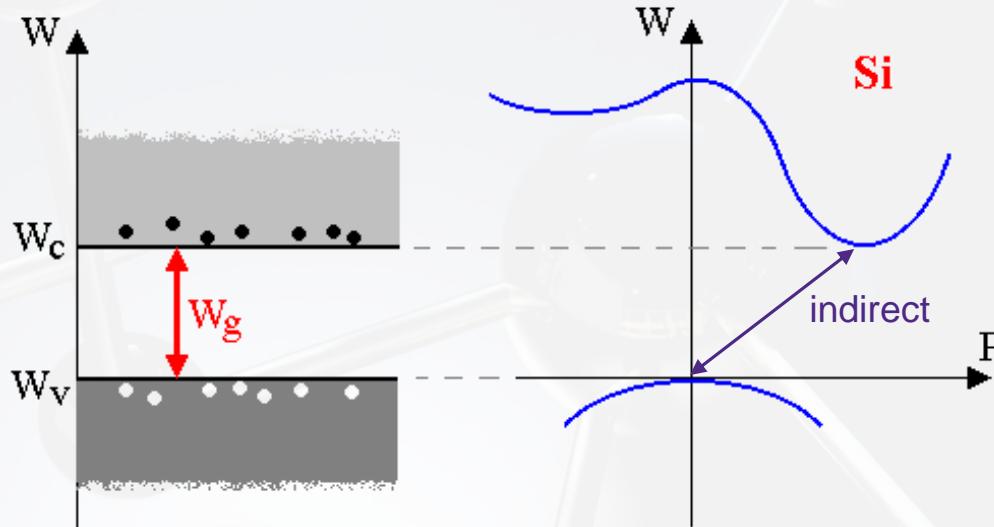
metal

For Si: $W_g = 1.12 \text{ eV}$

for SiO_2 : $W_g = 4.3 \text{ eV}$

$$1 \text{ eV} = 0.16 \text{ aJ} = 0.16 \cdot 10^{-18} \text{ J}$$

Band structure of semiconductors

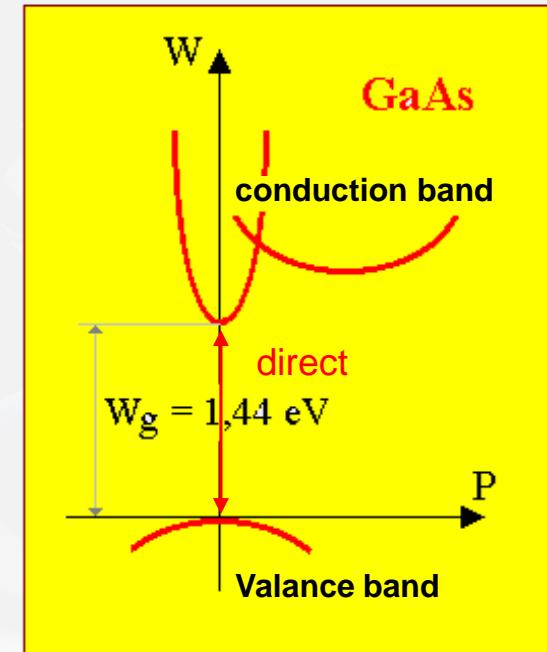


In case of indirect band semiconductor the laws of energy and impulse conservations must be fulfilled.

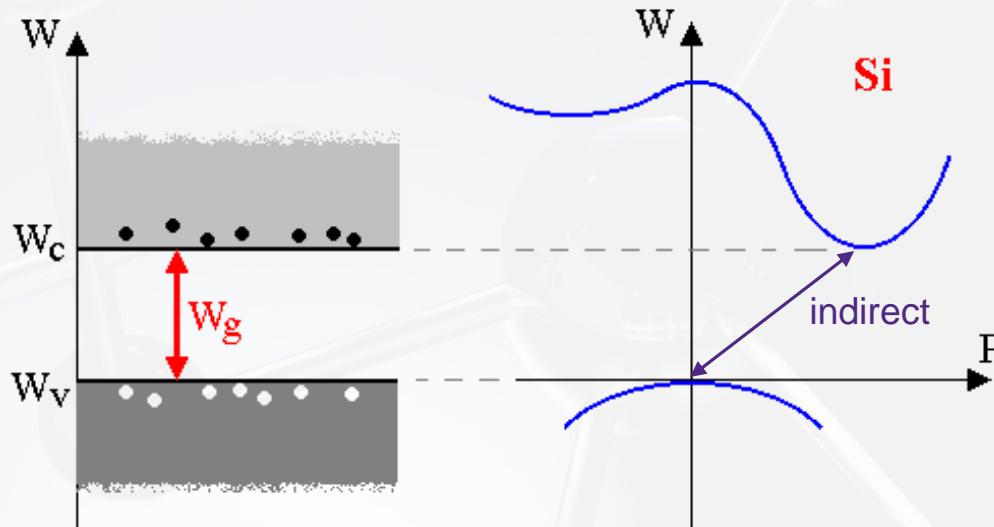
$$F = \frac{dP}{dt}$$

$$P = \frac{\hbar}{2\pi} k$$

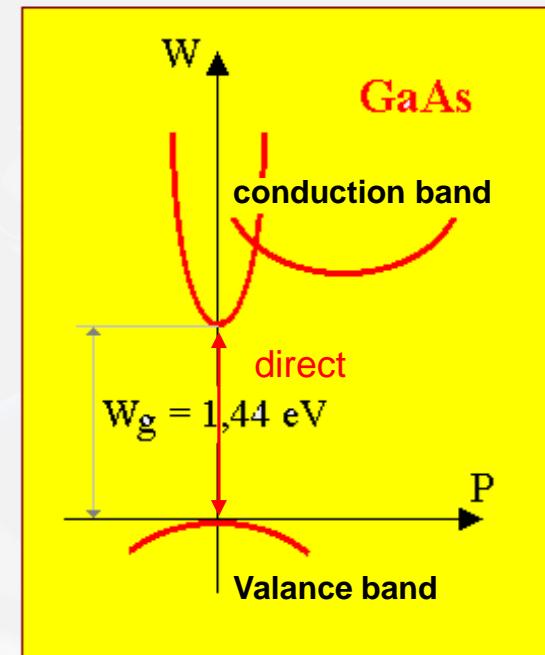
GaAs: direct band \Rightarrow opto-electronic devices (LEDs)
Si: indirect band



Band structure of semiconductors



$$W = \frac{1}{2m} p^2 \rightarrow W = \frac{1}{2m_{eff}} P^2$$



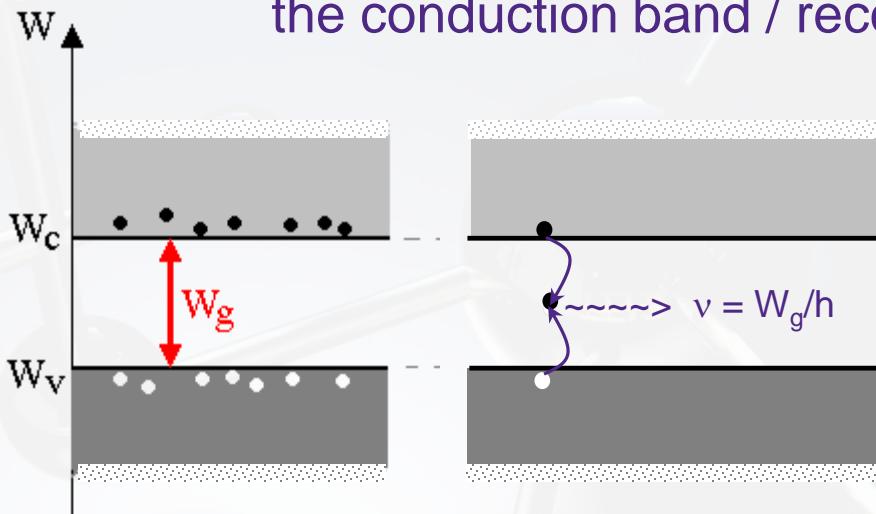
$$F = \frac{dP}{dt}$$

$$P = \frac{\hbar}{2\pi} k$$

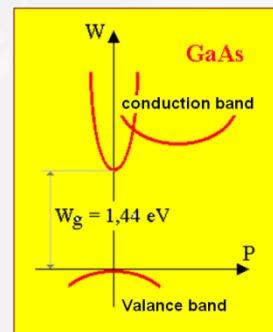
GaAs: direct band \Rightarrow opto-electronic devices (LEDs)
Si: indirect band

Generation / recombination

Spontaneous process: thermal excitation – jump into the conduction band / recombination \rightarrow equilibrium



Direct **recombination**
may result in light
emission (LEDs)



$$\nu = \frac{W_g}{h}$$

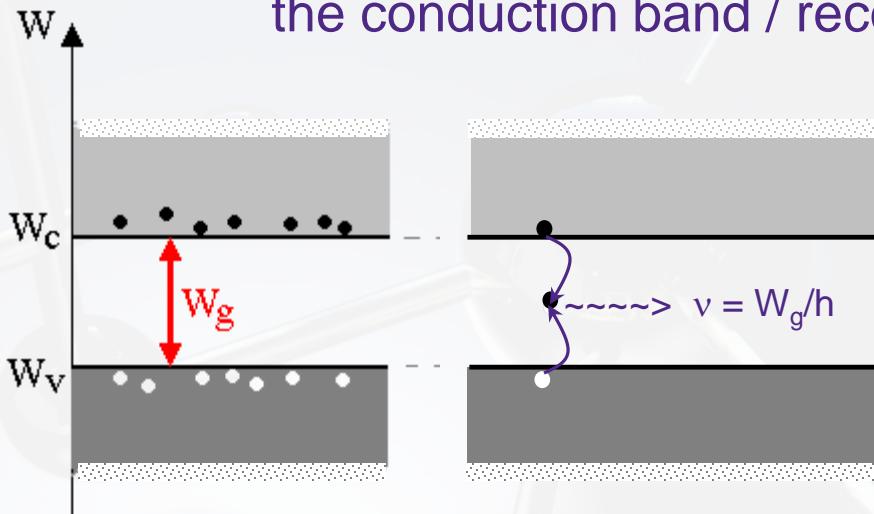
$$\lambda = \frac{c}{\nu}$$

In case of red laser
 $W_g \approx 1.5 \text{ eV}$

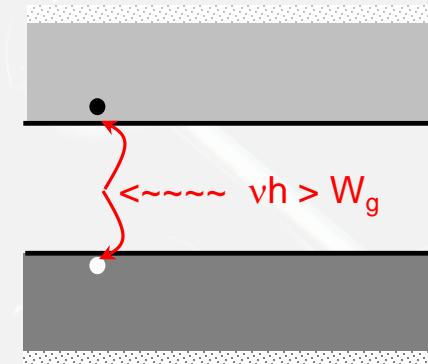
Indirect **recombination** is not radioactive, but always accompanied by heat dissipation (Phonon – lattice vibration)

Generation / recombination

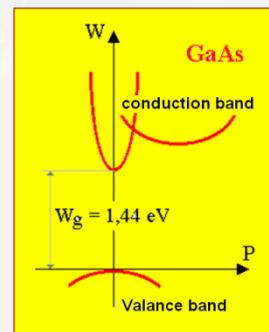
Spontaneous process: thermal excitation – jump into the conduction band / recombination \rightarrow equilibrium



Direct **recombination**
may result in light
emission (LEDs)



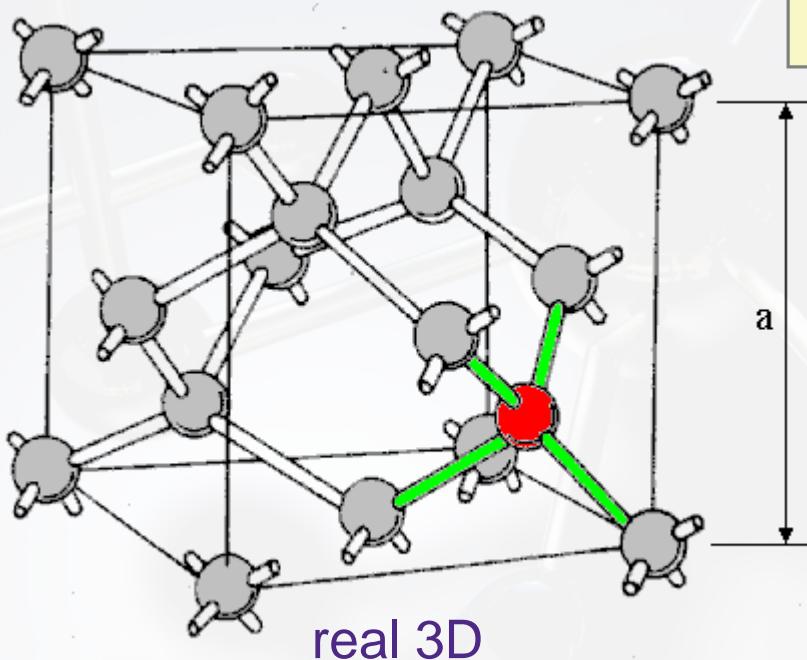
Light absorption results
in **generation** (solar
cells)



Experiment

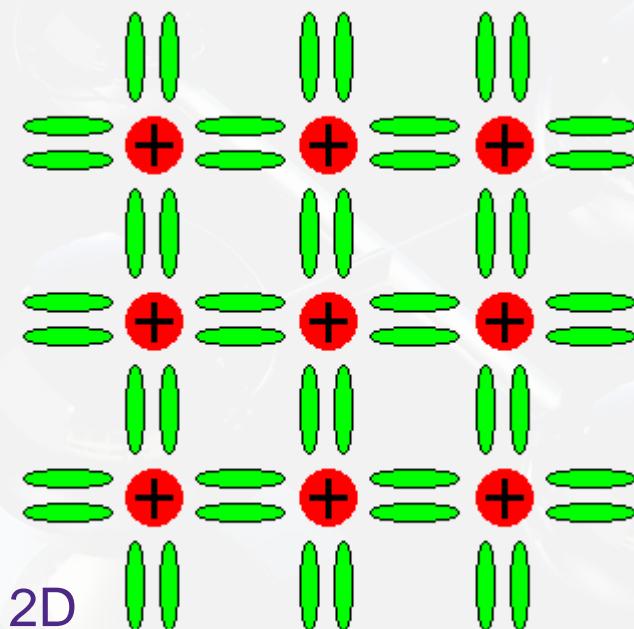
Crystalline structure of Si

- Si N=14 4 bonds, IV-th column of the periodic table



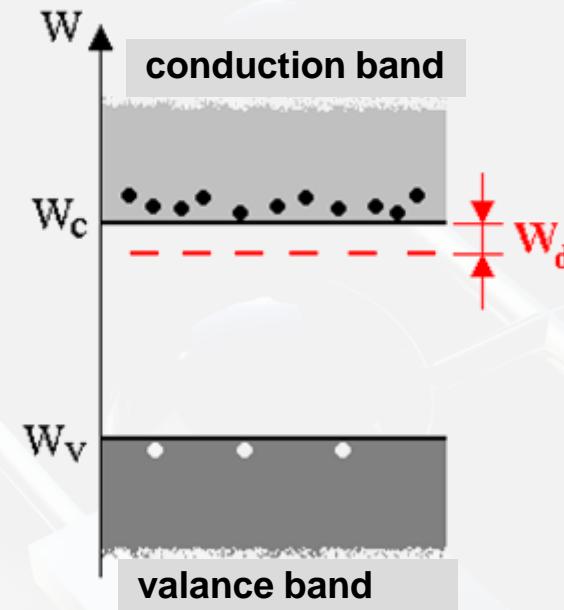
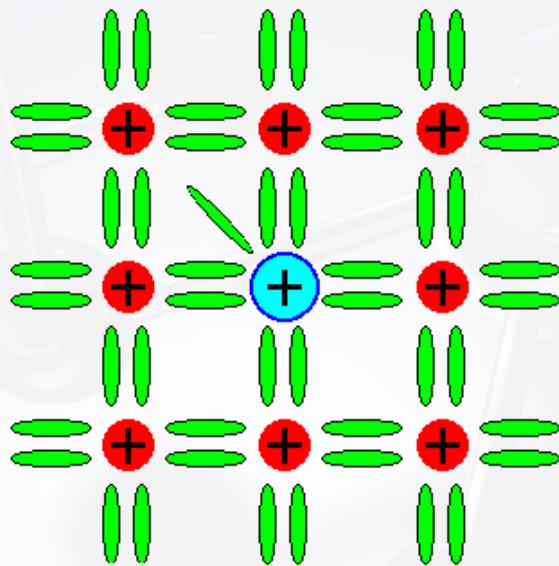
undoped or intrinsic semiconductor

simplified 2D



- Diamond lattice, *lattice constant $a=0.543$ nm*
- Each atom has 4 nearest neighbor

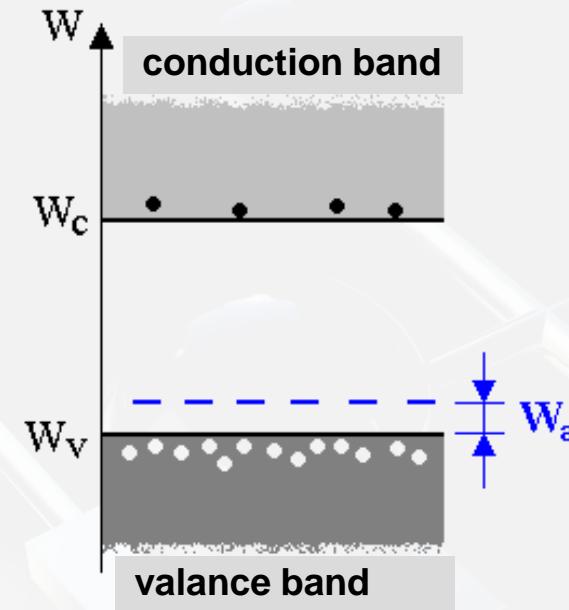
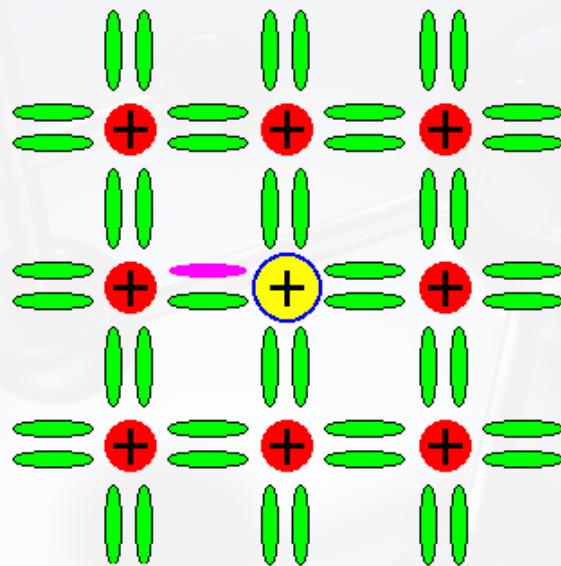
5 valence dopant: donor (As, P, Sb)



- **Electron:** majority carrier
- **Hole:** minority carrier

n-type semiconductor

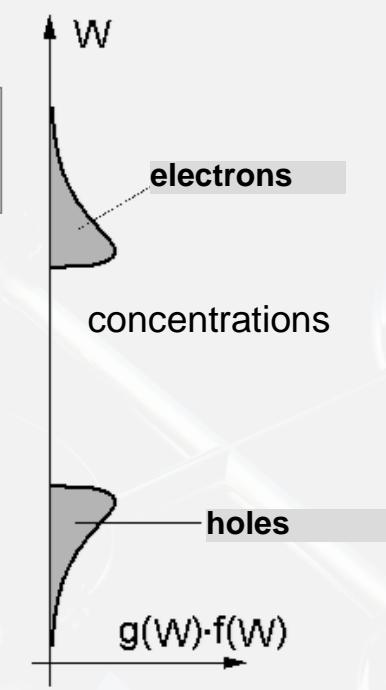
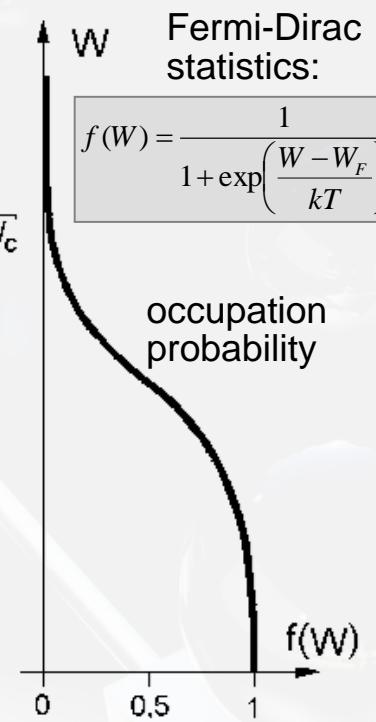
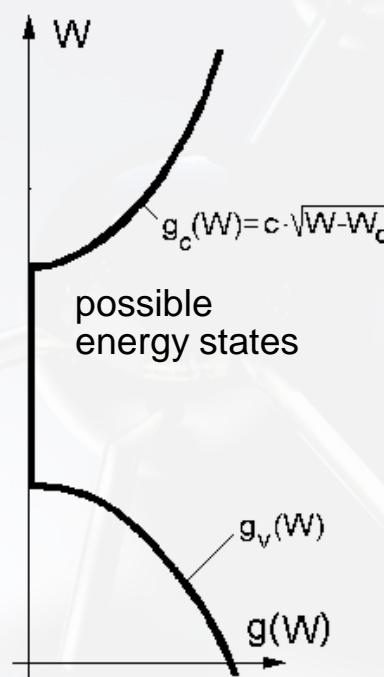
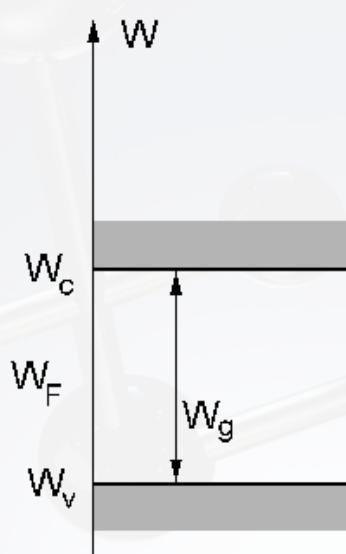
3 valence dopant: acceptor (B, Ga, In)



- Electron: minority carrier
- Hole: majority carrier

p-type semiconductor

Calculation of carrier concentration



$$n = \int_{W_c}^{\infty} g_c(W) f(W) dW$$

$$p = \int_0^{W_v} g_v(W) [1 - f(W)] dW$$

Calculation of carrier concentration

- The results is:

$$n = \text{const } T^{3/2} \exp\left(-\frac{W_c - W_F}{kT}\right)$$

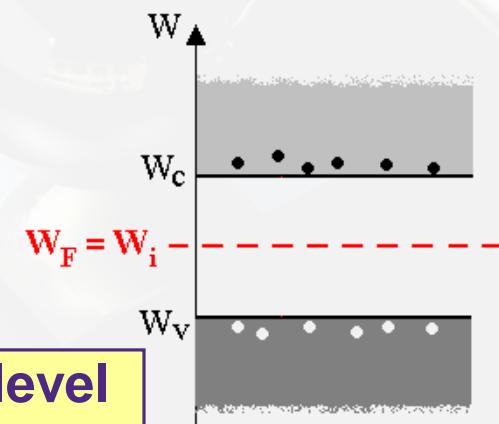
$$p = \text{const } T^{3/2} \exp\left(-\frac{W_F - W_v}{kT}\right)$$

- If there is no doping: $n = p = n_i$
 - it is called ***intrinsic*** material

$$W_c - W_F = W_F - W_v$$

$$W_F = \frac{W_c + W_v}{2} = W_i$$

W_F: Fermi-level

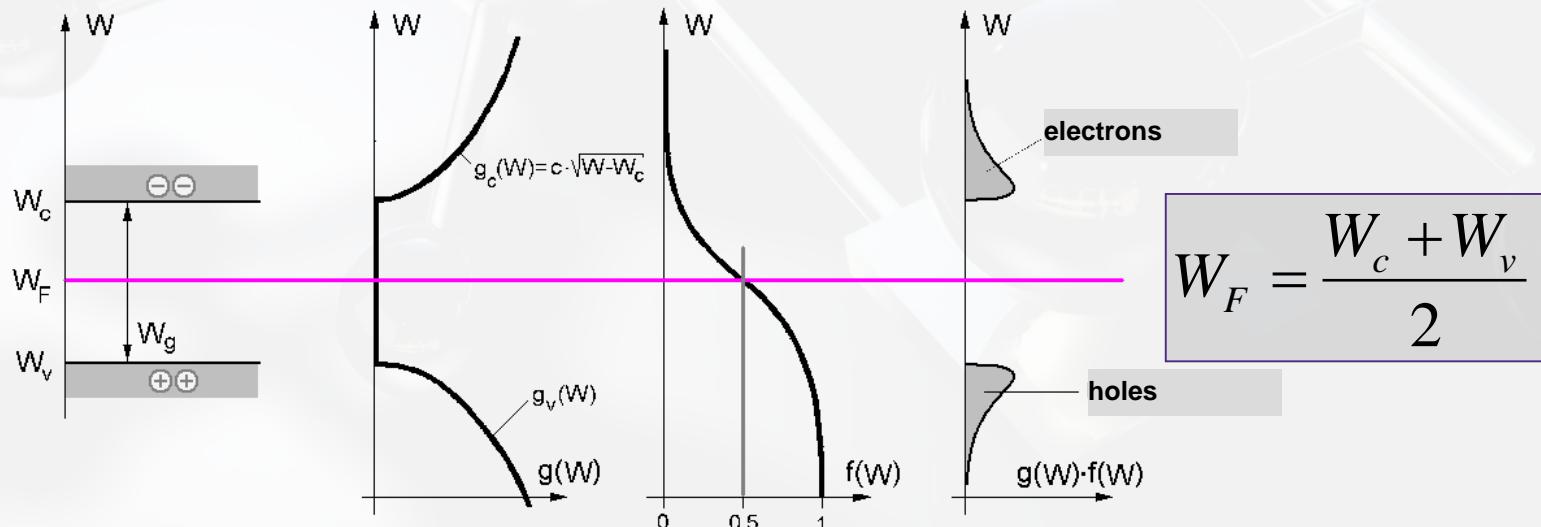


The Fermi-level

- Formal definition of the Fermi-level: the energy level where the probability of occupancy is 0.5:

$$f(W) = \frac{1}{1 + \exp\left(\frac{W - W_F}{kT}\right)} = 0.5$$

- In case of intrinsic semiconductor this is in the middle of the band gap:



- This is the intrinsic Fermi-level W_i

Carrier concentrations

$$n = \text{const} T^{3/2} \exp\left(-\frac{W_c - W_F}{kT}\right)$$

$$p = \text{const} T^{3/2} \exp\left(-\frac{W_F - W_v}{kT}\right)$$

$$n \cdot p = \text{const} \cdot T^3 \exp(-W_g / kT)$$

- **Depends on temperature only**, does not depend on doping concentration

$$n \cdot p = n_i^2$$

Mass action law

For silicon at 300 K absolute temperature

$$n_i = 10^{10} / \text{cm}^3$$

(10^{10} electrons in a cube of size
0.01 mm x 0.01 mm x 0.01 mm)

Carrier concentrations

Problem

- Si, T = 300 K, donor concentration $N_D = 10^{17} \text{ /cm}^3$
- What are the hole and the electron concentrations?
 - Donor doping $\Rightarrow n \approx N_D = \underline{\underline{10^{17} \text{ /cm}^3}}$
 - Hole concentration: $p = n_i^2/n = 10^{20}/10^{17} = \underline{\underline{10^3 \text{ /cm}^3}}$
- What is the relative density of the dopants?
 - 1 cm³ Si contains $5 \cdot 10^{22}$ atoms
 - thus, $10^{17} / 5 \cdot 10^{22} = \underline{\underline{2 \cdot 10^{-6}}}$
 - The purity of doped Si is 0.999998

Carrier concentrations

$$n = \text{const } T^{3/2} \exp\left(-\frac{W_c - W_F}{kT}\right)$$

$$n_i = \text{const } T^{3/2} \exp\left(-\frac{W_c - W_i}{kT}\right)$$

$$\frac{n}{n_i} = \exp\left(\frac{W_F - W_i}{kT}\right)$$

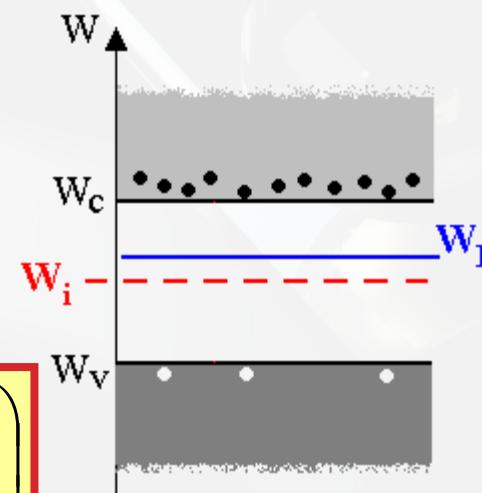
$$n = n_i \exp\left(\frac{W_F - W_i}{kT}\right)$$

$$p = n_i \exp\left(-\frac{W_F - W_i}{kT}\right)$$

- Just re-order the equations

**kT = 1.38·10⁻²³ VAs/K · 300
 K = 4,14 · 10⁻²¹ J = 0.026 eV
 = 26 meV**

Thermal energy



In doped semiconductors the Fermi-level is shifted wrt the intrinsic Fermi-level

Temperature dependence

$$n_i^2 = n \cdot p = const \cdot T^3 \exp(-W_g / kT)$$

$$\frac{d}{dT} n_i^2 = n_i^2 \left(\frac{3}{T} + \frac{W_g}{kT^2} \right)$$

$$\frac{d n_i^2}{n_i^2} = \left(3 + \frac{W_g}{kT} \right) \frac{dT}{T}$$

► How strong is it for Si?

Problem

$$\frac{d n_i^2}{n_i^2} = \left(3 + \frac{1,12}{0,026} \right) \frac{dT}{300} \approx 0.15 dT \approx 15\% / ^\circ\text{C}$$

Temperature dependence of carrier concentrations

Problem

Si, T = 300 K, donor dopant concentration $N_D = 10^{17} \text{ /cm}^3$

$$n \approx N_D = 10^{17} \text{ /cm}^3$$

$$p = n_i^2 / n = 10^{20} / 10^{17} = 10^3 \text{ /cm}^3 \quad \Leftarrow \quad n \cdot p = n_i^2$$

How do n and p change, if T is increased by 25 degrees?

$$n \approx N_D = 10^{17} \text{ /cm}^3 - \underline{\text{unchanged}}$$

$$n_i^2 = 10^{20} \cdot 1.15^{25} = 33 \cdot 10^{20}$$

$$\Rightarrow p = n_i^2 / n = 33 \cdot 10^{20} / 10^{17} = 3.3 \cdot 10^4 \text{ /cm}^3$$

Only the minority carrier concentration increased!

$$\Delta T = 16.5 \text{ }^\circ\text{C} \rightarrow 10 \times$$

Currents in semiconductors

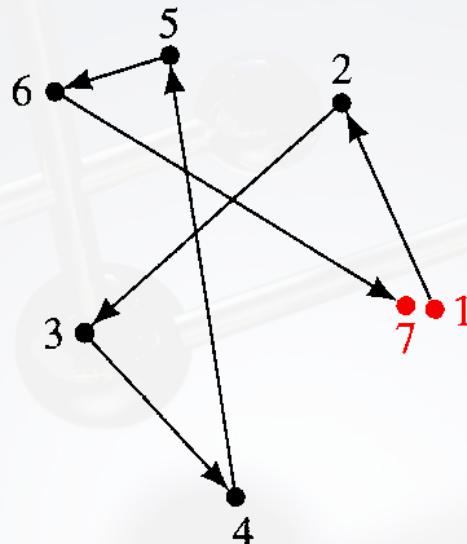
- Drift current
- Diffusion current

Not discussed:

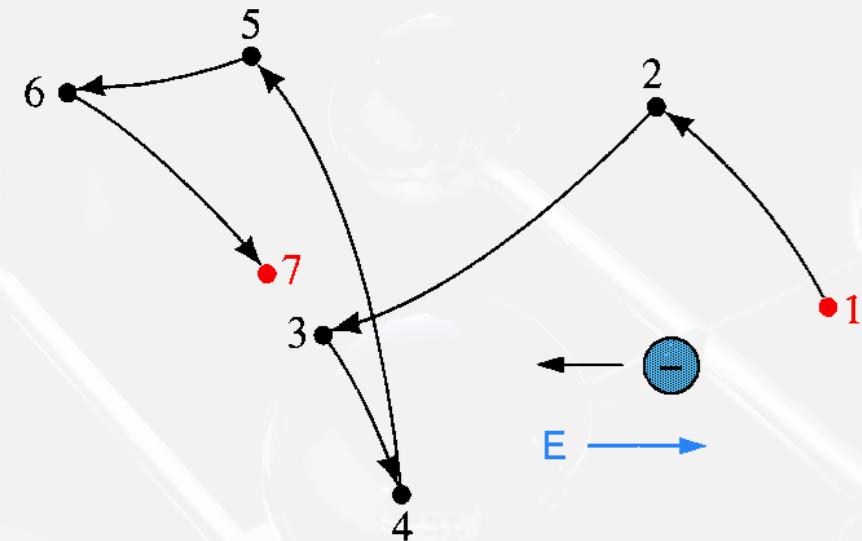
- currents due to temperature gradients
- currents induced by magnetic fields
- energy transport besides carrier transport
- combined transport phenomena

Drift current

Thermal movement of electrons



No electrical field



There is E electrical field

$$\bar{v}_s = \mu \bar{E}$$

μ = mobility

m^2/Vs

Drift current

$$\bar{J} = \rho \bar{v}$$

ρ charge density
 v velocity (average)

$$\bar{v}_s = \mu \bar{E}$$

$$\bar{J}_n = q n \mu_n \bar{E}$$

$$\bar{J}_p = q p \mu_p \bar{E}$$

$$\bar{J} = q(n \mu_n + p \mu_p) \bar{E}$$

$$\bar{J} = \sigma_e \bar{E}$$

Differential
 Ohm's law

$$\rho_e = \frac{1}{\sigma_e}$$

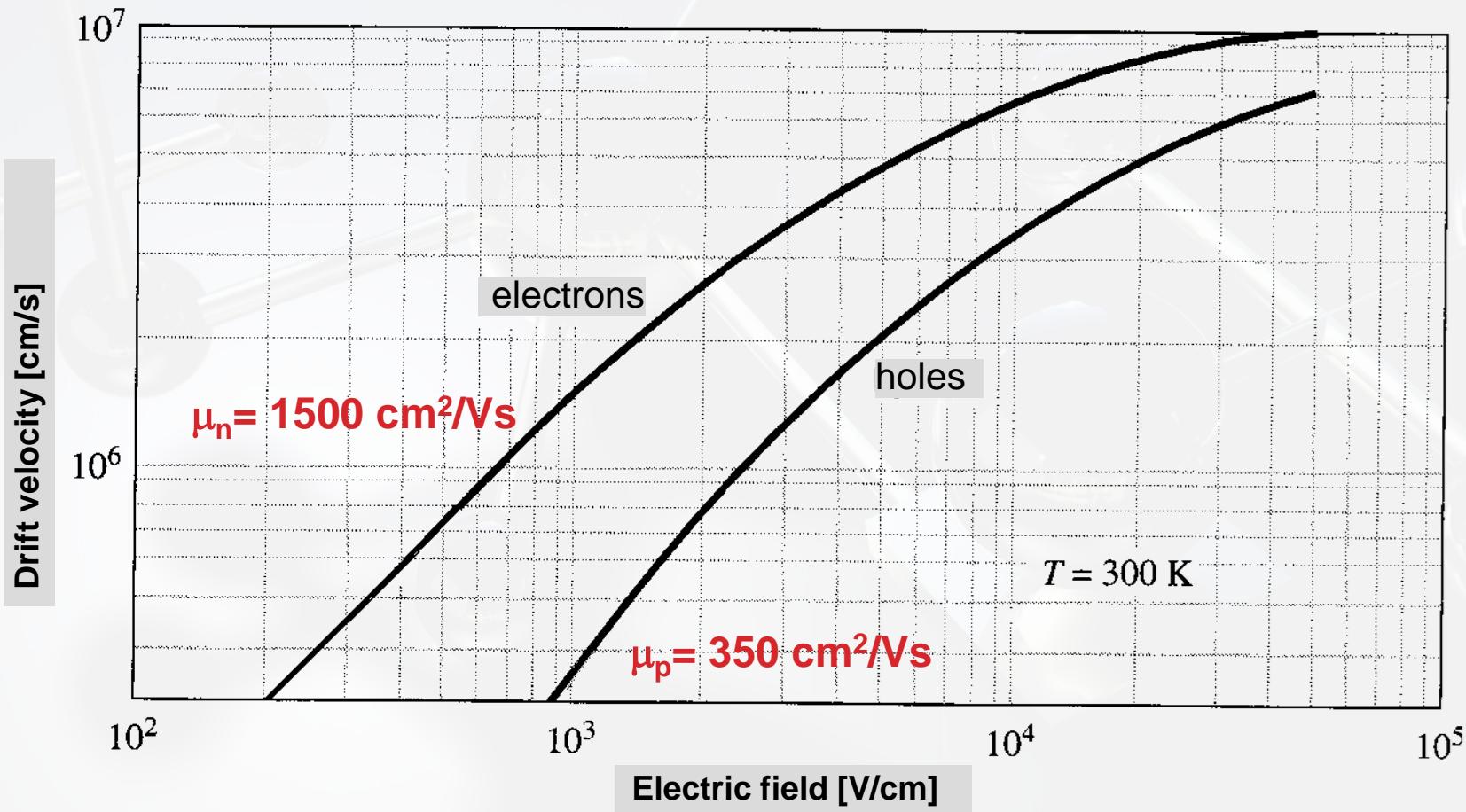
Specific resistance

$$\sigma_e = q(n \mu_n + p \mu_p)$$

Specific electrical conductivity of the
 semiconductor

Carrier mobilities

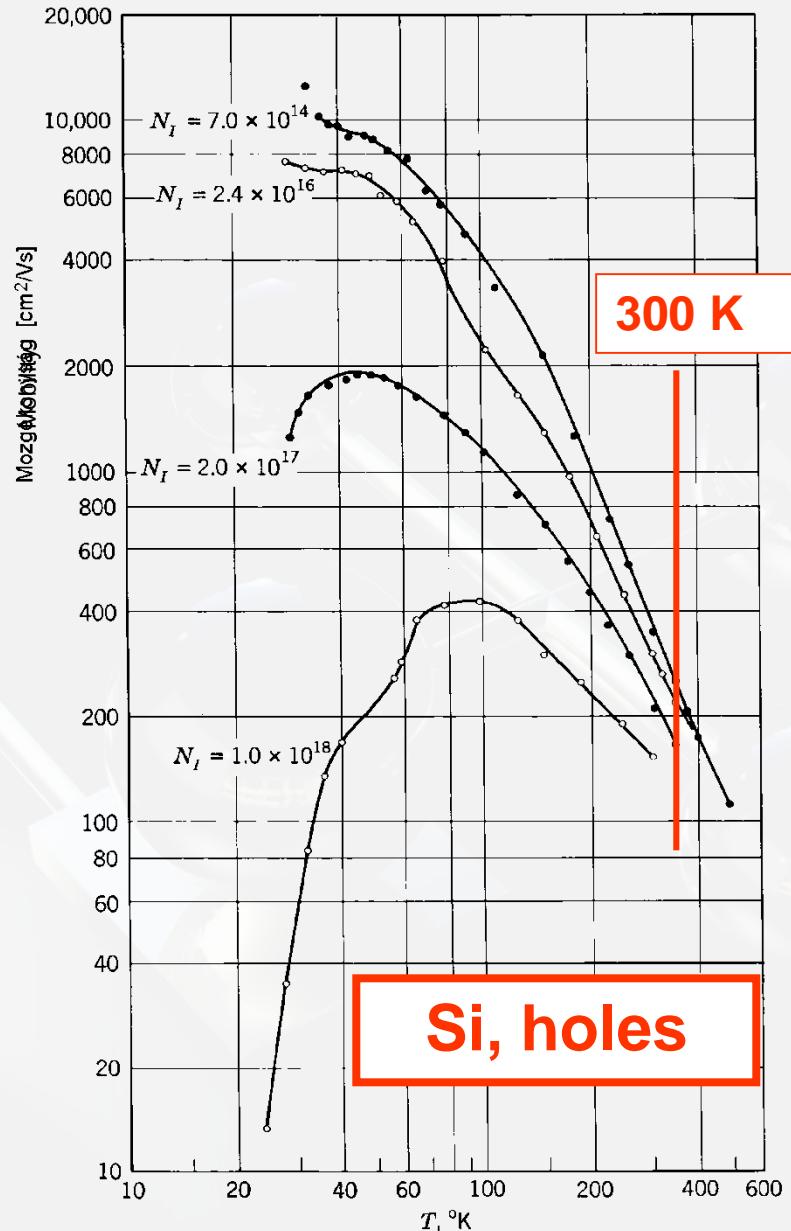
Si



Carrier mobilities

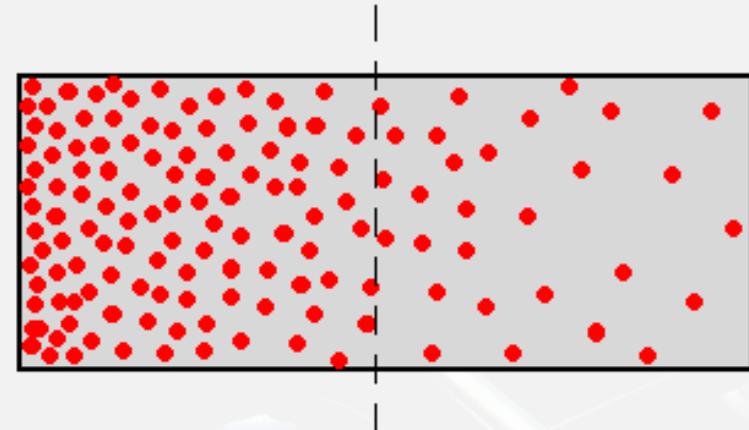
- Mobility decreases with increasing doping concentration
- At around room temperature mobilities decrease as temperature increases

$$\mu \sim T^{-3/2}$$



Diffusion current

- Reasons:
 - concentration difference (gradient)
 - thermal movement
- Proportional to the gradient
- D: diffusion constant [m^2/s]



$$\overline{J}_n = q D_n \overline{\text{grad}} n$$

$$\overline{J}_p = -q D_p \overline{\text{grad}} p$$

Total currents

$$\overline{\overline{J}_n} = q n \mu_n \overline{\overline{E}} + q D_n \overline{\overline{\text{grad}}} \ n$$

$$\overline{\overline{J}_p} = q p \mu_p \overline{\overline{E}} - q D_p \overline{\overline{\text{grad}}} \ p$$

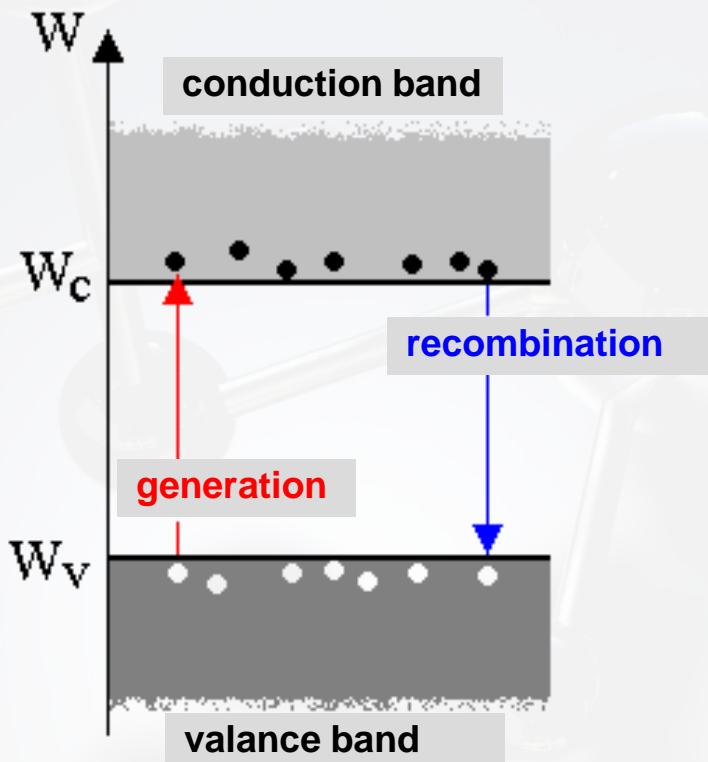
$$D = \frac{kT}{q} \mu$$

Einstein's relationship

$$U_T = \left. \frac{kT}{q} \right|_{T=300K} = \frac{1.38 \cdot 10^{-23} [\text{VAs/K}] \cdot 300[\text{K}]}{1.6 \cdot 10^{-19} [\text{As}]} \cong 0.026 \text{ V} = 26 \text{ mV}$$

Thermal voltage

Generation, recombination



- **Life-time:** average time an electron spends in the conduction band
 - This value is influenced by the impurities (recombination centers, allowed states in the band gap)
- τ_n, τ_p
- $1 \text{ ns} \dots 1 \mu\text{s}$
- If: τ_n the life-time of an e^- → probability of recombination within dt time is dt/τ_n
- **Recombination rate:** r [$1/\text{m}^3\text{s}$]

Number of recombined carriers in a unit volume within a unit timeframe.

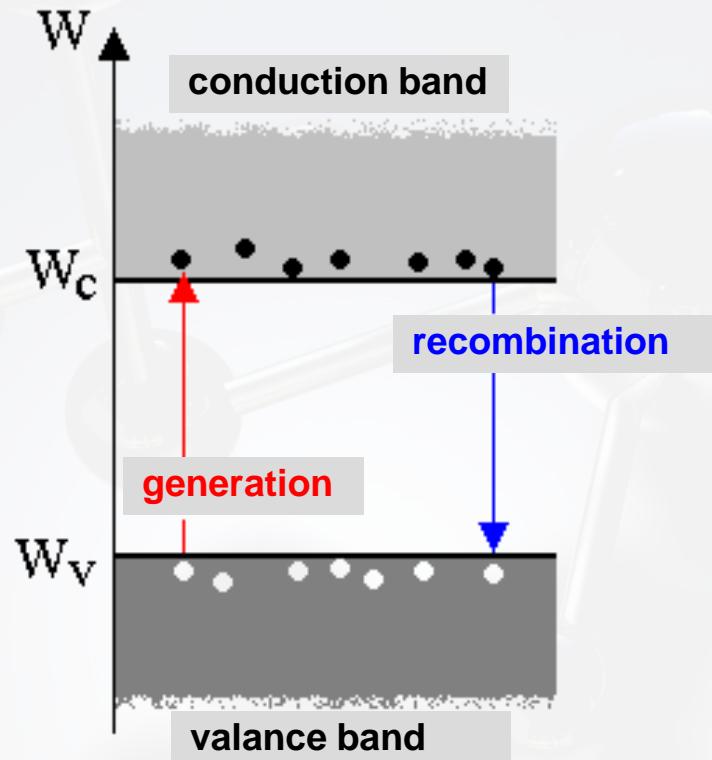
$$r_n = \frac{n \cdot dt}{\tau_n}$$

unit time,
unit V



$$r_n = \frac{n}{\tau_n}$$

Generation, recombination



- **Life-time:** average time an electron spends in the conduction band

$$\tau_n, \tau_p$$

1 ns ... 1 μ s

- **Generation rate:** g [1/m³s]

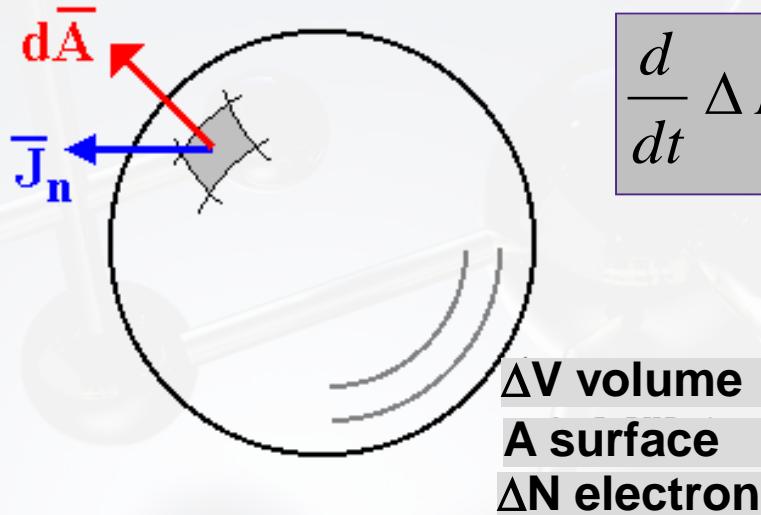
- **Recombination rate:** r [1/m³s]

$$r_n = \frac{n}{\tau_n}$$

$$r_p = \frac{p}{\tau_p}$$

$$g_n = r_n|_{equilibrium} = \frac{n_0}{\tau_n}$$

Continuity equation



$$\frac{d}{dt} \Delta N = -\frac{1}{-q} \oint_A \bar{J}_n d\bar{A} + g_n \cdot \Delta V - \frac{n}{\tau_n} \Delta V$$

$$\frac{d}{dt} \frac{\Delta N}{\Delta V} = \frac{1}{q} \frac{1}{\Delta V} \oint_A \bar{J}_n d\bar{A} + g_n - \frac{n}{\tau_n}$$



$$\frac{dn}{dt} = \frac{1}{q} \operatorname{div}(\bar{J}_n) + g_n - \frac{n}{\tau_n}$$

Diffusion equation

$$\frac{dn}{dt} = \frac{1}{q} \operatorname{div}(\bar{J}_n) + g_n - \frac{n}{\tau_n}$$

$$\bar{J}_n = q n \mu_n \bar{E} + q D_n \operatorname{grad} n$$

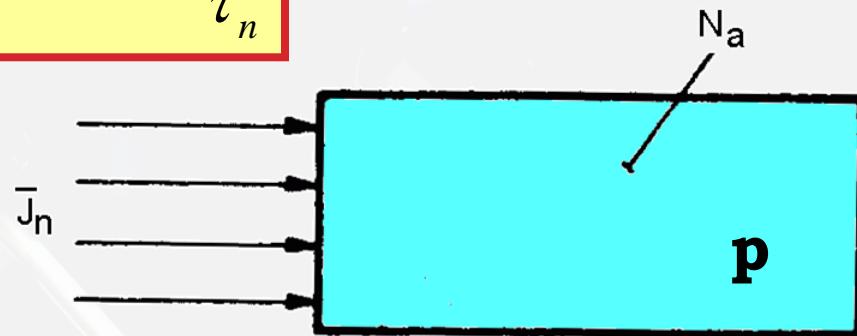
$$\frac{dn}{dt} = \mu_n \operatorname{div}(n \bar{E}) + D_n \operatorname{divgrad} n + g_n - \frac{n}{\tau_n}$$

$$\frac{dp}{dt} = -\mu_p \operatorname{div}(p \bar{E}) + D_p \operatorname{divgrad} p + g_p - \frac{p}{\tau_p}$$

Example for the solution of the diff. eq.:

$$\frac{dn}{dt} = \mu_n \operatorname{div}(n \bar{E}) + D_n \operatorname{divgrad} n + g_n - \frac{n}{\tau_n}$$

- ▶ Si block doped homogenously by p
- ▶ e^- injected with constant current density
- ▶ e^- are moving according to the diffusion ($E=0$) while recombining
- ▶ Steady-state ($dn/dt=0$)



- ▶ What is the $n(x)$ distribution of the injected minority carriers?
- ▶ What is the average penetration depth, prior to recombination?

Example for the solution of the diff. eq.:

$$\frac{dn}{dt} = \mu_n \operatorname{div}(n \bar{E}) + D_n \operatorname{divgrad} n + g_n - \frac{n}{\tau_n}$$

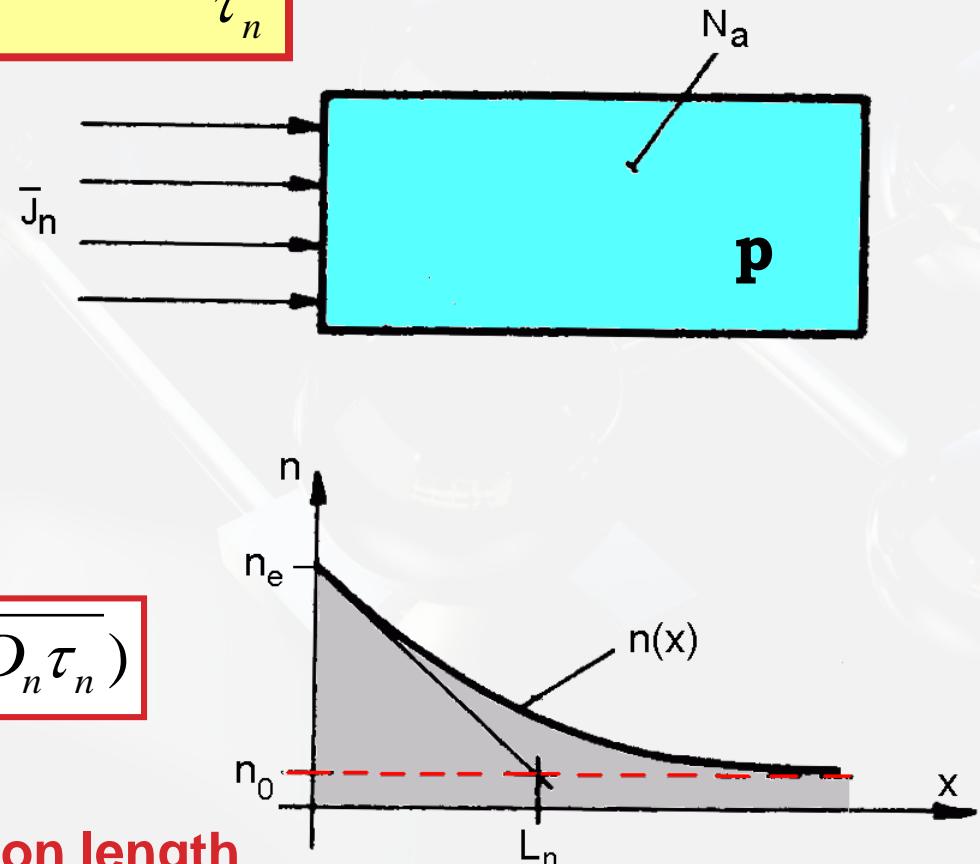
$$0 = D_n \frac{d^2 n}{dx^2} + g_n - \frac{n}{\tau_n}$$

$$0 = D_n \frac{d^2 n}{dx^2} + \frac{n_0}{\tau_n} - \frac{n}{\tau_n}$$

$$n(x) = n_0 + (n_e - n_0) \exp(-x / \sqrt{D_n \tau_n})$$

$$L_n = \sqrt{D_n \tau_n}$$

diffusion length



Omne ignotum pro magnifico.

Everything unknown seems magnificent.
Tacitus, 55-120 AD, Roman historian



Budapest University of Technology and Economics
Department of Electron Devices

Microelectronics, BSc course

**Operation of PN junctions:
Electrostatic conditions**

Diodes: basics

- What are they? Data sheets
- How are they made?
- How do they work?

Diodes: what are they? We learnt:

- ...as diodes are presented in vendors' data sheets:

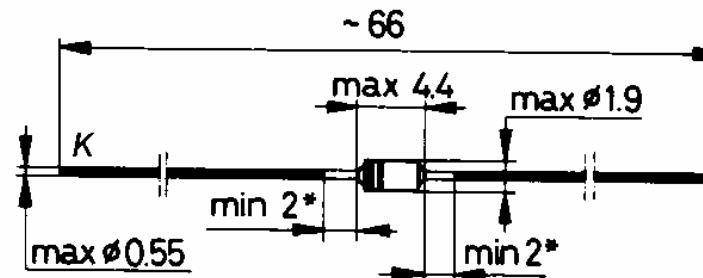
1N 4151 (BAY 95), 1N 4154 (BAY 94)

Silicon Epitaxial Planar Low-Capacitance Diodes

for very fast switching applications.

Dimensions in mm

Band: cathode



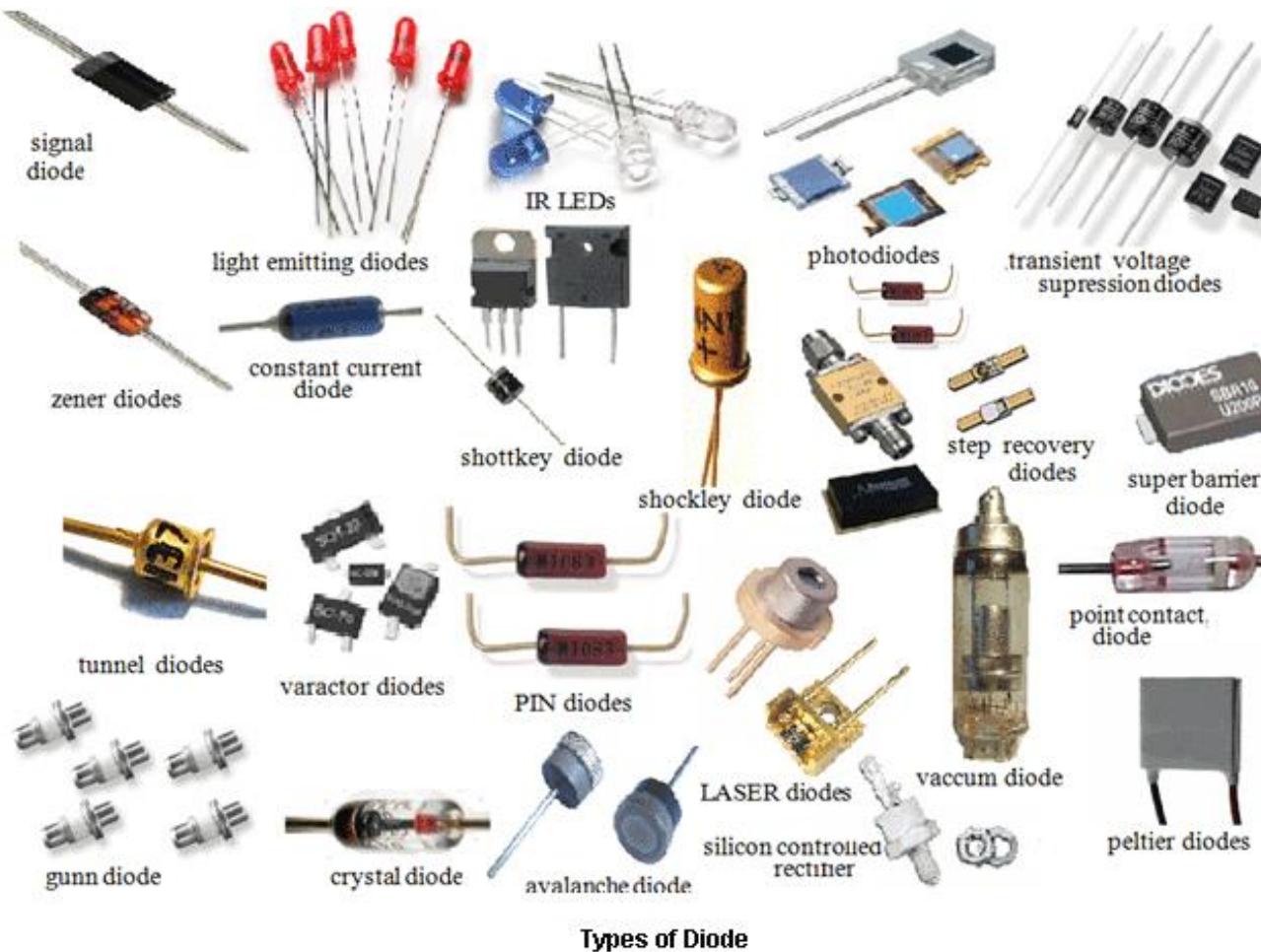
Case: DO-35

* not tinned

Mass: approx. 0.15 g

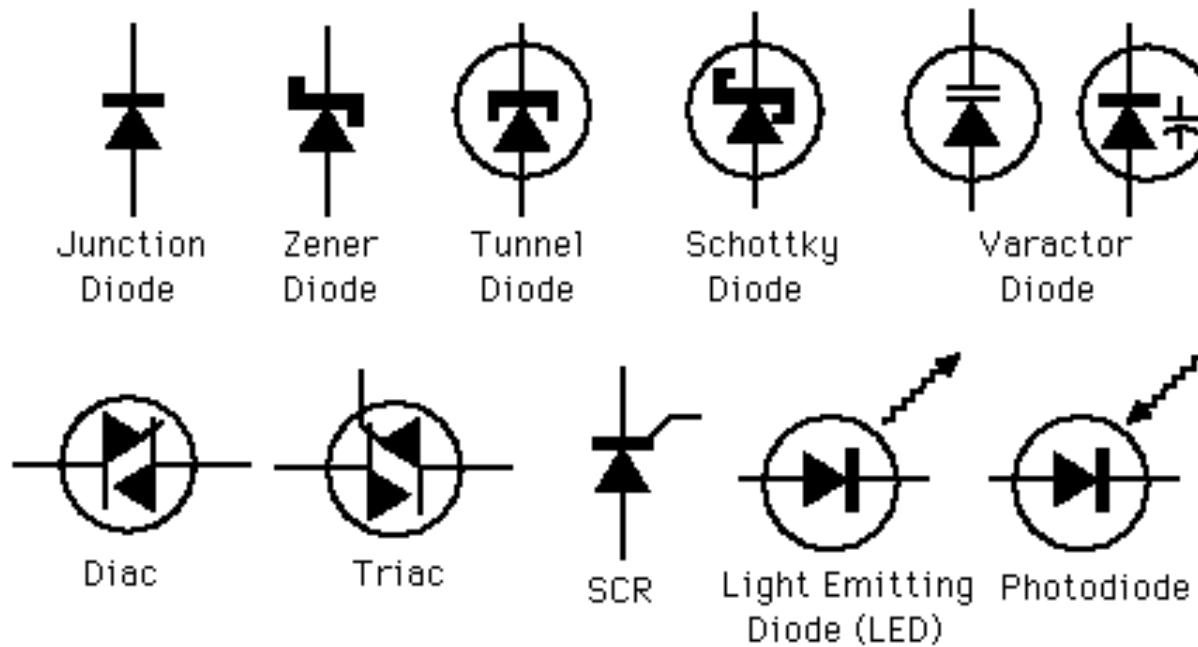
Diodes: what are they? We learnt:

- ...as they are actually look like:

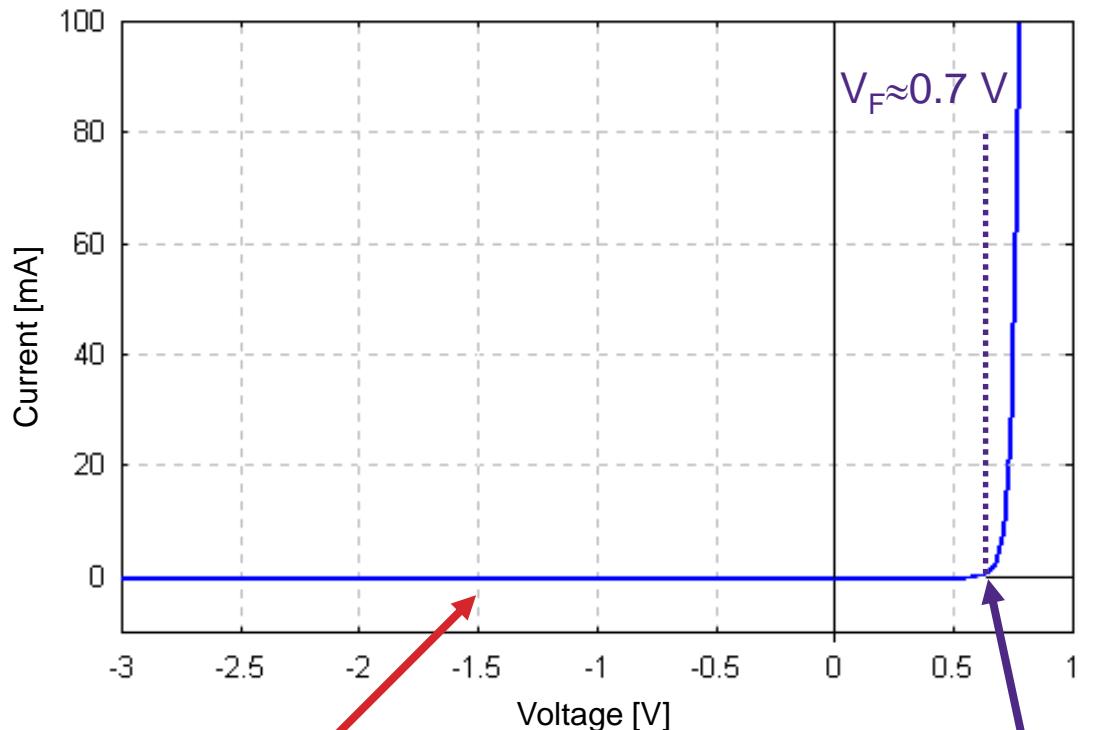


Diodes: what are they? We learnt:

- ...and their symbols:



Main features



Reverse region
 $I \sim 10^{-12} \text{ A/mm}^2$
(Si, T=300 K)

Forward region
 $I \sim \exp(V/V_T)$

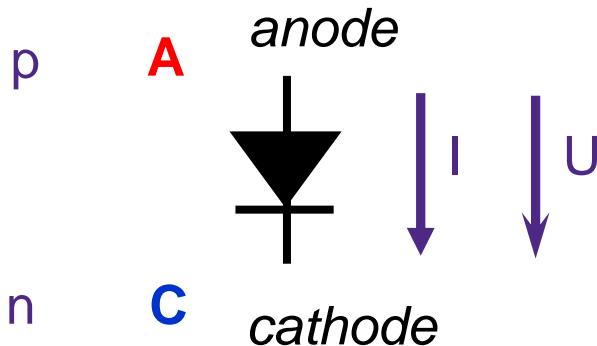
Rectifies

The
characteristic:

$$I = f(V)$$

Main features

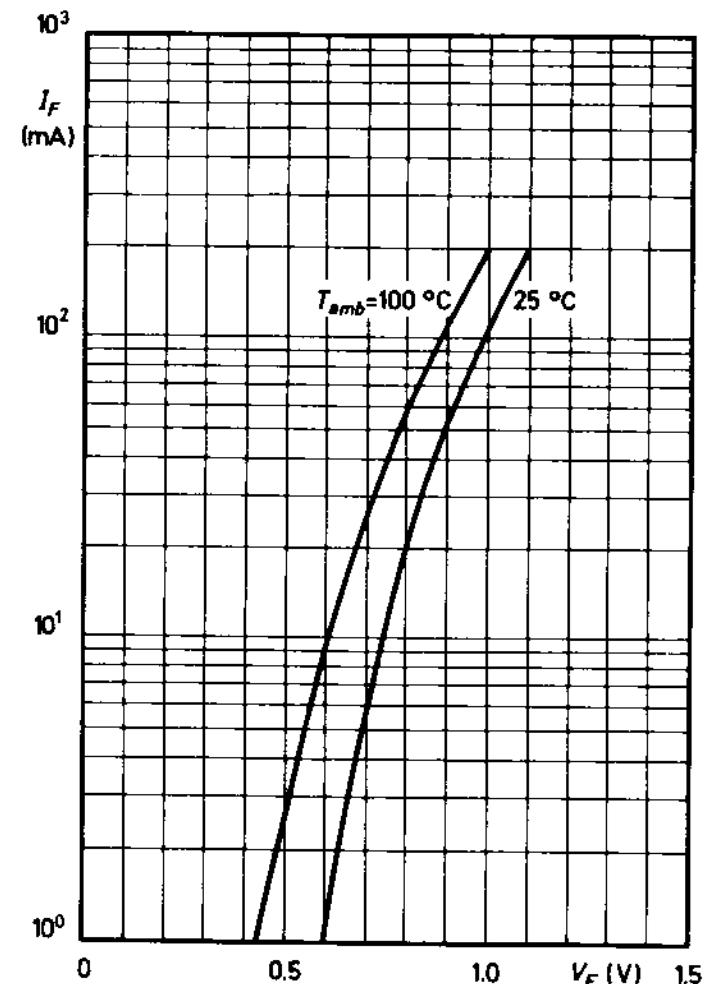
Symbol, reference directions



U_F or V_F forward voltage

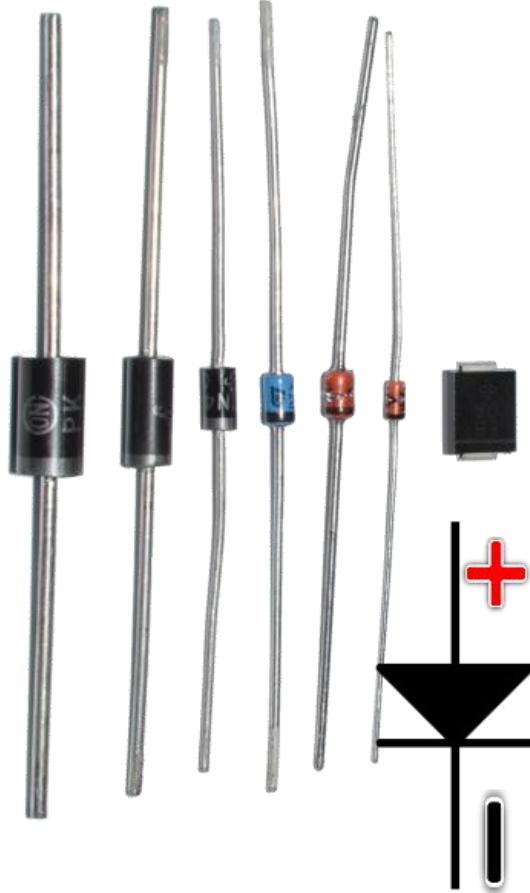
I_F forward current

Forward characteristics $I_F = f(V_F)$

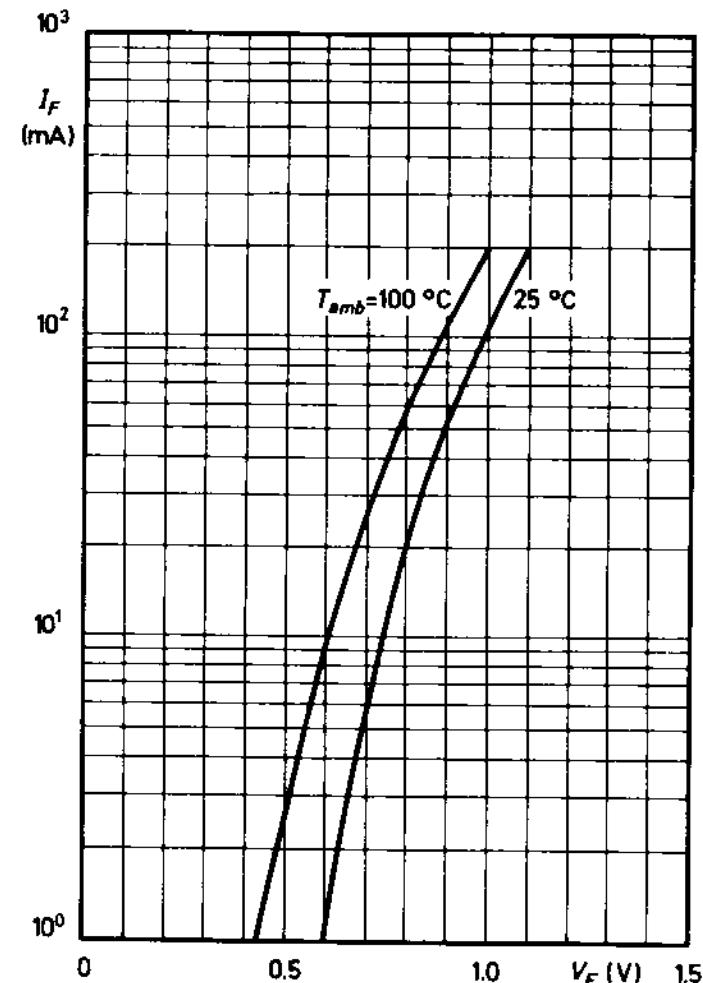


Main features

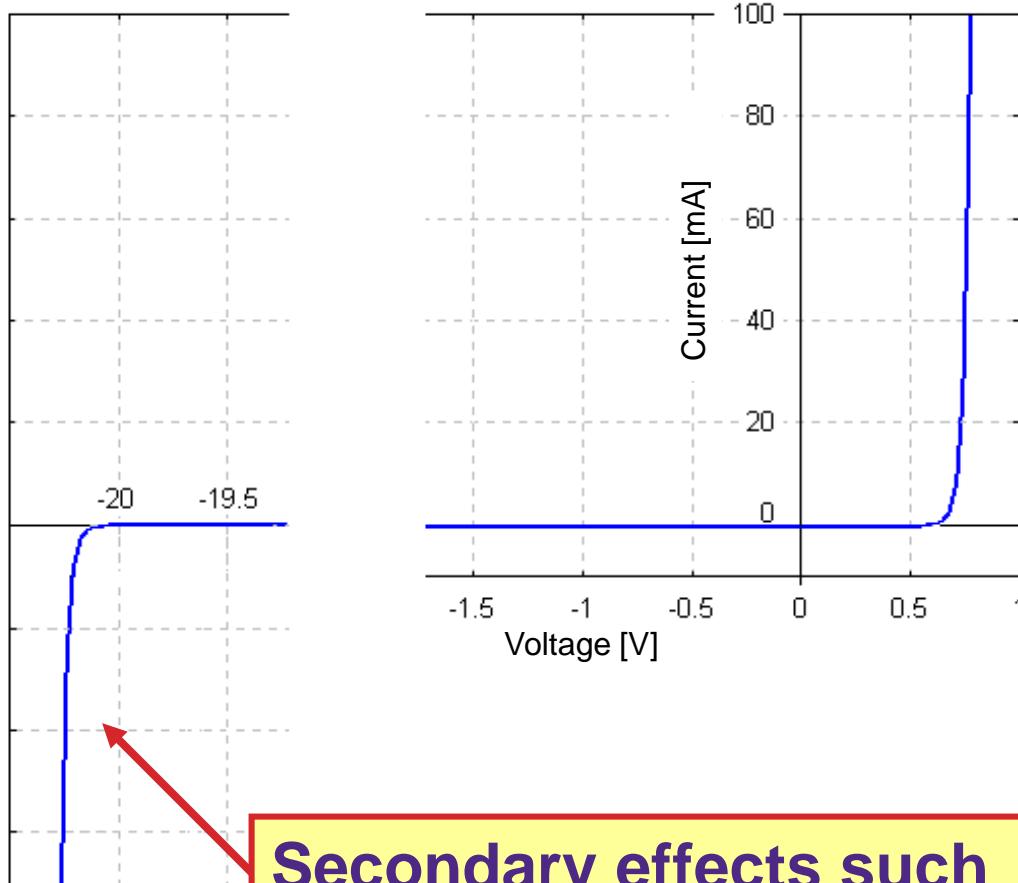
Symbol, reference directions



Forward characteristics $I_F = f(V_F)$



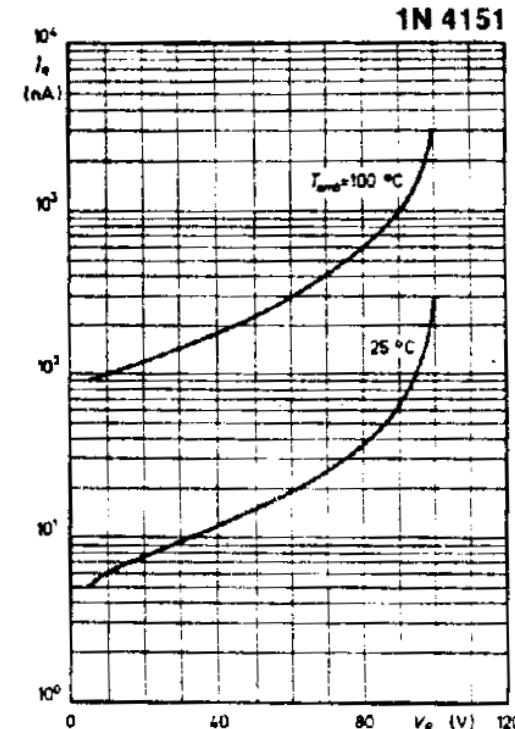
Main features



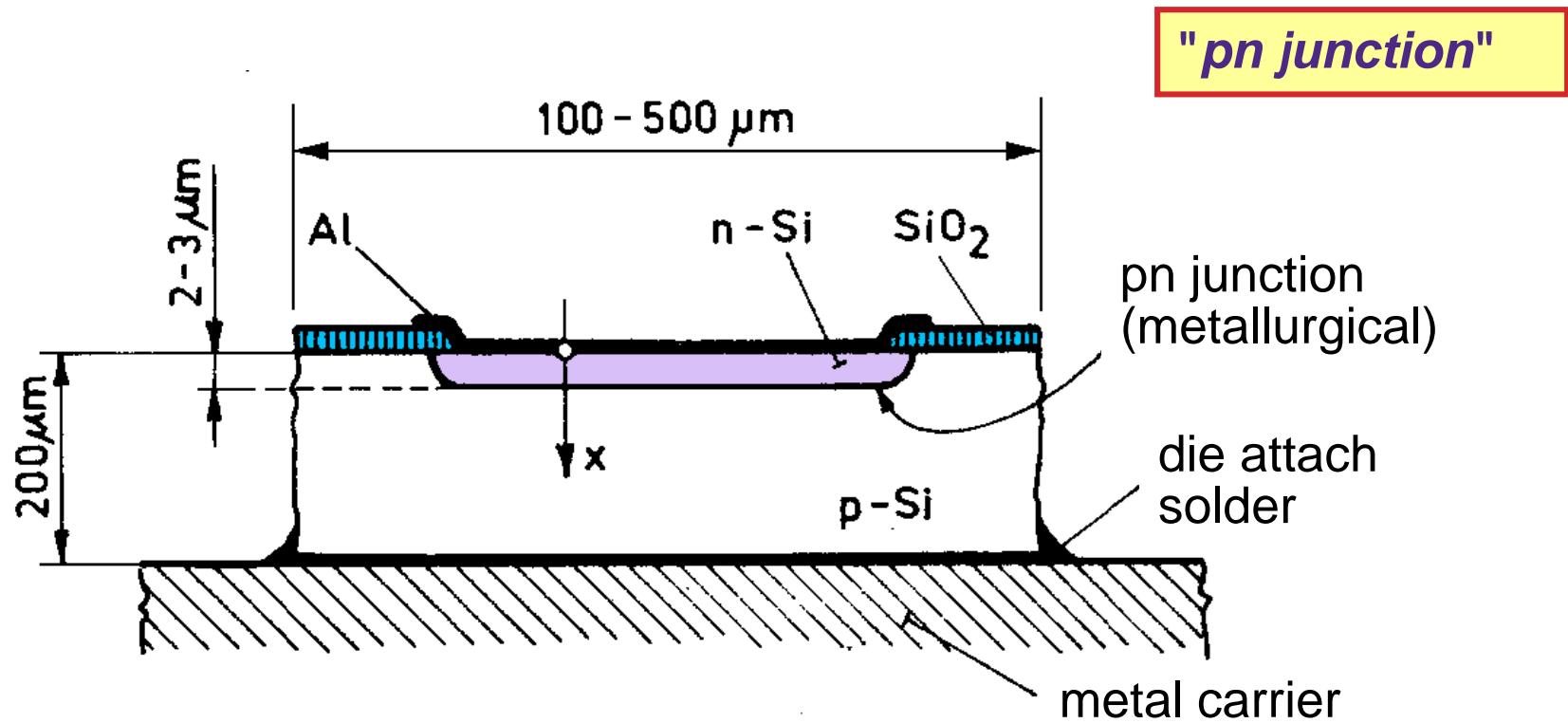
Secondary effects such
as breakdown

Dynamic properties:
capacitance, finite
speed of operation

Reverse characteristics



How does it look like?

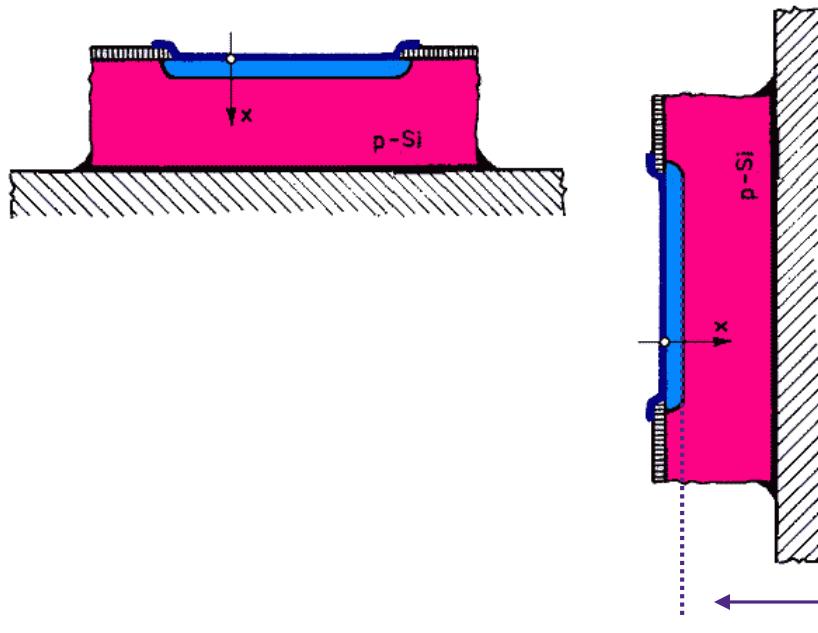


Start from: single crystalline Si wafer

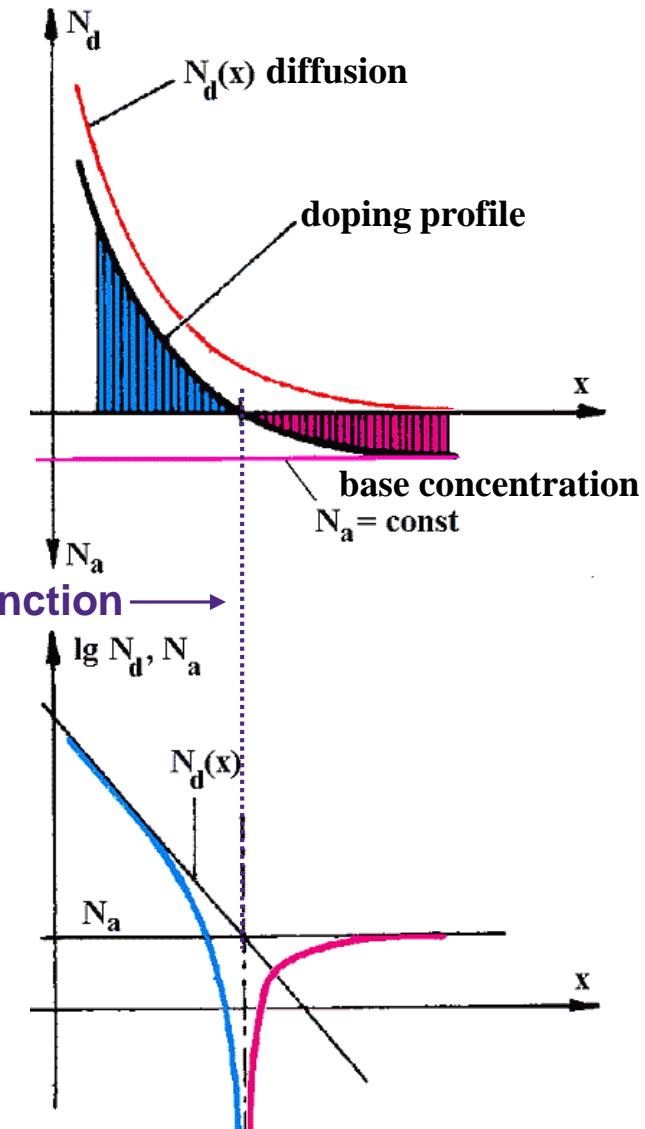
Oxidation, window opening, n diffusion, metallization

Dicing, die attach soldering, packaging

Diode – doping profile



Doping profile: dopant concentration as function of depth

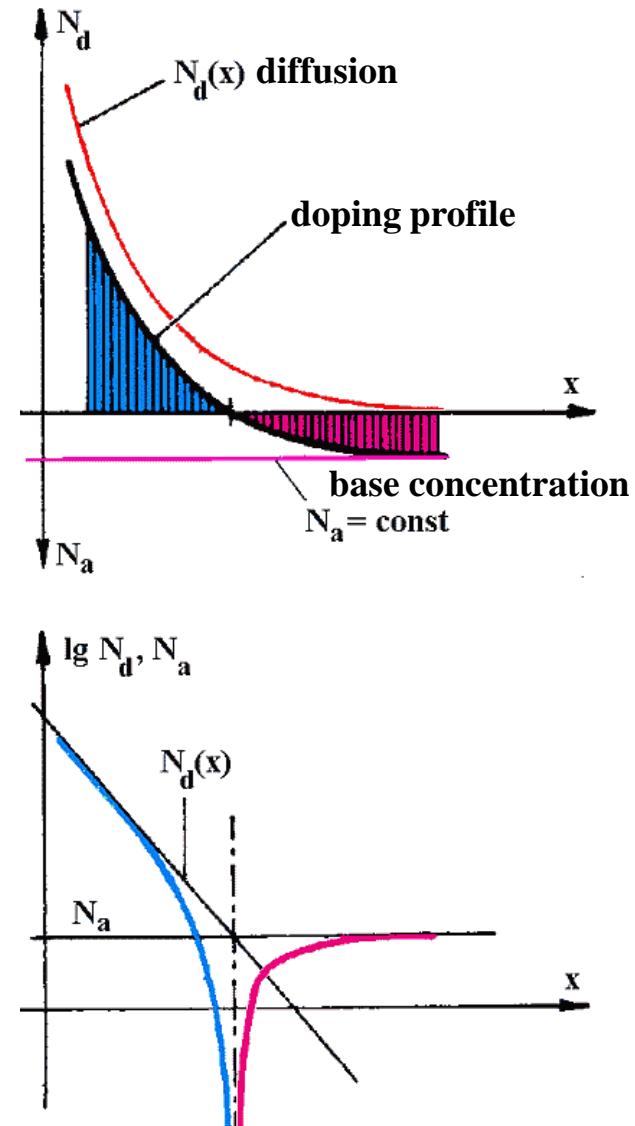


Diode – doping profile

Doping profile: doping density as a function of depth

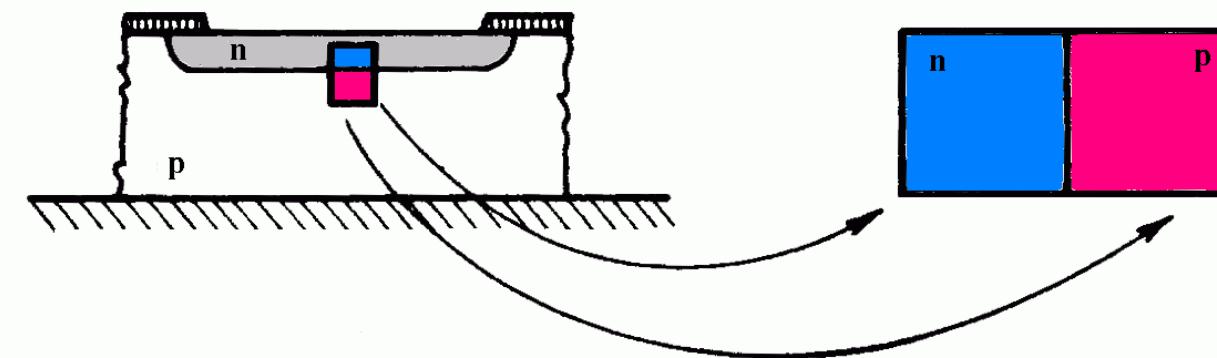
The doping profile depends on the manufacturing technology and of the application of the diode!

- diffusion (exponential profile)
- epitaxial layer growth (sudden/abrupt transition, within 0.1um distance p doping is changed to n, homogeneous doping)
- ion-implantation (sudden transition, possibility of homogeneous doping)



Our method of study

1. 1D analysis, internal PN-junction only

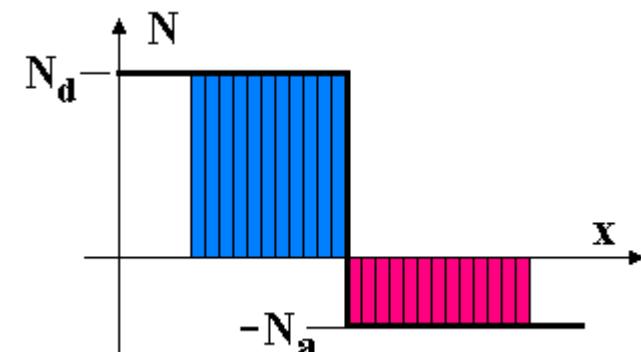


2. Homogeneous doping

“abrupt” profile

3. One side is more heavily doped than the other side

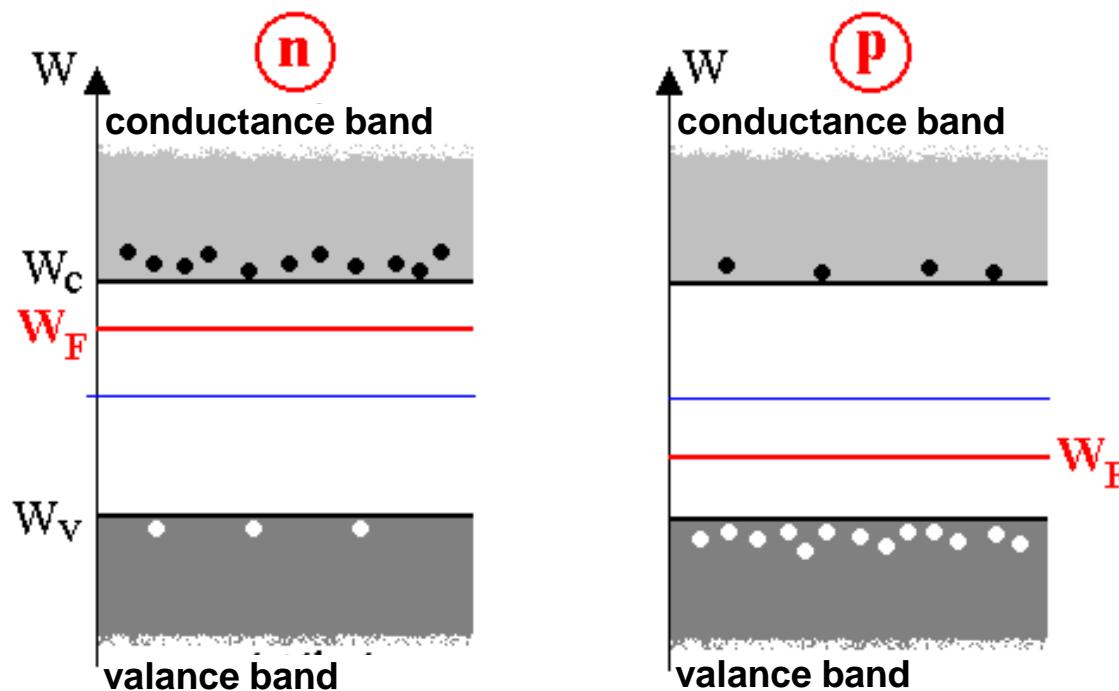
(Let it be the n-side)



$$N_d \gg N_a$$

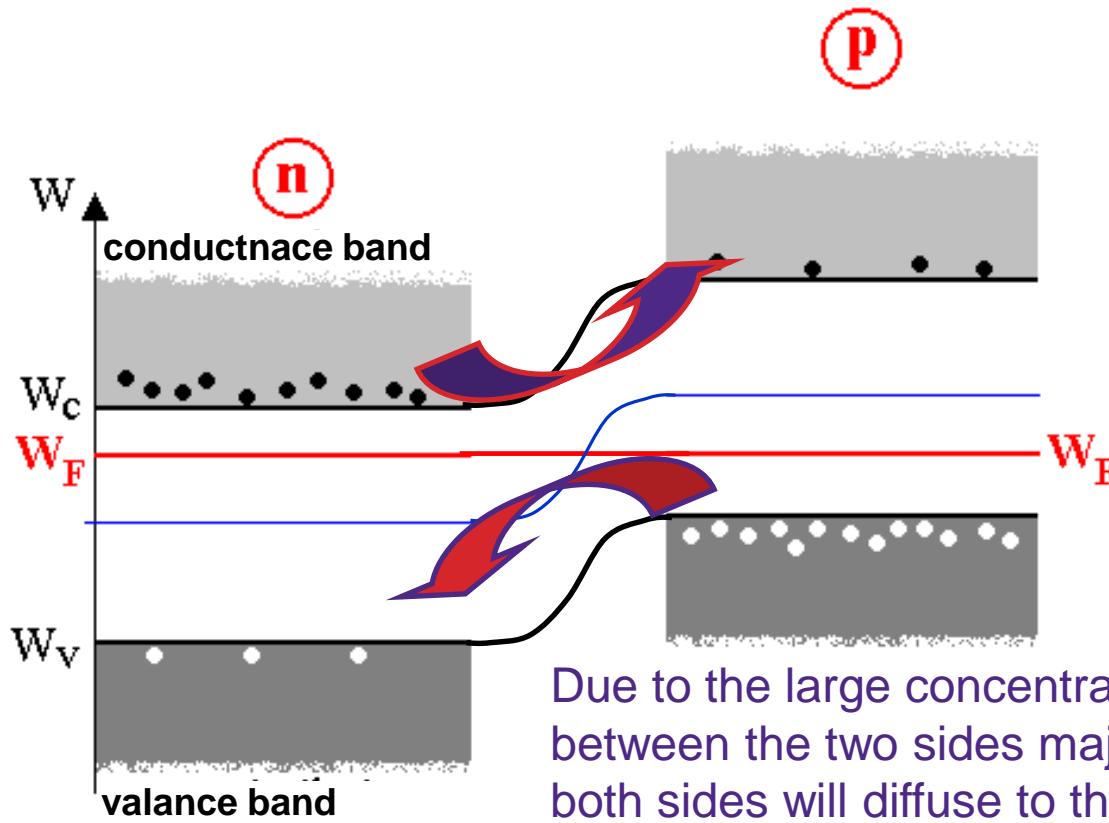
Two separate pieces of doped Si

- The Fermi-levels shift from the intrinsic level according to the doping:



PN junction

- A potential step develops between the p and n sides. This will be so high that the Fermi-levels of both sides will be equal:

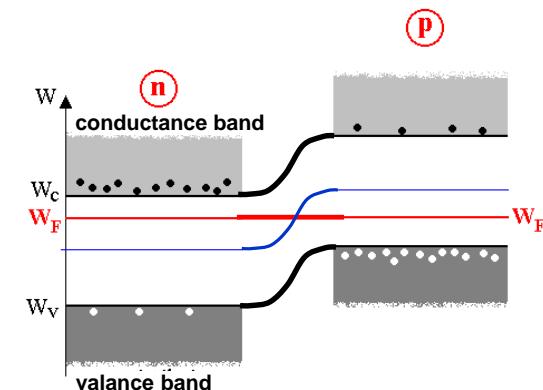


Due to the large concentration gradient between the two sides majority carriers of both sides will diffuse to the other side until the Fermi-levels get equal.

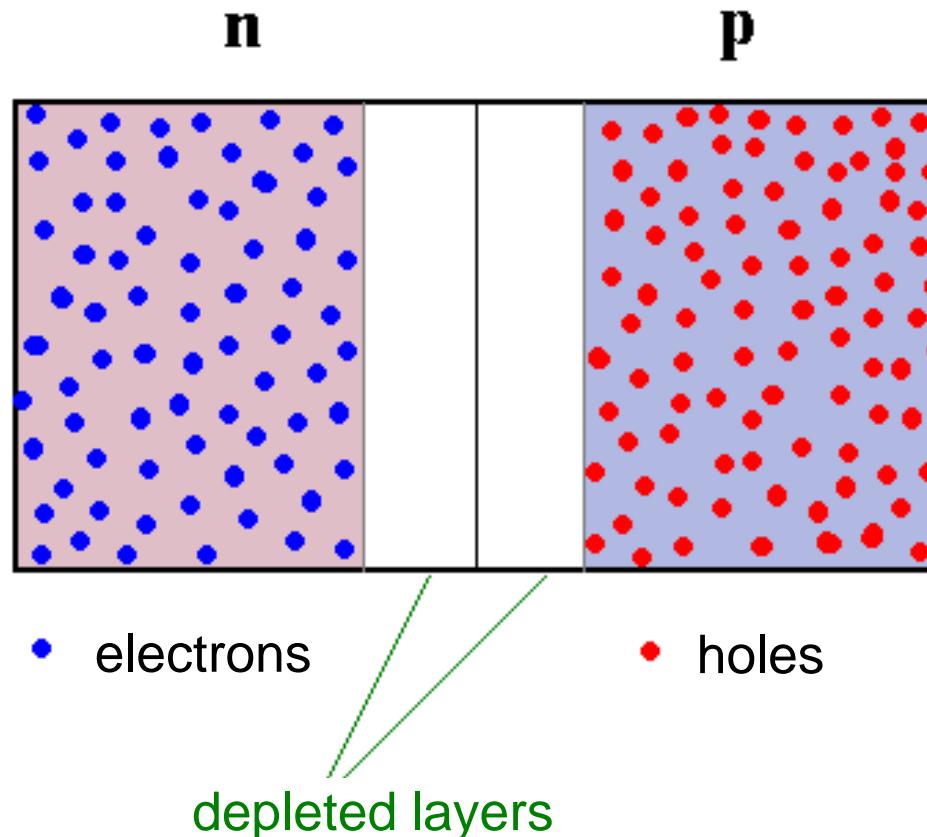
PN junction

- Significant difference of carriers between the two sides
- In the n side e^- -s, in the p side holes are the majority
- The density gradient (grad n) causes diffusion current
- Holes flow from the p side, e^- flow from the n side (same direction!)
- Opposite effect is needed for the balance!
- \bar{E} is needed for the drift current!
- A potential step must be developed (*contact potential, contact pot. difference*)

$$J_n(x) = -qD_n \frac{dn}{dx}$$



Electrostatic conditions



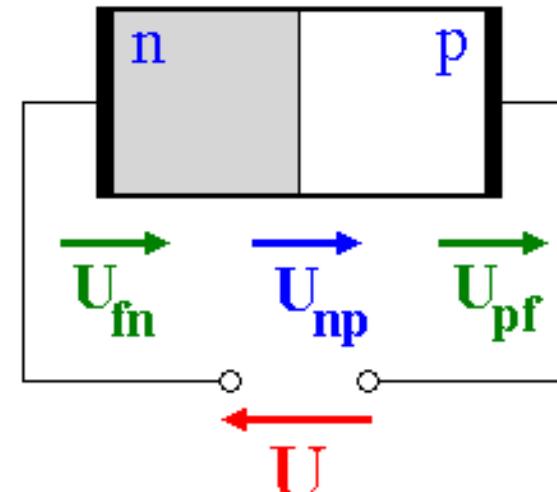
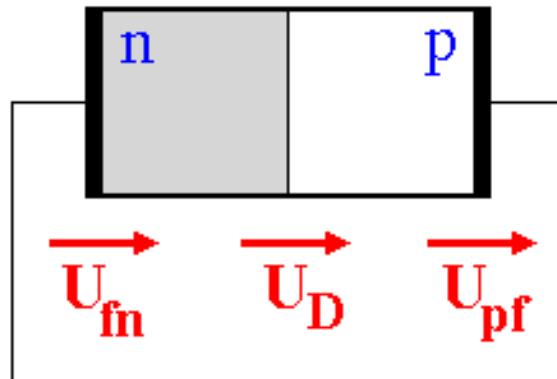
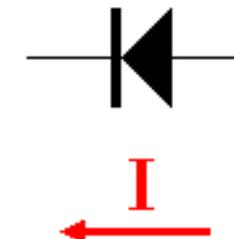
Depleted layers (space charge layers)

Contact & diffusion potentials

U_{fn} metal – n-Si contact potential

U_D diffusion potential between p & n sides

U_{pf} p-Si – metal contact potential



According to Kirchoff's voltage law:

$$U_D + U_{fn} + U_{pf} = 0$$

$$U_{np} + U_{fn} + U_{pf} = -U$$

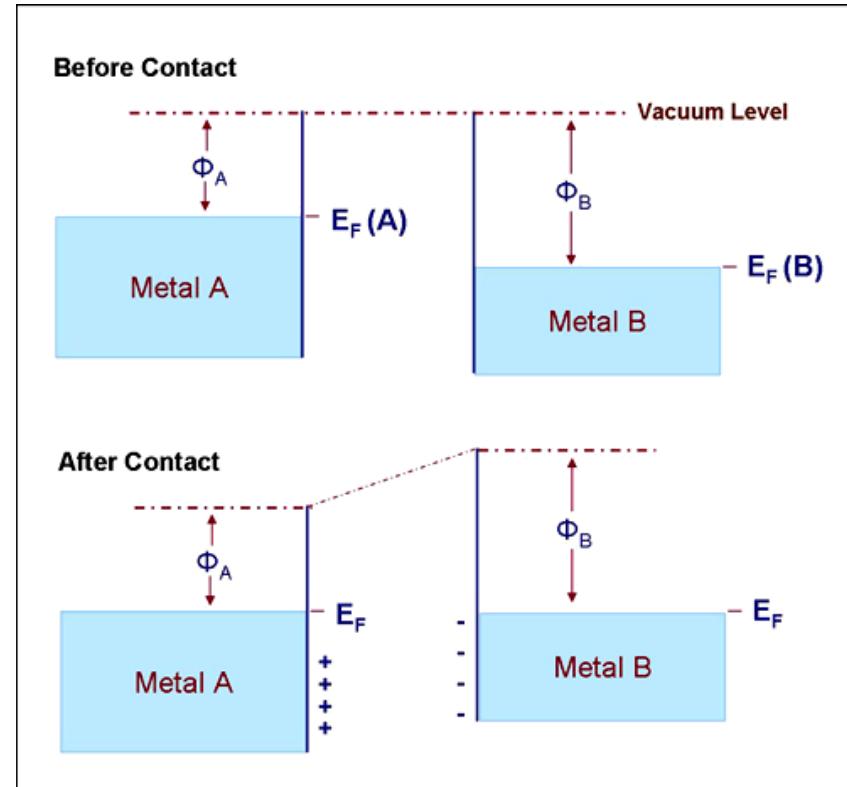
$$U_{np} = U_D - U$$

Contact potential metal-metal

- Temperature sensor
Peltier, Seebeck eff.
- Before contacting the metals have different work functions

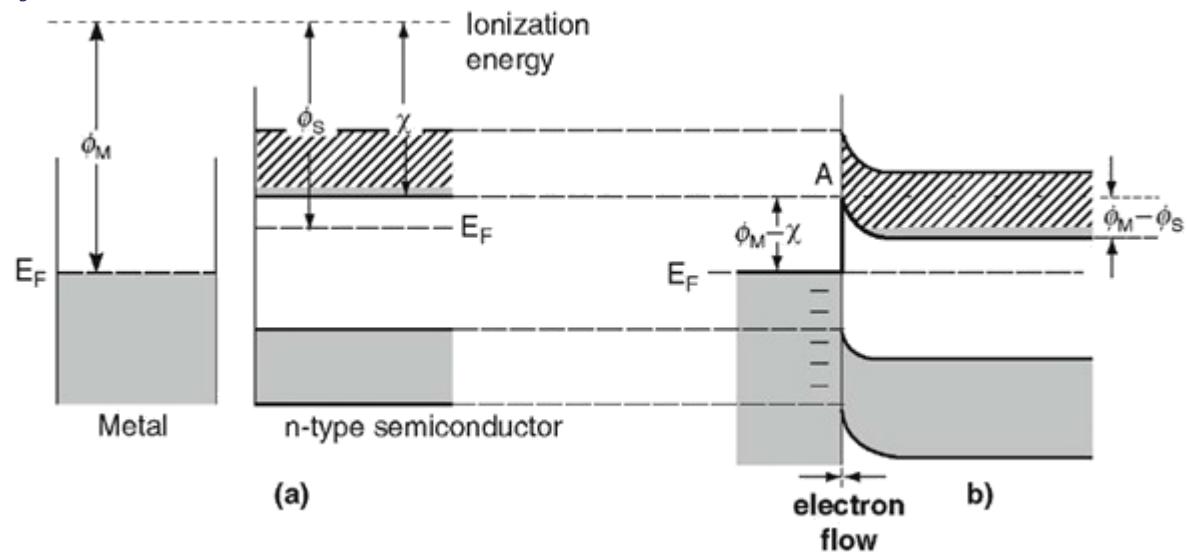
$$\text{Einstein formula: } E = h \cdot v = \phi_A + \frac{1}{2}mv^2$$

- Different Fermi-levels
- Contact $\rightarrow e^-$ current
- Metals with smaller ϕ_A : e^- deficit, apparent positive surplus, E_f decrease
- Potential of the other metal becomes more negative, E_f increases



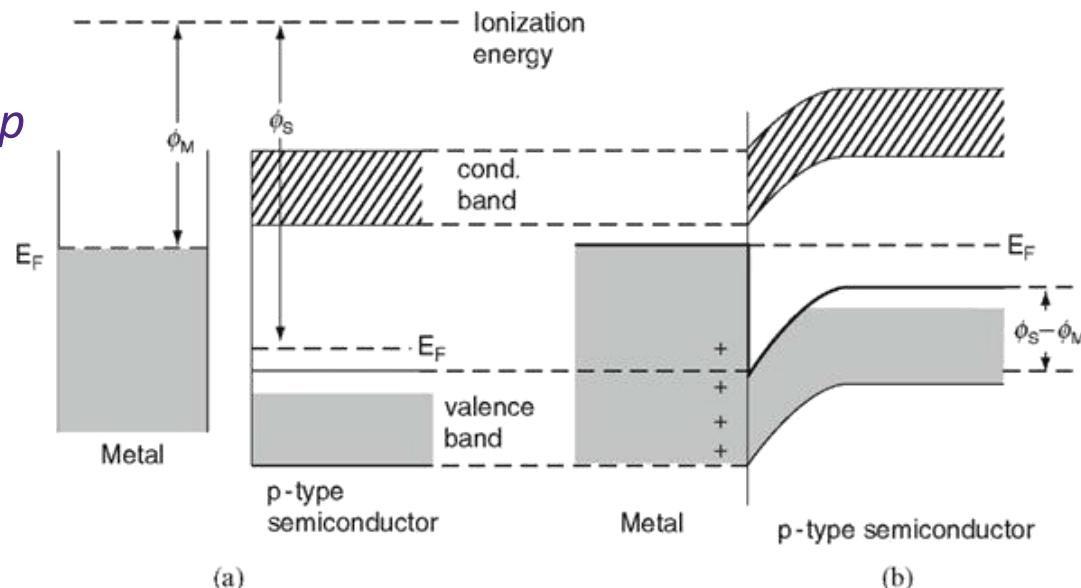
Contact potential metal-semicond.

- ▶ n type semiconductor, e^- current into the metal
- ▶ Metal: e^- surplus, apparent negative potential
- ▶ Semiconductor: e^- shortage, band bending!
- ▶ Potential barrier: $\Phi_b = \Phi_M - X$
- ▶ X electron affinity

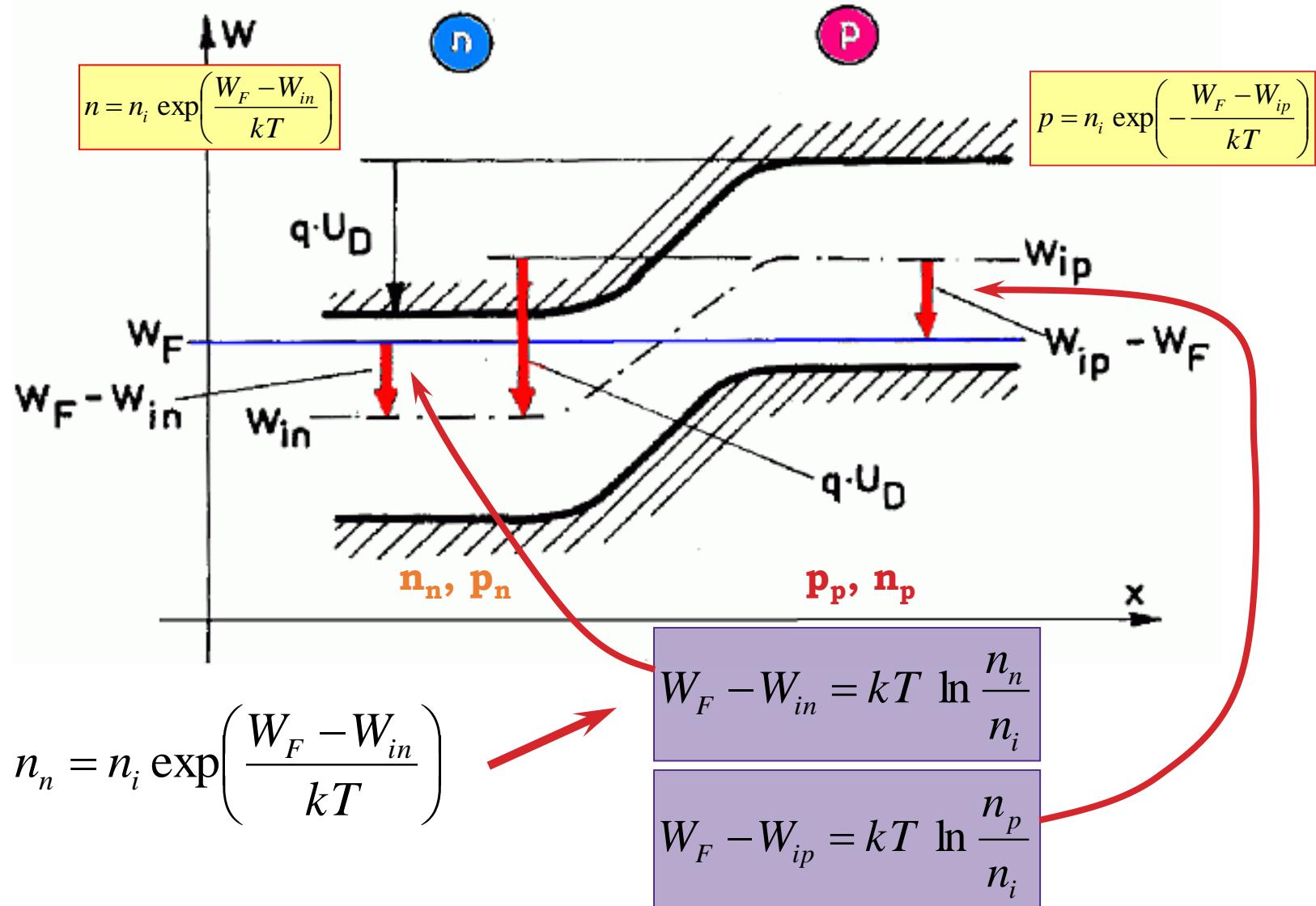


Contact potential metal-semicond.

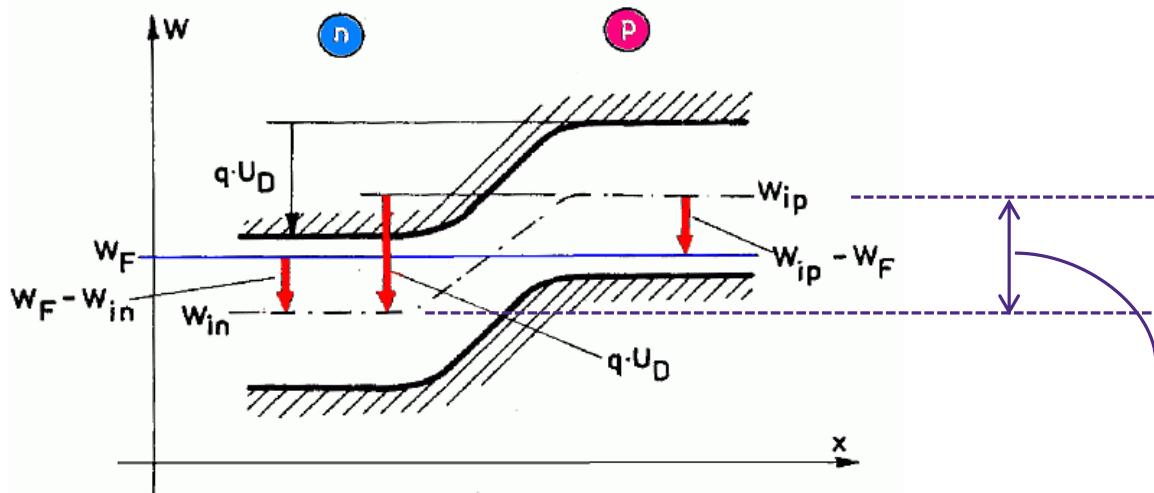
- ▶ p type semiconductor, e^- current into the semicond.
- ▶ Metal: e^- shortage, apparent positive potential
- ▶ Semiconductor: e^- surplus, band bending!
- ▶ Potential barrier:: $\Phi_b = E_g - (\Phi_M - X)$
- ▶ Influenced by:
 - Surface states
Energy levels in the band gap
 - Thicknes of the surface layers



Calculation of the diffusion potential



Calculation of the diffusion potential



$$W_F - W_{in} = kT \ln \frac{n_n}{n_i}$$

$$W_F - W_{ip} = kT \ln \frac{n_p}{n_i}$$

$$W_{ip} - W_{in} = kT \ln \frac{n_n}{n_p}$$

$$U_D = \frac{W_{in} - W_{ip}}{-q} = \frac{kT}{q} \ln \frac{n_n}{n_p} = \frac{kT}{q} \ln \frac{n_n p_p}{n_i^2}$$

$$U_D = U_T \ln \frac{N_d N_a}{n_i^2}$$

„built-in“ voltage

$$n_p = n_i^2 / p_p$$

mass effect law

Calculation of the diffusion potential

$$U_D = U_T \ln \frac{N_d N_a}{n_i^2}$$

Problem

Doping levels of an abrupt Si diode:
 $N_d = 10^{18}/\text{cm}^3$, $N_a = 10^{16}/\text{cm}^3$.

Let us calculate the diffusion potential at room temperature!

$$U_D = 0.026 \cdot \ln \frac{10^{18} \cdot 10^{16}}{10^{20}} = 0.026 \cdot \ln 10^{14} = 0.838 \text{ V}$$

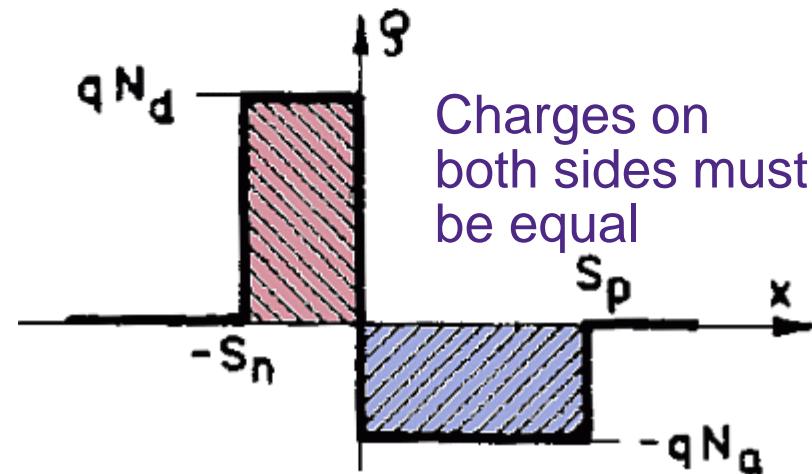
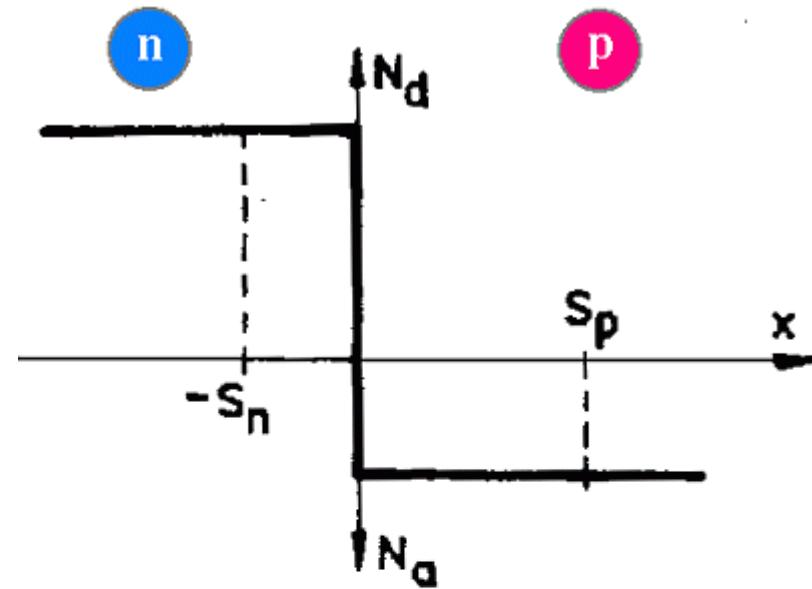
Obviously $U_D < U_g$, U_D is usually 70-80% of U_g

Calculations for the depletion layer

$$q S_n N_d = q S_p N_a$$

$$\frac{N_a}{N_d} = \frac{S_n}{S_p}$$

The depletion layer is wider on the less doped side.



Calculations for the depletion layer

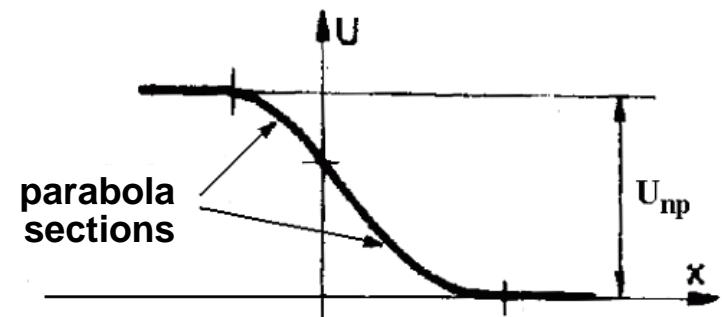
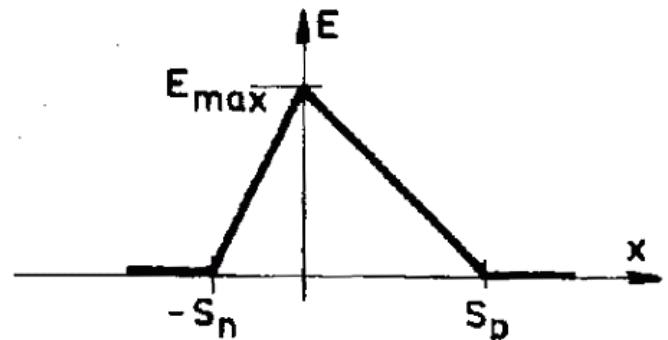
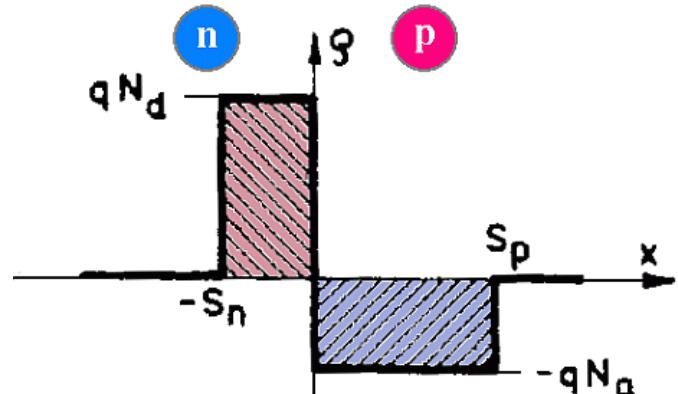
Poisson eq.:

$$\frac{dE}{dx} = \frac{\rho(x)}{\epsilon}$$

$$E(x) = \frac{1}{\epsilon} \int_{-\infty}^x \rho(\xi) d\xi$$

$$E_{\max} = \frac{1}{\epsilon} \int_{-S_n}^0 q \cdot N_d dx$$

$$E_{\max} = \frac{q N_d S_n}{\epsilon} = \frac{q N_a S_p}{\epsilon}$$

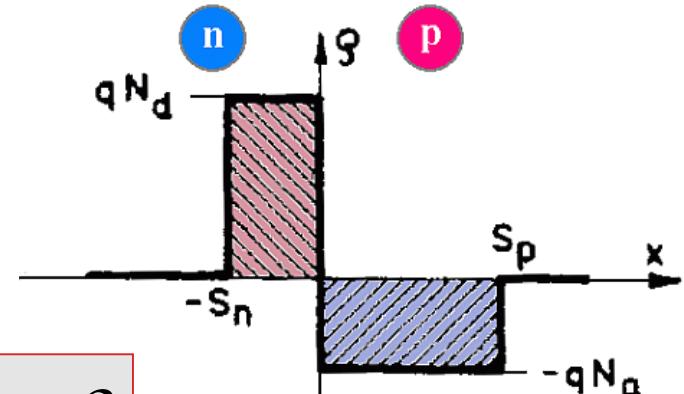


Calculations for the depletion layer

$$E_{\max} = \frac{q N_a}{\epsilon} S_p$$

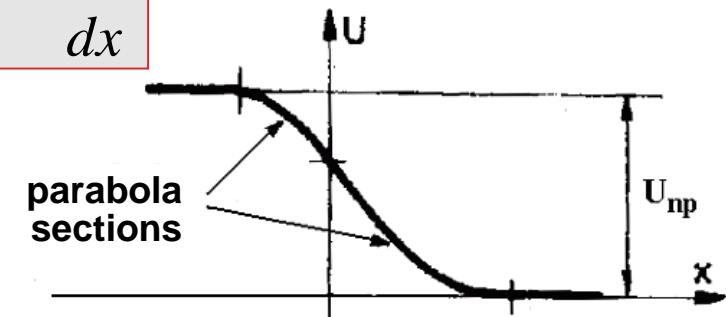
$$U_{np} = \frac{1}{2} E_{\max} (S_n + S_p) \approx \frac{1}{2} E_{\max} S_p$$

$$U_{np} = \frac{1}{2} \frac{q N_a}{\epsilon} S_p^2$$



$$\frac{dE}{dx} = \frac{\rho}{\epsilon}$$

$$E = -\frac{dU}{dx}$$



Calculations for the depletion layer

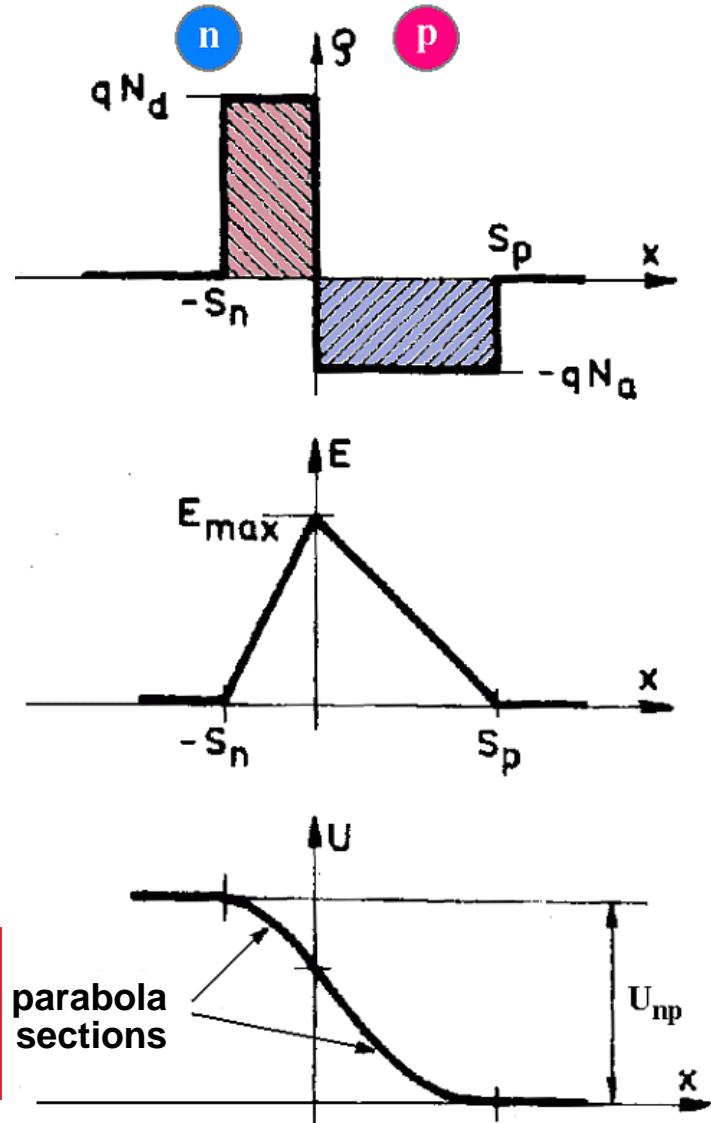
$$U_{np} = \frac{1}{2} \frac{qN_a}{\epsilon} S_p^2$$

$$S_p = \sqrt{\frac{2\epsilon}{qN_a}} \sqrt{U_{np}} = \sqrt{\frac{2\epsilon}{qN_a}} \sqrt{U_D - U}$$

$$S_n = \frac{N_a}{N_d} S_p$$

$$E_{max} = \frac{q N_a}{\epsilon} S_p$$

$$E_{max} = \frac{q N_a}{\epsilon} \sqrt{\frac{2\epsilon}{qN_a}} \sqrt{U_D - U} = \sqrt{\frac{2q N_a}{\epsilon}} \sqrt{U_D - U}$$



Calculations for the depletion layer

$$S_p = \sqrt{\frac{2\epsilon}{qN_a}} \sqrt{U_{np}} = \sqrt{\frac{2\epsilon}{qN_a}} \sqrt{U_D - U}$$

$$S_n = \frac{N_a}{N_d} S_p$$

Problem

Doping data of an abrupt Si diode:

$N_d = 10^{18}/\text{cm}^3$, $N_a = 10^{16}/\text{cm}^3$.

Calculate the widths of the depletion layers!

($\epsilon_r = 11.8$, $U = 0\text{V}$)

$$S_p = \sqrt{\frac{2 \cdot 11.8 \cdot 8.86 \cdot 10^{-12}}{1.6 \cdot 10^{-19} \cdot 10^{22}}} \sqrt{0.838} = 0.331 \mu\text{m} \quad S_n = 0.003 \mu\text{m}$$

And if $U = -100\text{V}$?

$$S_p = 0.331 \cdot \sqrt{\frac{0.838 + 100}{0.838}} = 3.63 \mu\text{m}$$



Budapest University of Technology and Economics
Department of Electron Devices

Microelectronics, BSc course

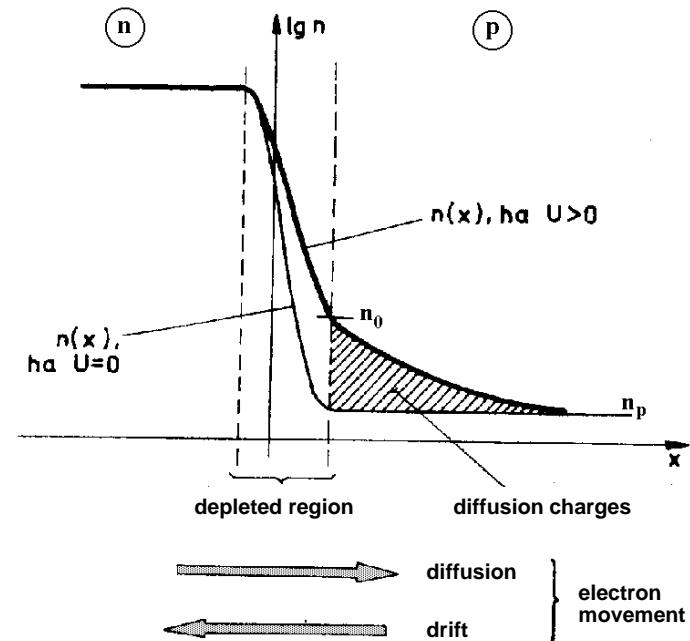
**Operation of PN junctions:
Characteristics**

Diode characteristics

- Forward and reverse mode operation
- Ideal characteristic
- Secondary effects

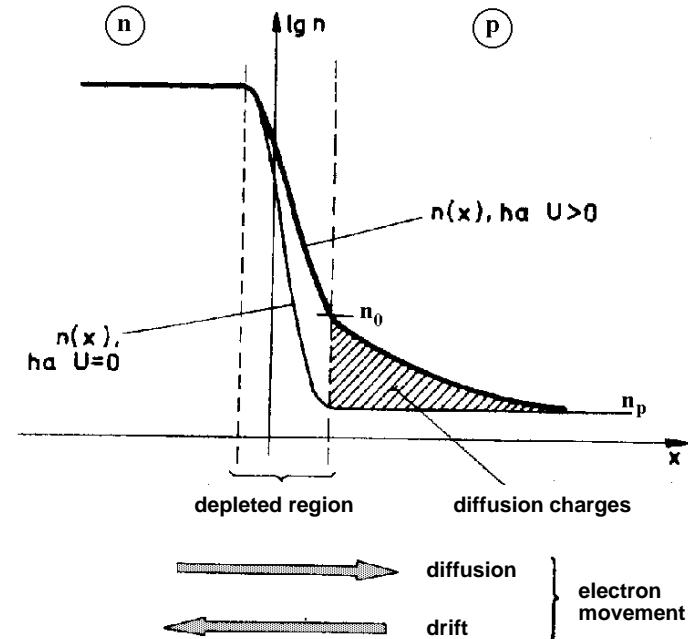
Forward operation of the diode

- ▶ Effect of forward (positive) U
 - ▶ Potential step decreases so as the electric field in the space charge region
 - ▶ Current balance disrupted: diffusion current became dominant, e^- diffusion from the n to the p region
- ▶ Majority carriers injected to the other side by diffusion
- ▶ e^- accumulation in the p region near the junction
- ▶ Diffusion carrier appear within the L_D distance (10um)



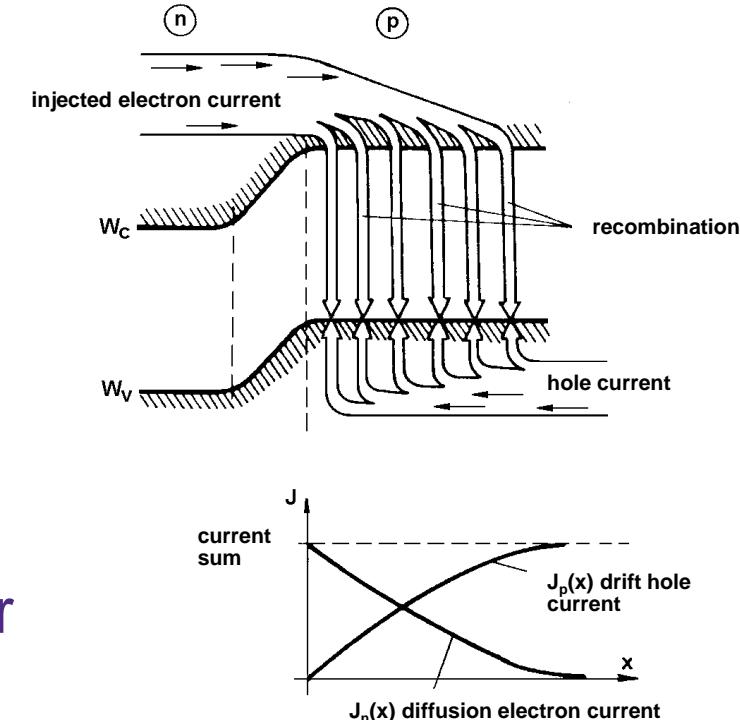
Forward operation of the diode

- ▶ Diffusion carrier accumulation:
 - ▶ $\text{grad}(n)$ decreases along the depleted region → **e^- current decreases in the depleted region**
 - ▶ $\text{grad}(n)$ in the p region → e^- that got through the junction moves further away (**diff. current**) from the junction towards the contact
 - ▶ Diffusion carriers increase still the two current become equal
- ▶ This balanced current determines the current of the PN junction!



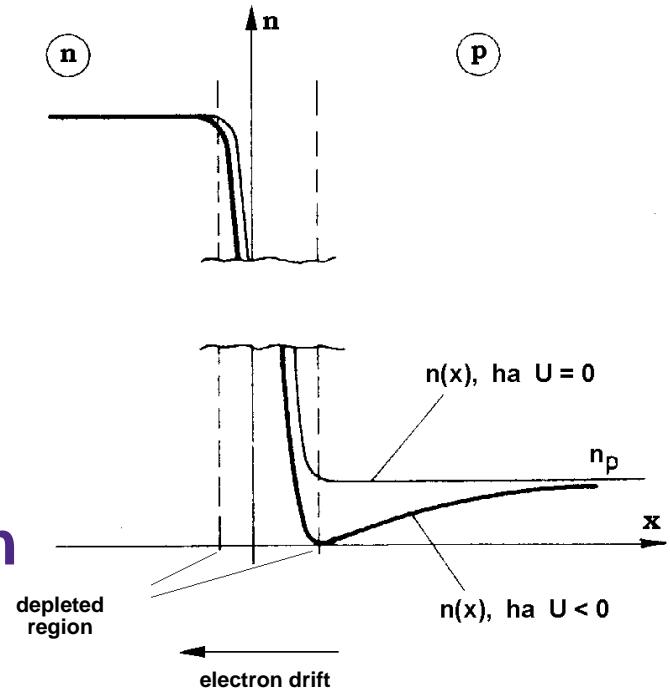
Forward operation of the diode

- ▶ The role of the e^- current is taken over by the holes in the p region!
- ▶ **Wide base structure**
 - ▶ Every e^- that got through the junction and recombines in the p region
- ▶ **Narrow base structure**
 - ▶ The thickness of p region is smaller than the diffusion length \rightarrow only a part of the e^- 's recombination
 - ▶ No difference in the (current) conduction!
 - ▶ Electric field moves the holes from the contact to the PN junction
 - ▶ *Small part of the forward voltage is dropped here*



Reverse operation of the diode

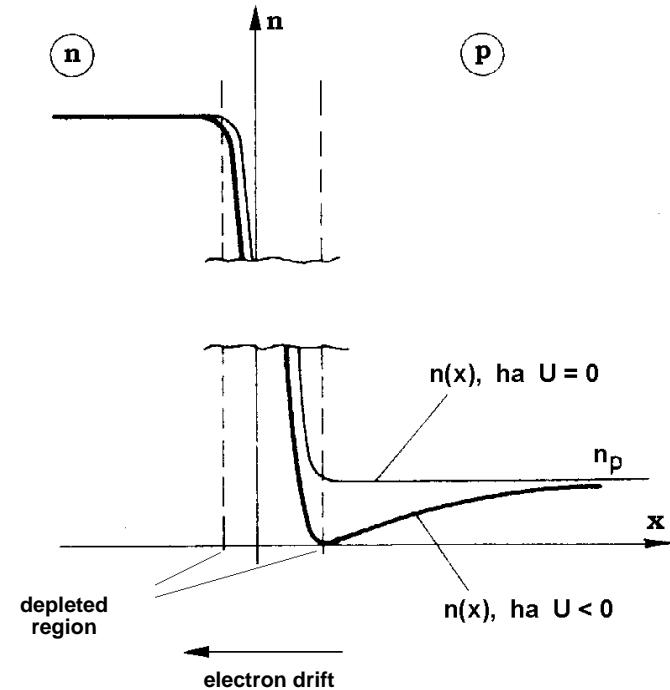
- ▶ Effect of reverse (negative) U
 - ▶ Potential step increases so as the electric field in the space charge region
 - ▶ Current balance disrupted: drift current became dominant, e^- drift from the p to the n region
 - ▶ **Drift of the minority carriers on both sides towards the other region!**
 - ▶ e^- concentration decreases in the p region near the junction
 - ▶ PN junction behaves as a sink for the minority carriers!



Reverse operation of the diode

- ▶ Reverse current is determined by the carrier generation rate

- ▶ e⁻ „supply” in the p region is determined by the generation rate.
- ▶ ~ $10^{10} \dots 10^{12}$ carrier/s/cm³ in Si
SLOW
- ▶ In the range of nA!

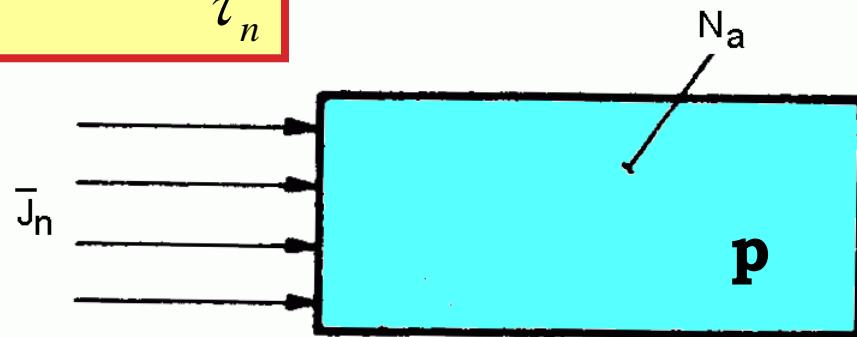


- ▶ I_R - is not affected by the U_R !
- ▶ Generation is not affected by the E in the junction!
- ▶ + Generation is a secondary effect!

Reminder – Diffusion eq.

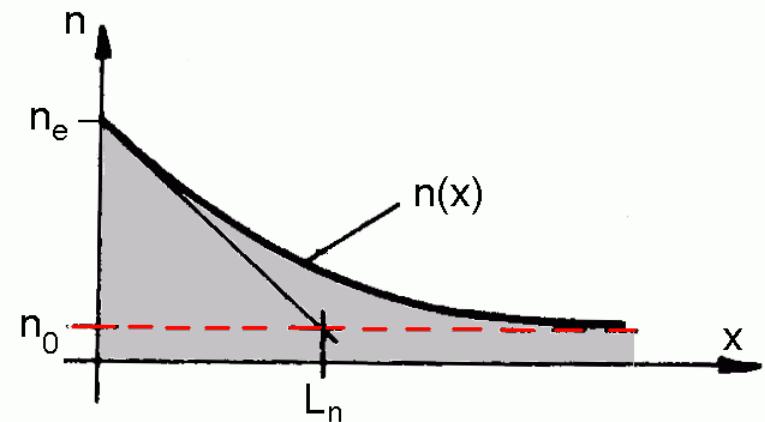
$$\frac{dn}{dt} = \mu_n \operatorname{div}(n \bar{E}) + D_n \operatorname{divgrad} n + g_n - \frac{n}{\tau_n}$$

$$0 = D_n \frac{d^2 n}{dx^2} + \frac{n_0}{\tau_n} - \frac{n}{\tau_n}$$



$$n(x) = n_0 + (n_e - n_0) \exp(-x / \sqrt{D_n \tau_n})$$

$$L_n = \sqrt{D_n \tau_n}$$



The ideal diode characteristic

$$n(x) = n_p + (n_0 - n_p) \exp(-x / L_n)$$

$$n_0 = n_p \exp\left(\frac{U}{U_T}\right)$$

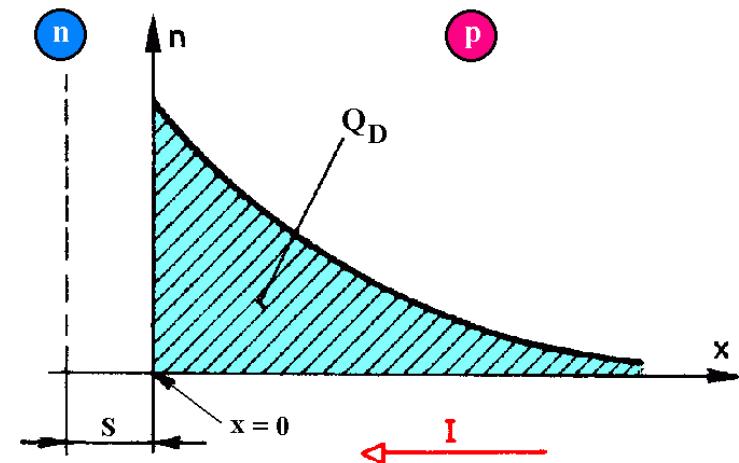
$$J_n(x) = -qD_n \frac{dn}{dx} = -qD_n (n_0 - n_p) \exp(-x / L_n) \left(\frac{-1}{L_n} \right)$$

$$J_n|_{x=0} = \frac{qD_n}{L_n} (n_0 - n_p) = \frac{qD_n n_p}{L_n} (\exp(U / U_T) - 1)$$

$$n_0 = n_p \exp\left(\frac{q \cdot U}{k \cdot T}\right) = n_p \exp\left(\frac{U}{U_T}\right)$$

$$J_p = \frac{qD_p p_n}{L_p} (\exp(U / U_T) - 1)$$

$$I = A(J_n + J_p)$$



The ideal diode characteristic

$$J_n \Big|_{x=0} = \frac{qD_n n_p}{L_n} (\exp(U / U_T) - 1)$$

$$I = I_0 (\exp(U / U_T) - 1)$$

$$J_p = \frac{qD_p p_n}{L_p} (\exp(U / U_T) - 1)$$

I_0 is proportional with the minority carrier concentrations

$$I = A(J_n + J_p)$$

$$I = Aq \underbrace{\left(\frac{D_n n_p}{L_n} + \frac{D_p p_n}{L_p} \right)}_{I_0} (\exp(U / U_T) - 1)$$

The ideal diode characteristic

$$I = I_0 (\exp(U / U_T) - 1)$$

$$U = U_T \ln(I / I_0 + 1)$$

Problem

Saturation current of Si diode: $I_0 = 10^{-13} \text{ A.}$

What is U_F , if I_F is 10 mA?

$$U \cong 0.026 \cdot \ln(10^{-2} / 10^{-13}) = 0.658 \text{ V}$$

Problem

How much should we increase the forward voltage if we want to increase the current 10x ?

$$\Delta U = U_2 - U_1 \cong U_T (\ln(I_2 / I_0) - \ln(I_1 / I_0)) = U_T \ln(I_2 / I_1)$$

$$\Delta U = 0.026 \cdot \ln 10 \cong 0.06 \text{ V} = 60 \text{ mV}$$

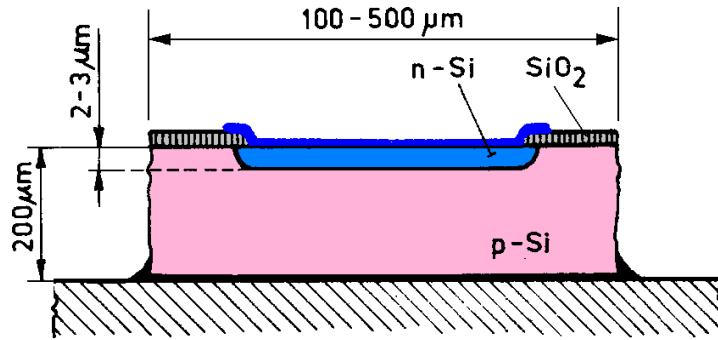
Secondary effects

- Series resistance
- Generation current
- Breakdown phenomena (a bit later)
- Recombination current (just mention)

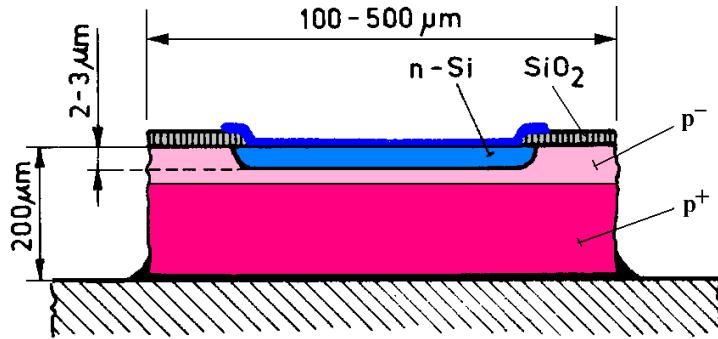
Secondary effects

The series resistance

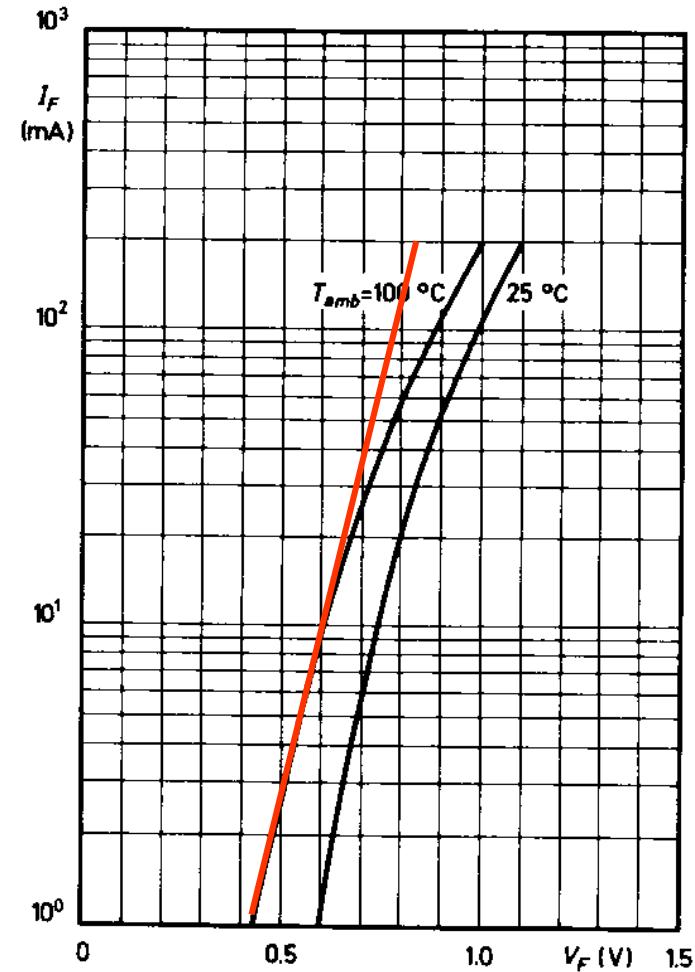
Appears at high current levels. Reason:



Solution: epitaxial structure



Forward characteristics $I_F = f(V_F)$



Secondary effects

The series resistance

Calculate the series resistance according to the 100°C characteristic!

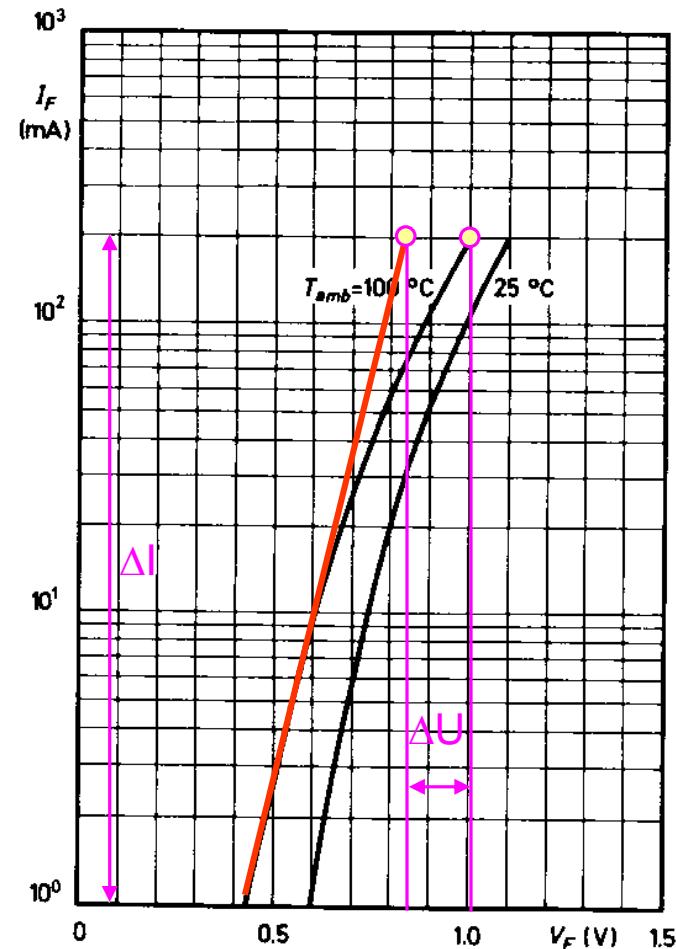
$$\Delta U = 160 \text{ mV}$$

$$I = 200 \text{ mA}$$

$$r_s = 160 / 200 = 0,8 \Omega$$

Problem

Forward characteristics $I_F = f(V_F)$



Secondary effects

The generation current

In reverse region, in theory:

$$I = I_0 (\exp(U / U_T) - 1) \Rightarrow -I_0$$

that would result in pA only

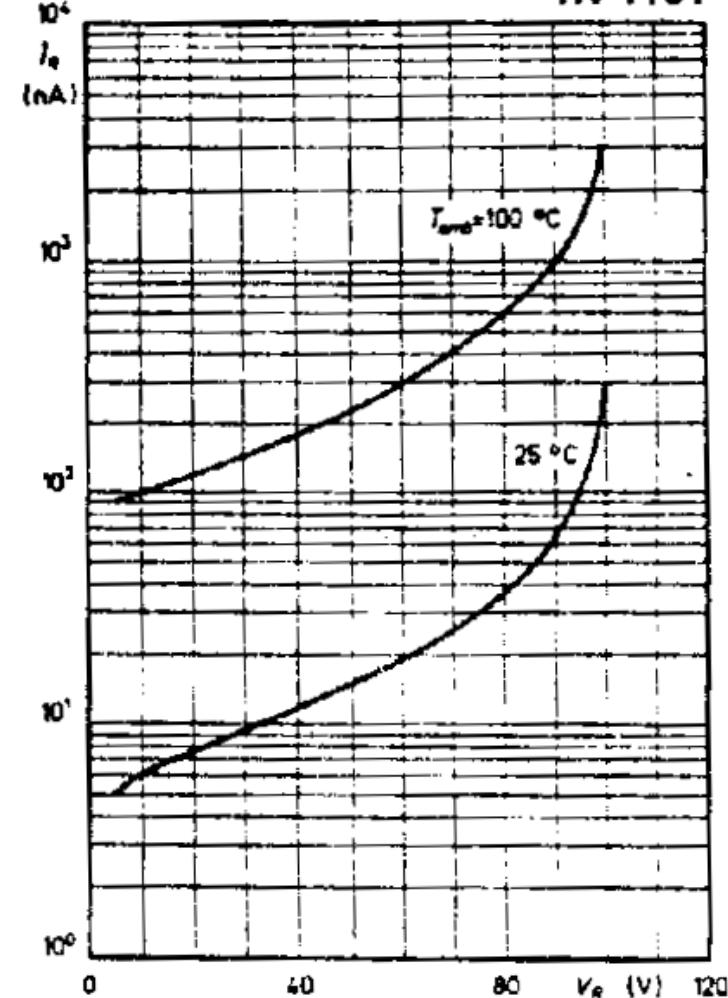
The experience is:

$$g = \frac{n_i}{2\tau}$$

$$I_R = \text{const} \cdot n_i \sqrt{-U_R}$$

Reverse characteristics

1N 4151



Secondary effects

The recombination current

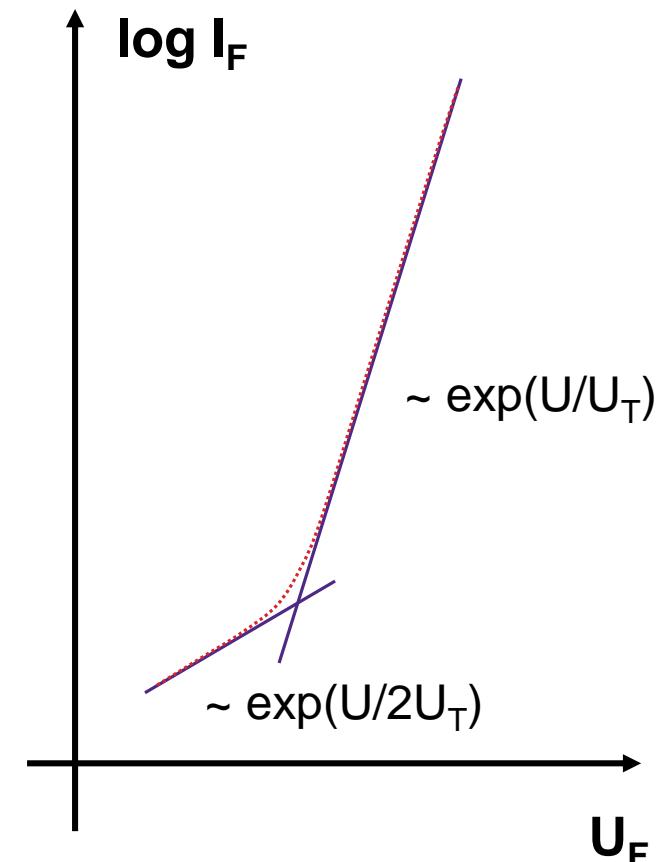
Phenomenon appearing in the forward region

$$I_{Rec} \approx const \cdot n_i \cdot \exp\left(U / 2U_T\right)$$

Can be well described by the Shockley-Read-Hall model for semiconductors with indirect band

$$I = I_0 \left(\exp\left(U / mU_T\right) - 1 \right) = -I_0$$

m: non-ideality factor, between 1..2



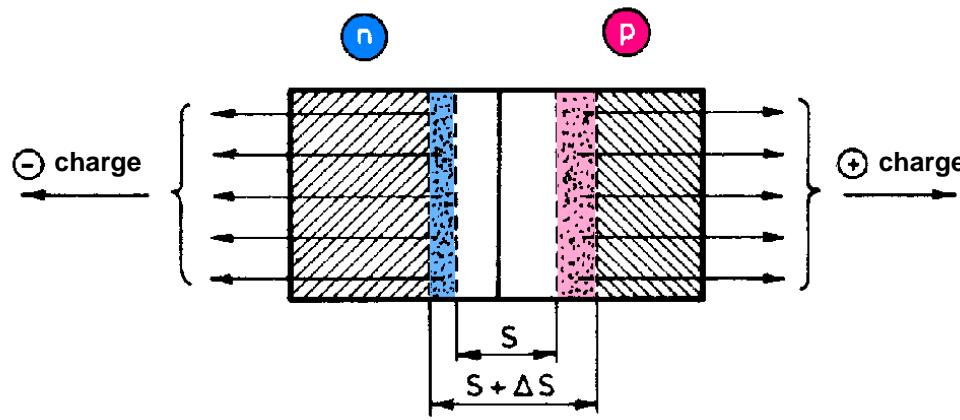
Capacitances of a diode

- Space charge capacitance
- Diffusion capacitance

Capacitances of the diode

Space charge capacitance

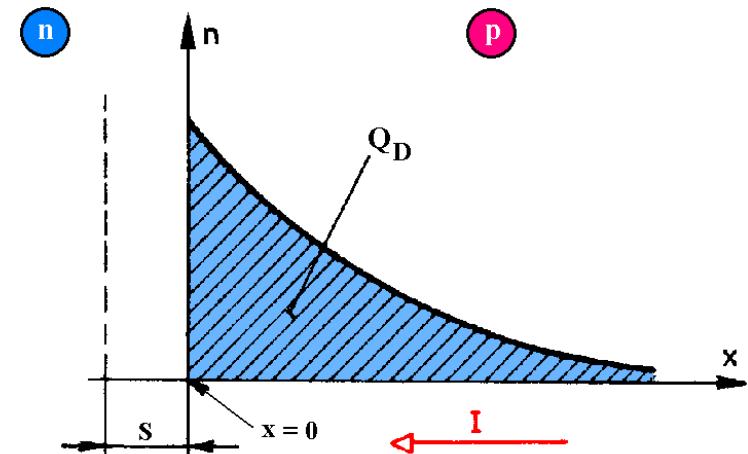
dominates in the reverse region



Interpretation as a differential at a given forward voltage/current

Diffusion capacitance

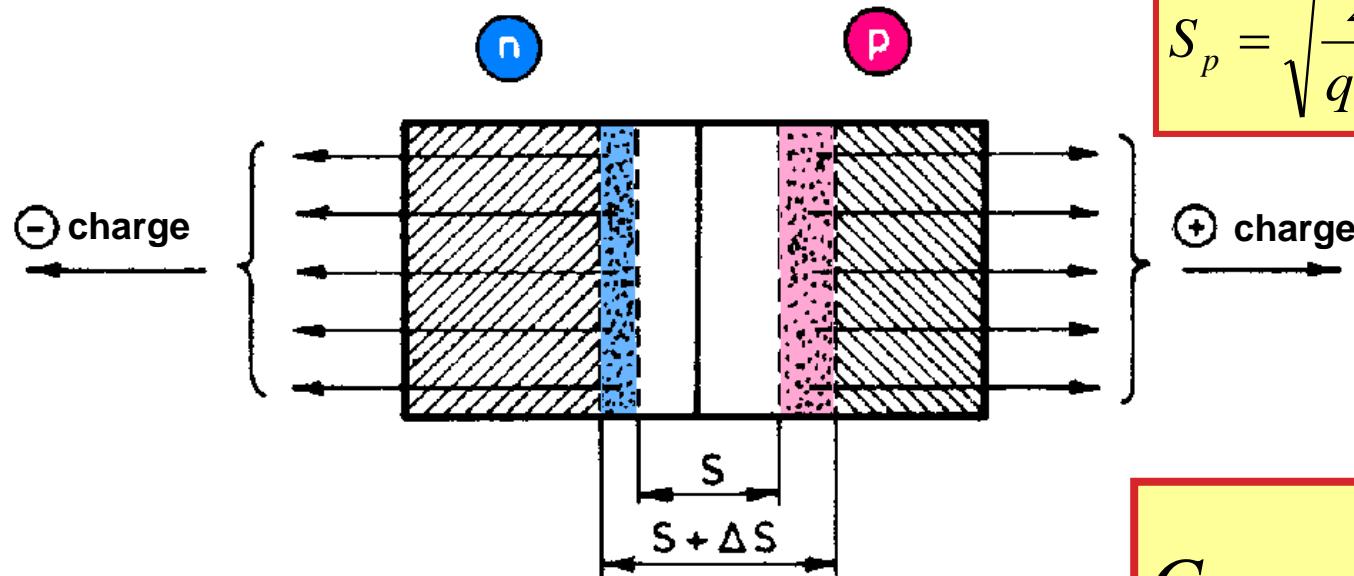
in forward region only



$$C = \frac{dQ}{dU}$$

Capacitances of the diode

The space charge capacitance



$$S_p = \sqrt{\frac{2\epsilon}{qN_a}} \sqrt{U_D - U}$$

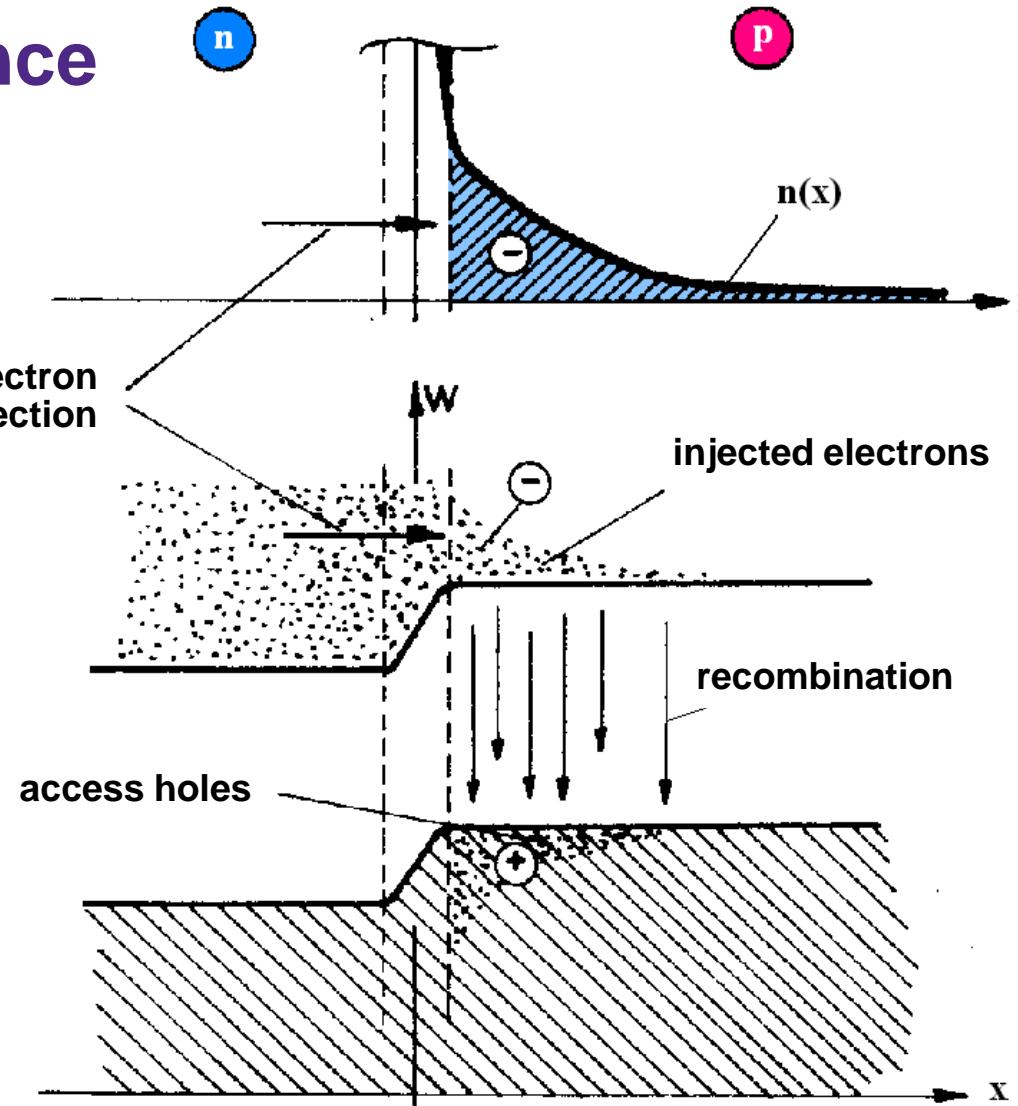
$$C_{Sp} = \frac{const}{\sqrt{U_D - U}}$$

$$C_{Sp} = \epsilon \frac{A}{S} = \epsilon A \sqrt{\frac{qN_a}{2\epsilon}} \frac{1}{\sqrt{U_D - U}} = A \sqrt{\frac{q\epsilon N_a}{2}} \frac{1}{\sqrt{U_D - U}}$$

Capacitances of the diode

The diffusion capacitance

Where are the opposite charges?

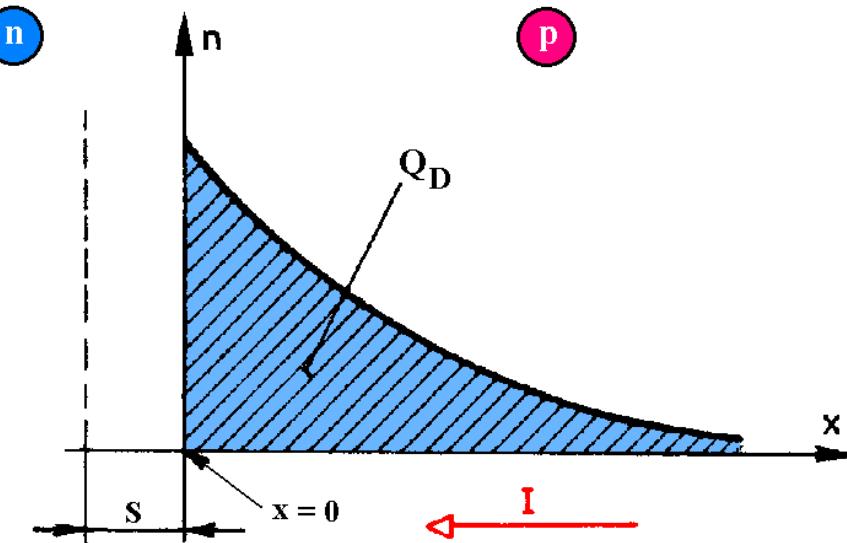


Capacitances of the diode

The diffusion capacitance

$$Q_D = I \tau_{n(p)}$$

$$C_D = \frac{dQ_D}{dU} = \frac{dQ_D}{dI} \frac{dI}{dU}$$



$$C_D = \tau_{n(p)} \frac{1}{r_d} = \tau_{n(p)} \frac{I}{U_T} = \text{const} \cdot I$$

$$C_D = \text{const} \cdot I$$

Harmful! Slows down the operation.

Reduction: decrease τ , narrow base diode

Capacitances of the diode

Problem

Let us calculate the space charge capacitance of a Si diode if the width of the depletion layer is 0.33 μm and the cross-sectional area is 0.02 mm²!

$$C_{Sp} = \epsilon \frac{A}{S} = 11.8 \cdot 8.86 \cdot 10^{-12} \frac{2 \cdot 10^{-8}}{0.33 \cdot 10^{-6}} = 6.34 \cdot 10^{-12} F = 6.34 \text{ pF}$$

Let us calculate the diffusion capacitance in the operating point of I=1 mA if τ=100 ns!

$$C_D = \tau \frac{I}{U_T} = 10^{-7} \frac{10^{-3}}{0.026} = 3.85 \cdot 10^{-9} F = 3.85 \text{ nF}$$

Capacitances of the diode

Orders of magnitude:

C_{Sp} 1-10 pF

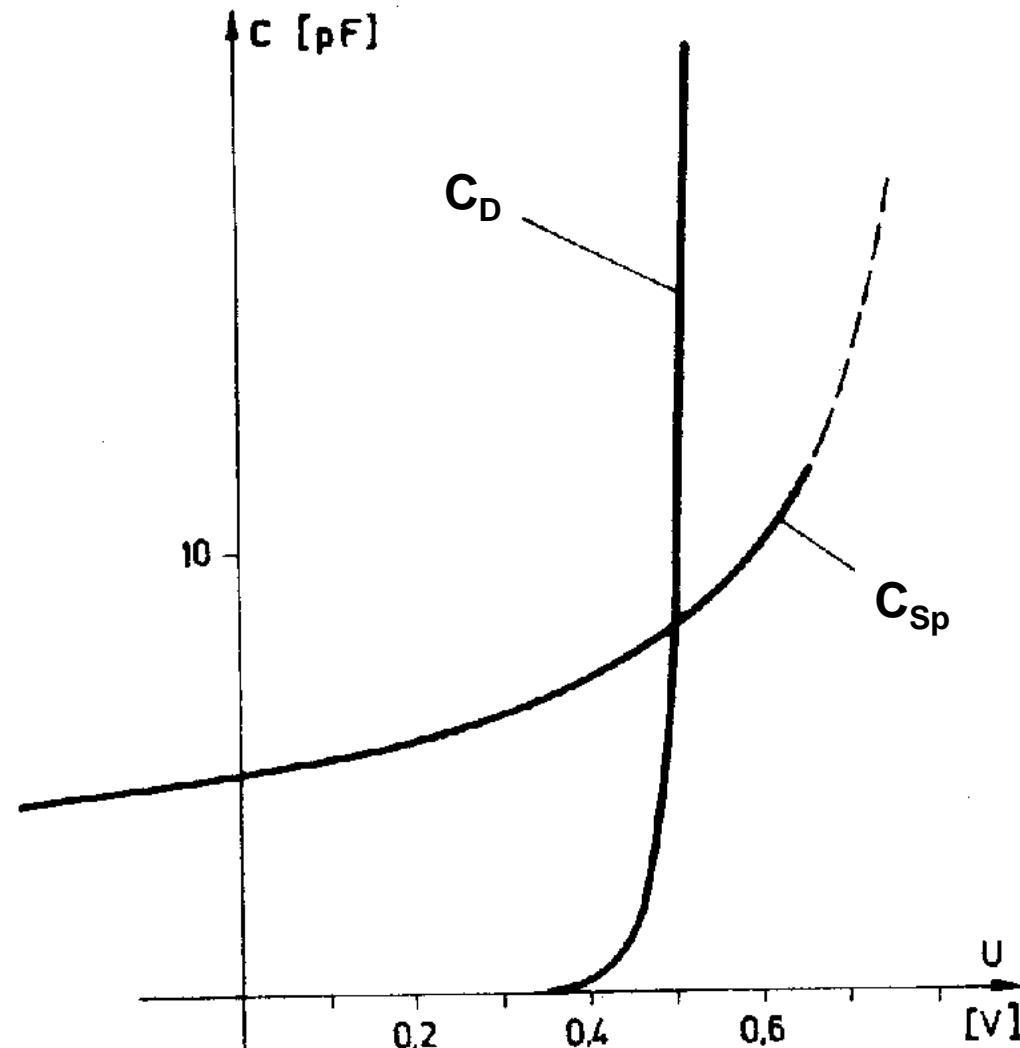
C_D nF-s

(for a small power diode)

Utilization

C_{Sp} tuning oscillators,
microwave amplification

C_D --



Operating point

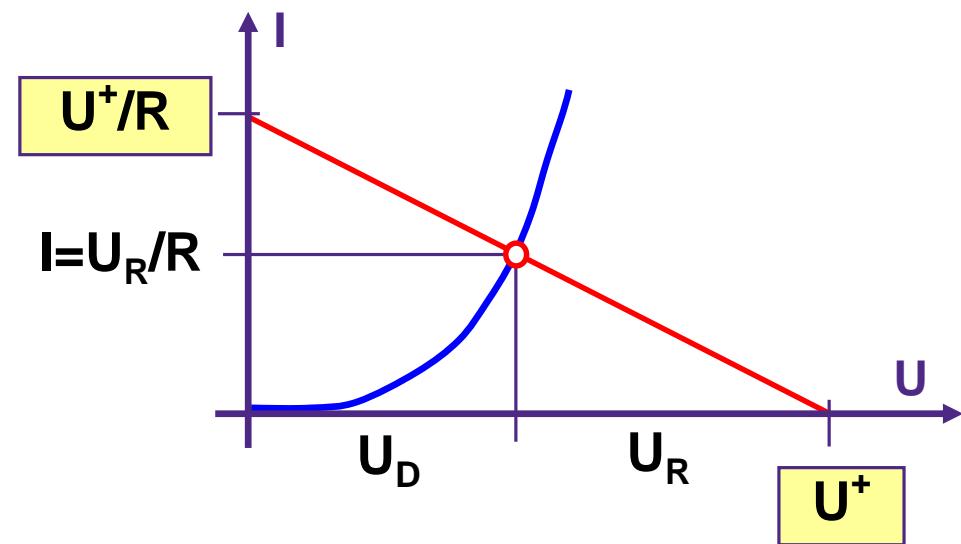
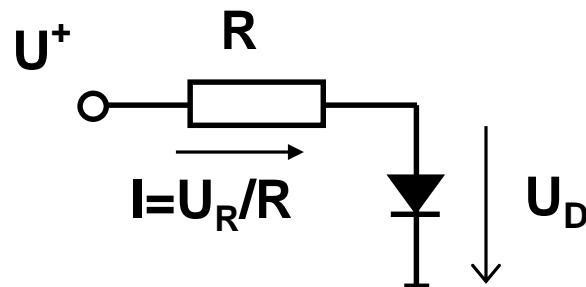
- Finding the DC operating point
- Linearization in the operating point, small signal operation
- Differential resistance, capacitance
- Models

The operating point

- **Characteristics:** defines the current-voltage pairs that may occur during the operation.
- During the real operation the **diode or any nonlinear element** works in one point of the characteristics, that is the ***operating point, or quiescent point.***
- This is determined also by the surrounding elements.

Finding the operating point

The problem: a linear element and a non-linear element connected in series:

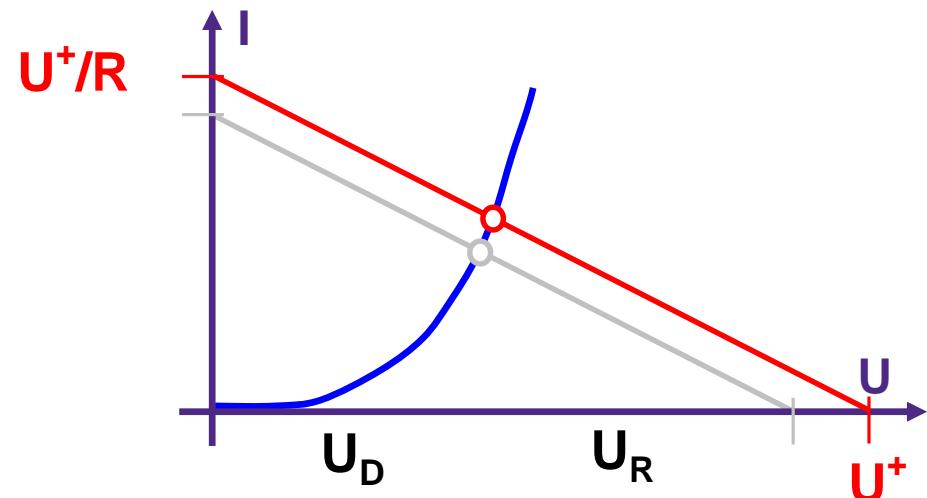
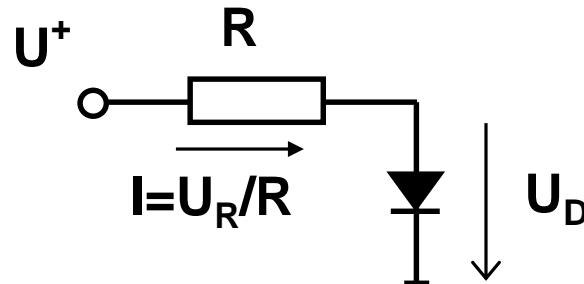


$$\left. \begin{aligned} I &= I(U_D) \\ I &= (U^+ - U_D)/R \end{aligned} \right\}$$

Graphical solution

Finding the operating point

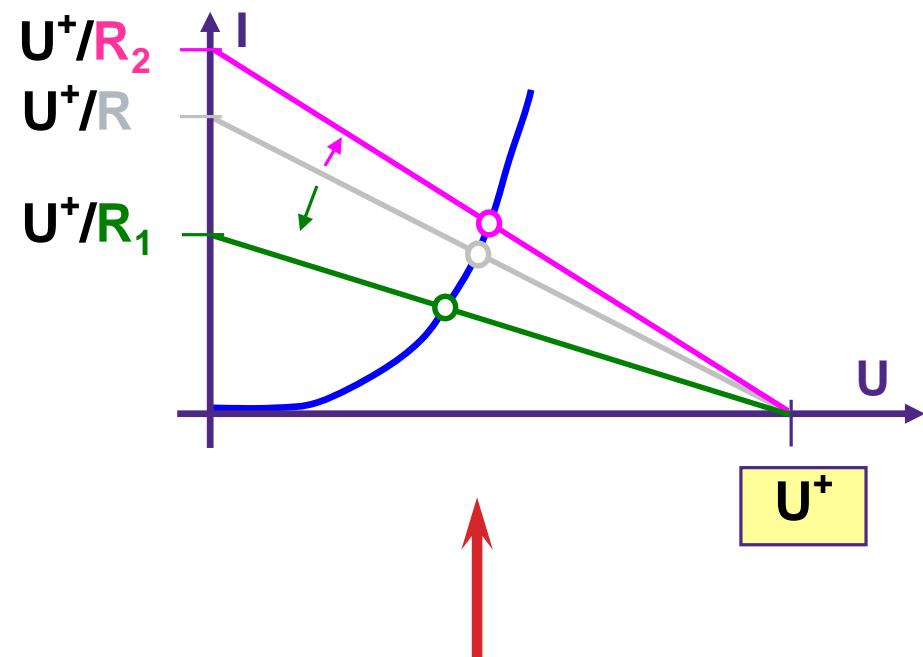
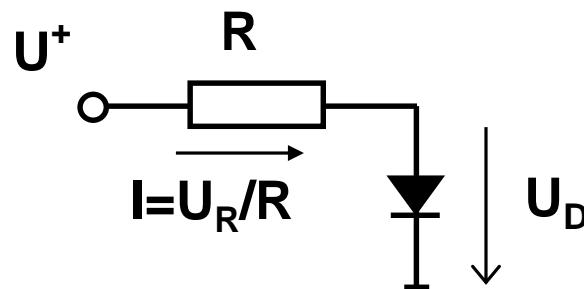
How does the operating point change if the U^+ supply voltage is increased?



The operating line is shifted in parallel

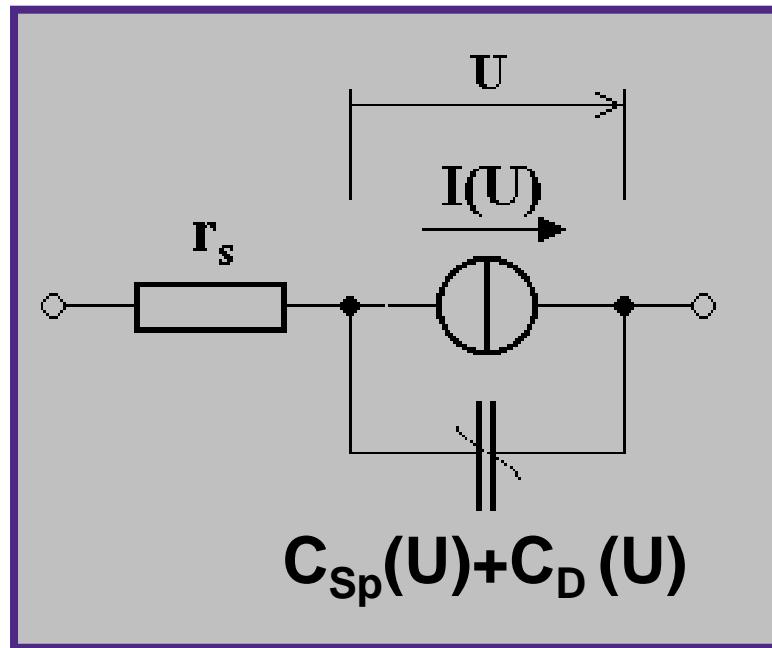
Finding the operating point

How does the operating line change if we change R ?



It turns around point U^+ - its slope will change

Large signal model of the diode



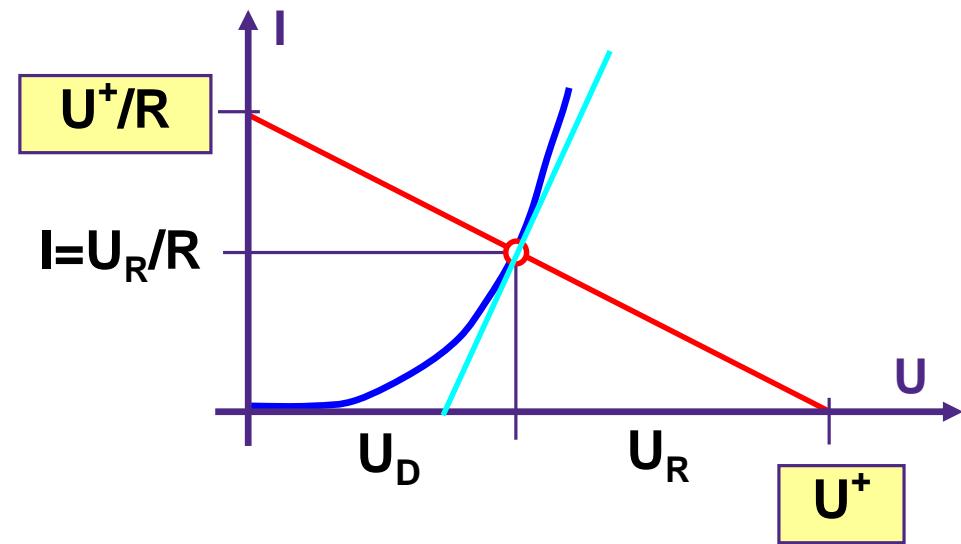
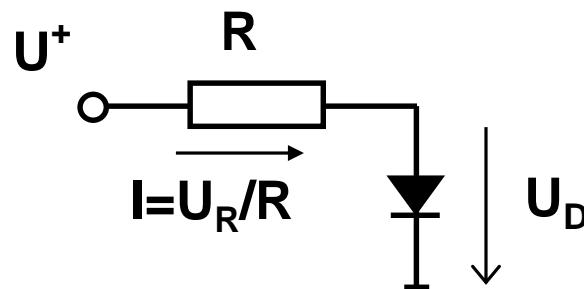
The computer simulation model also looks like this.

Also needed:

model equations (e.g. $I=I_0(\exp(U/U_T)-1)$)

model parameters (e.g.. I_0 , r_s , etc.)

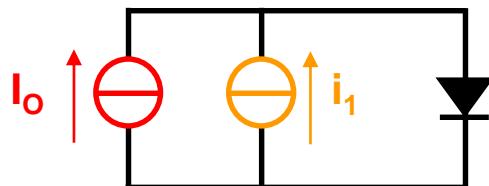
Linearization in the operating point



For small changes we can linearize the characteristics

Small signal operation of diodes

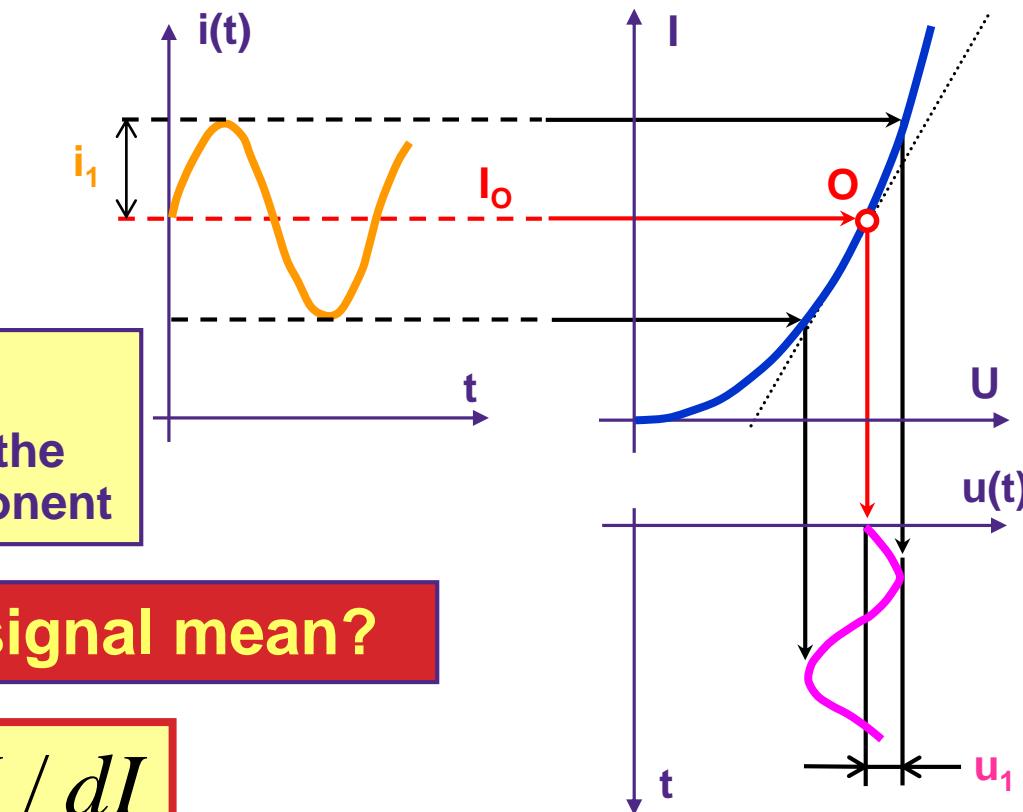
The operating point



Small signal:
linearized analysis, for the
 alternating current component

What does small signal mean?

$$r_{diff} = u_1 / i_1 = dU / dI$$



r_{diff} operating point dependent

Differential resistance of the diode

$$U = U_T \ln(I / I_0 + 1)$$

$$r_d = dU / dI = U_T \frac{1}{I / I_0 + 1} \frac{1}{I_0} = \frac{U_T}{I + I_0}$$

Forward region, $I \gg I_0$:

$$r_d = \frac{U_T}{I}$$

If we consider the series resistance as well:

$$r_d = \frac{U_T}{I} + r_s$$

Differential resistance of the diode

Problem

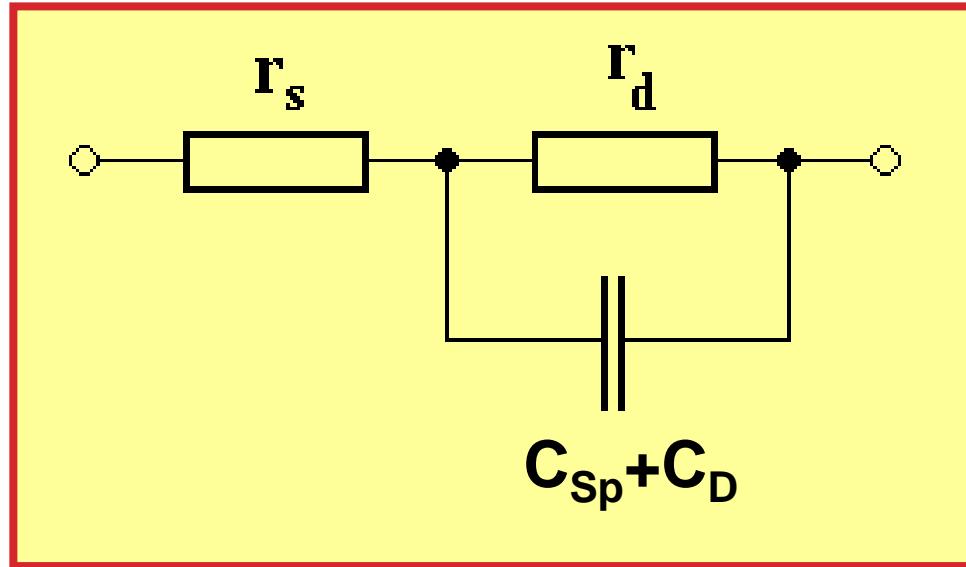
The series resistance of a diode is 2Ω . Let us calculate its differential resistance in the $I=1 \text{ mA}$, 10 mA , 100 mA operating points!

$$r_d|_{1mA} = \frac{26}{1} + 2 = 28 \Omega$$

$$r_d|_{10mA} = \frac{26}{10} + 2 = 4.6 \Omega$$

$$r_d|_{100mA} = \frac{26}{100} + 2 = 2.26 \Omega$$

Small signal model of the diode



Element values are operating point dependent!

Recall:

$$r_d = \frac{U_T}{I} \quad C_{Sp} = \frac{const}{\sqrt{U_D - U}} \quad C_D = const \cdot I$$

Temperature dependence

Temperature dependence

- ▶ The characteristics shows *strong temperature dependence*
- ▶ Reason: temperature dependence of the minority carriers
 - **Forward voltages:** V_F at I_F **decreases with about 2mV for 1°C increase**
 - *linear temperature dependence in a large range* → appropriate for temperature measurements
 - **Reverse voltages:** I_R at U_R **decreases with $\approx 7\text{-}10\%$ for 1°C**
(that means doubling at each 10 °C)

Temperature dependence

Reverse region:

For a Si diode: $I_R \sim n_i \rightarrow \sqrt{1,15} \cong 1,075 \rightarrow 7,5\text{ \%}/^\circ\text{C}$

Forward region:

$$U = U_T \ln \frac{I}{I_0} = \frac{kT}{q} \ln \frac{I}{I_0(T)} \quad \frac{d}{n_i^2} = \left(3 + \frac{W_g}{kT} \right) \frac{dT}{T} = \frac{dI_0}{I_0}$$

$$\frac{dU}{dT} = \frac{U}{T} + U_T \frac{I_0}{I} \left(\frac{-I}{I_0^2} \right) \frac{dI_0}{dT} = \frac{U}{T} - U_T \frac{1}{I_0} \frac{dI_0}{dT}$$

$$\boxed{\frac{dU}{dT} = \frac{U}{T} - U_T \left(3 + \frac{W_g}{kT} \right) \frac{1}{T} = \frac{U - 3U_T - W_g/q}{T}}$$

Temperature dependence

Forward region:

$$\frac{dU}{dT} = \frac{U - 3U_T - W_g / q}{T}$$

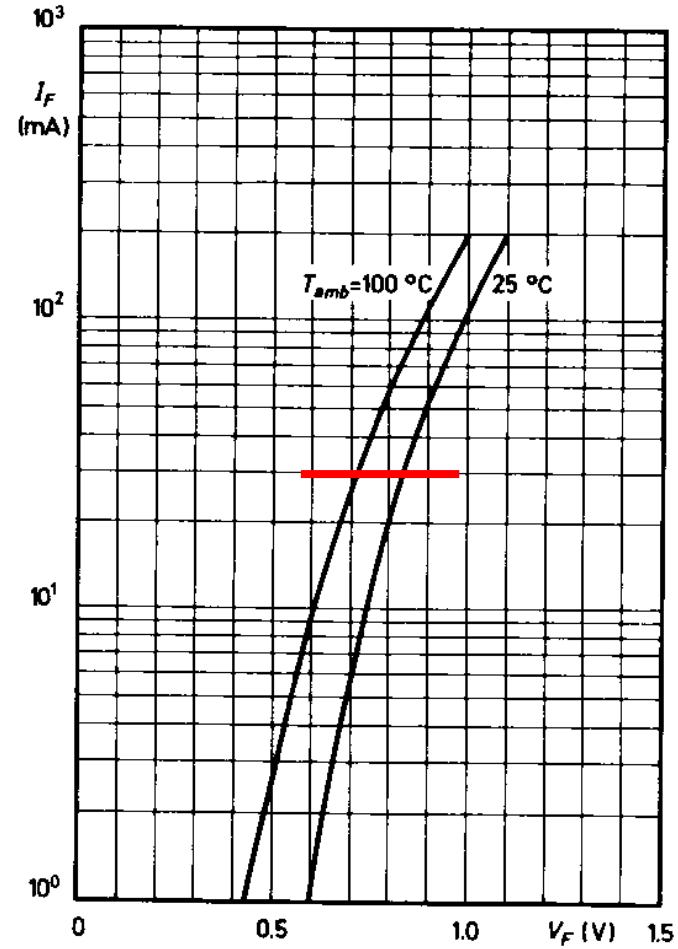
Problem

If $U=700$ mV, what is dU/dT for a Si diode?

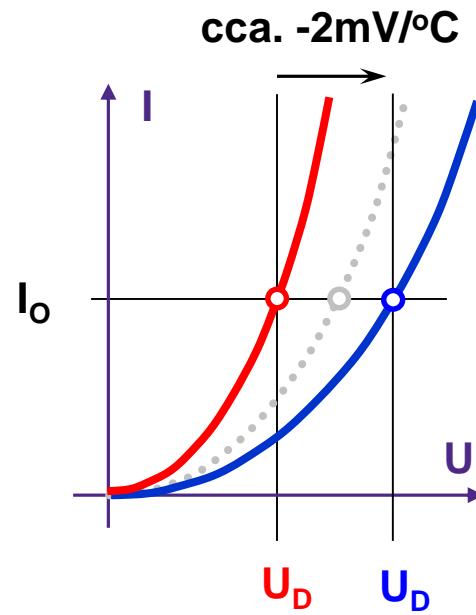
$$\frac{dU}{dT} = \frac{700 - 3 \cdot 26 - 1120}{300} = -1.66 \text{ mV } ^\circ C$$

Compare with the characteristics!

Forward characteristics $I_F = f(V_F)$



Temperature dependence



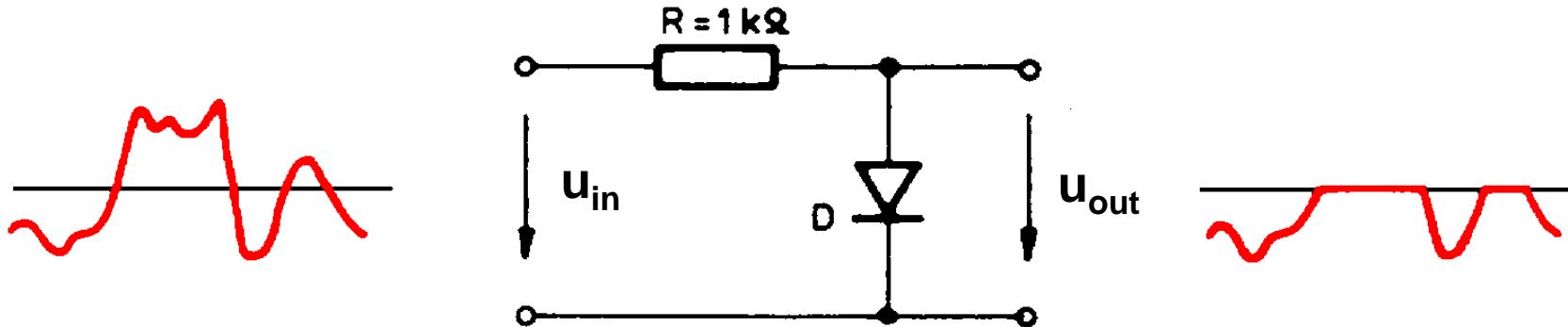
$$\frac{dU}{dT} = \frac{U - 3U_T - W_g / q}{T}$$

In case of a forced current the forward voltage of a pn junction is an excellent temperature sensor...

The sensitivity slightly depends on the I_o current

The diode in switching mode

Diodes as rectifiers

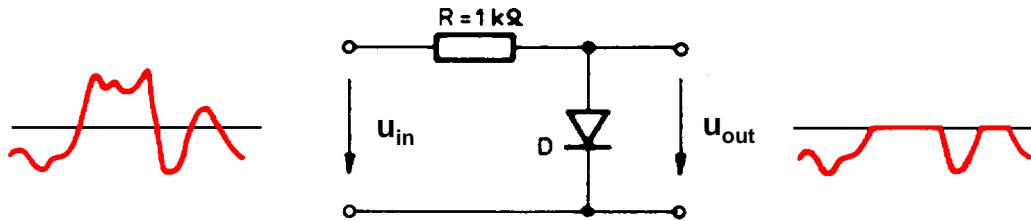


$$U_{out}(t) = \begin{cases} 0, & \text{if } U_{in}(t) \geq 0 \\ U_{in}(t), & \text{if } U_{in}(t) < 0 \end{cases}$$

The diode was considered to be ideal!

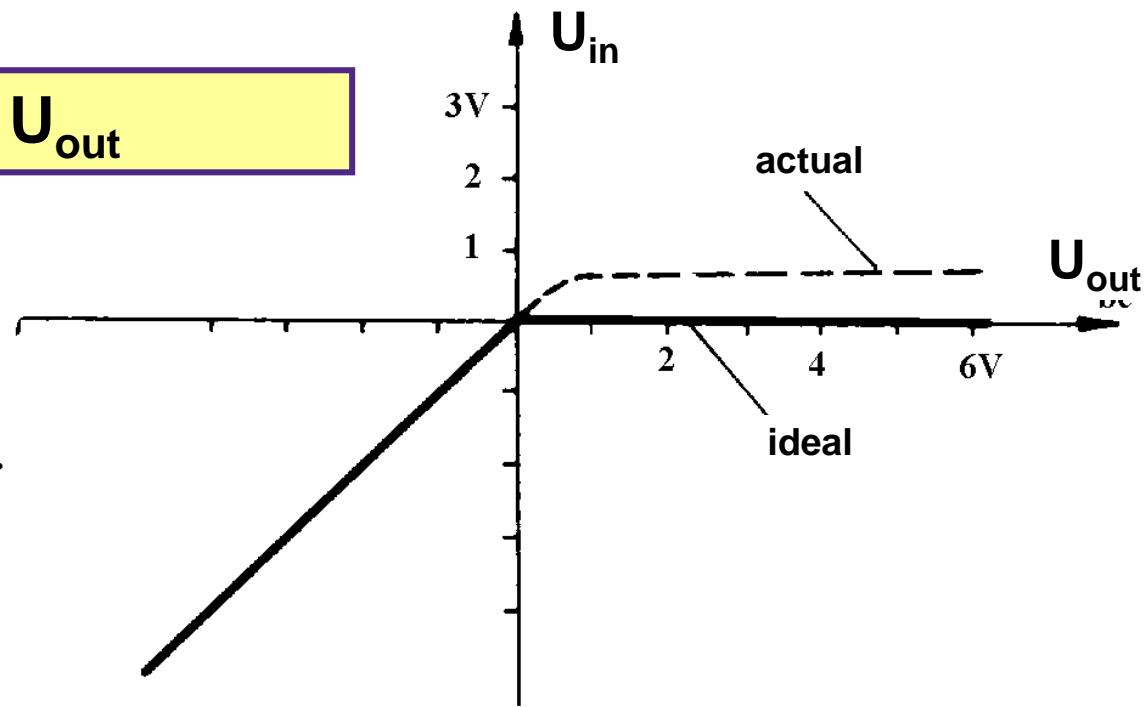
What if this is not the case?

Diodes as rectifiers

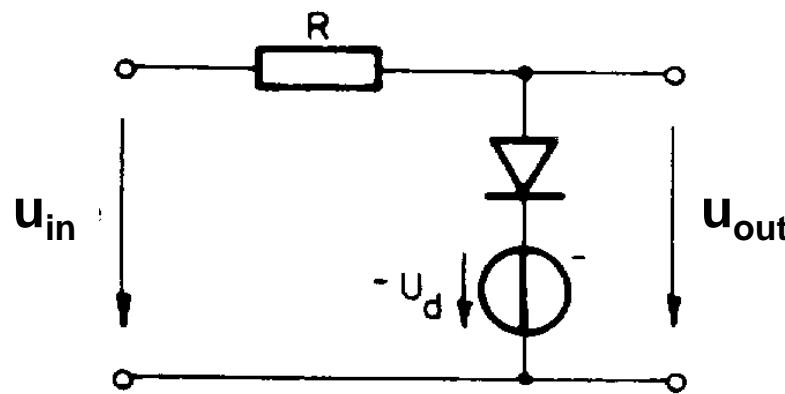
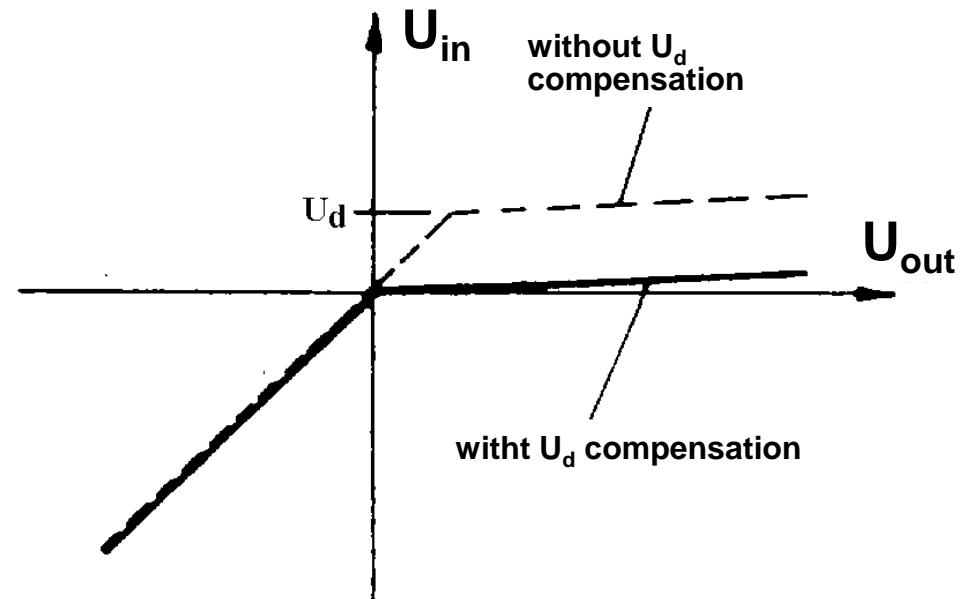
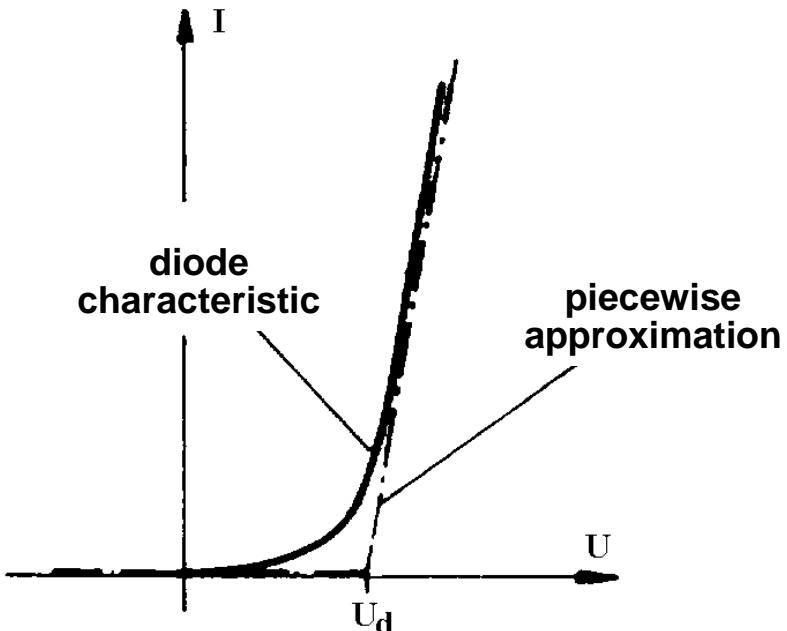


Relation of U_{in} and U_{out}

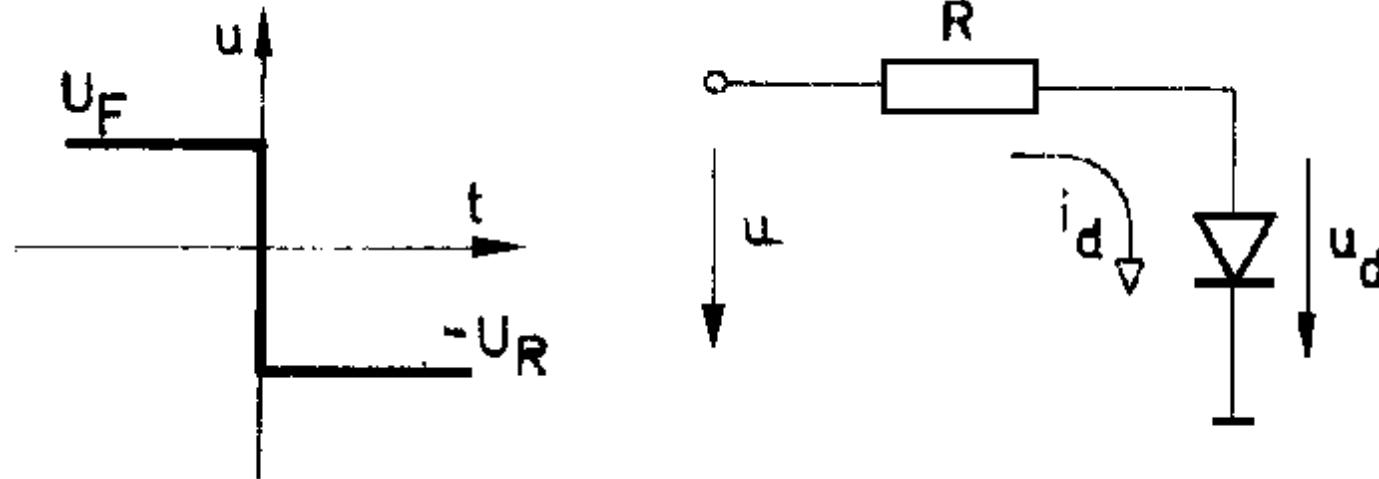
Transfer characteristic



Diodes as rectifiers

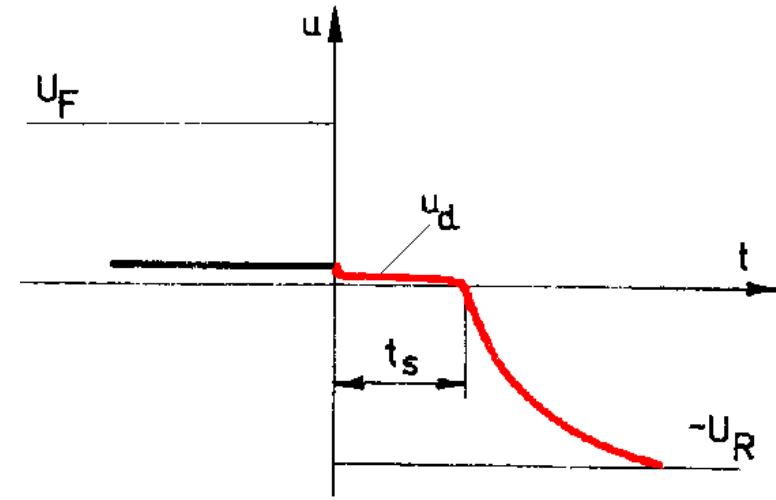
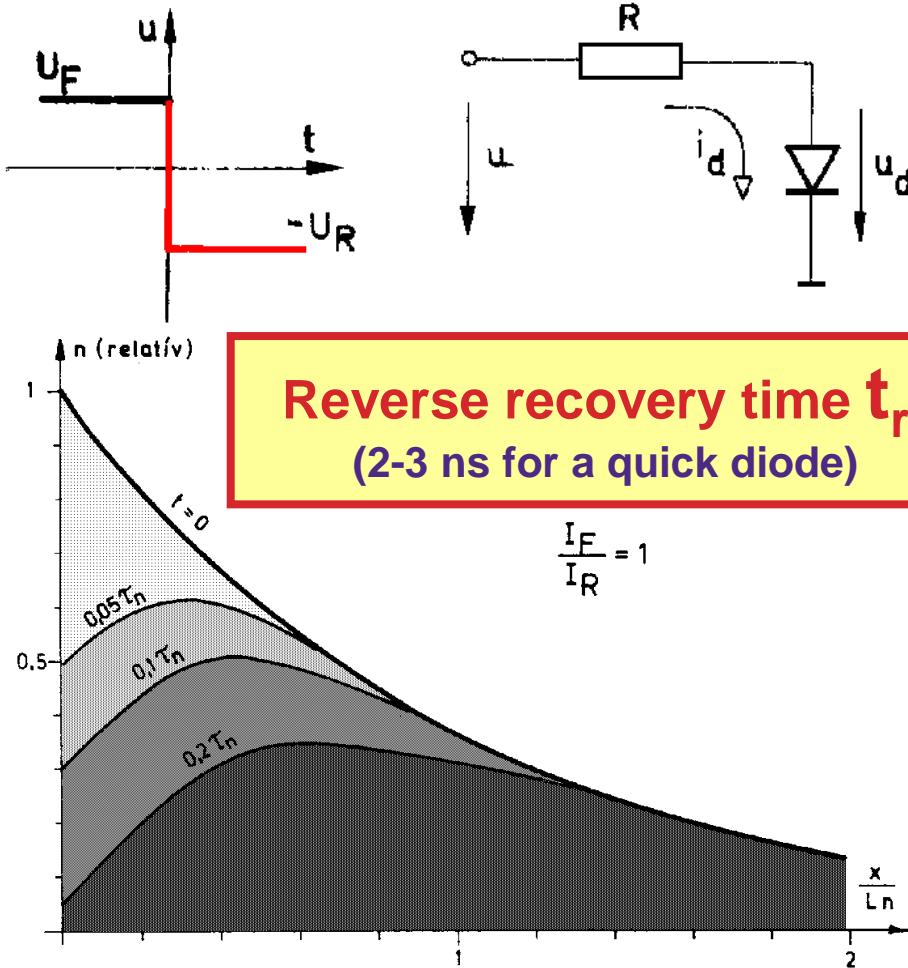


Transient phenomena



Abrupt switching from forward to reverse voltage:
due to its capacitances, the diode is open for some time.
This is called **reverse recovery**.

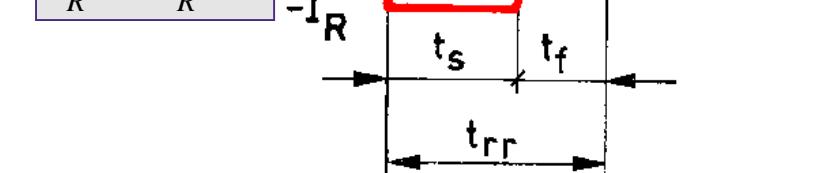
Reverse recovery



$$I_F = U_F / R$$

$$I_F$$

$$I_R = U_R / R$$

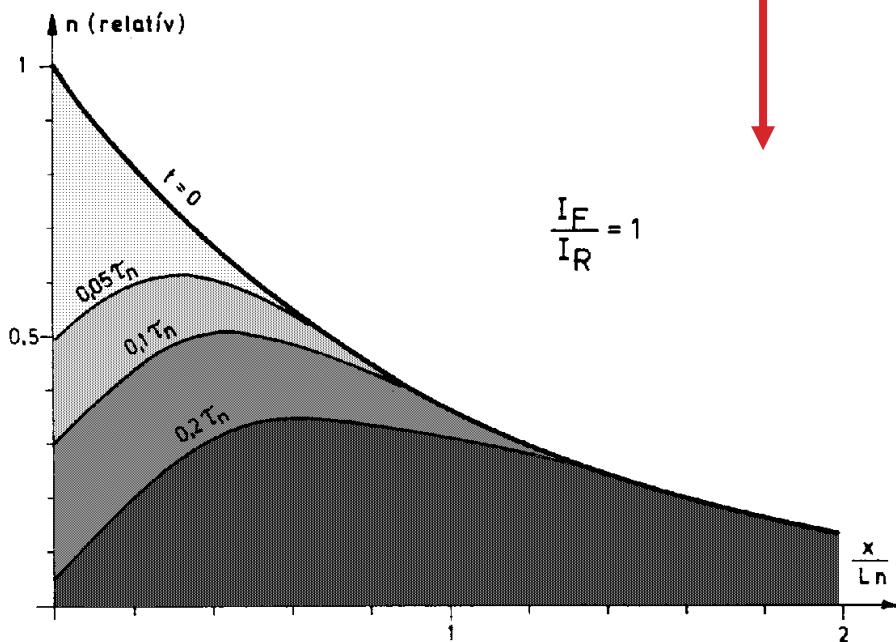


Transient behavior of the diode

The diffusion equation: →

We calculate $n(x,t)$ from this

$$\frac{\partial n}{\partial t} = D_n \frac{\partial^2 n}{\partial x^2} - \frac{n - n_p}{\tau_n}$$



Simplification:
instead of $n(x,t)$
we calculate with
 $Q(t)$

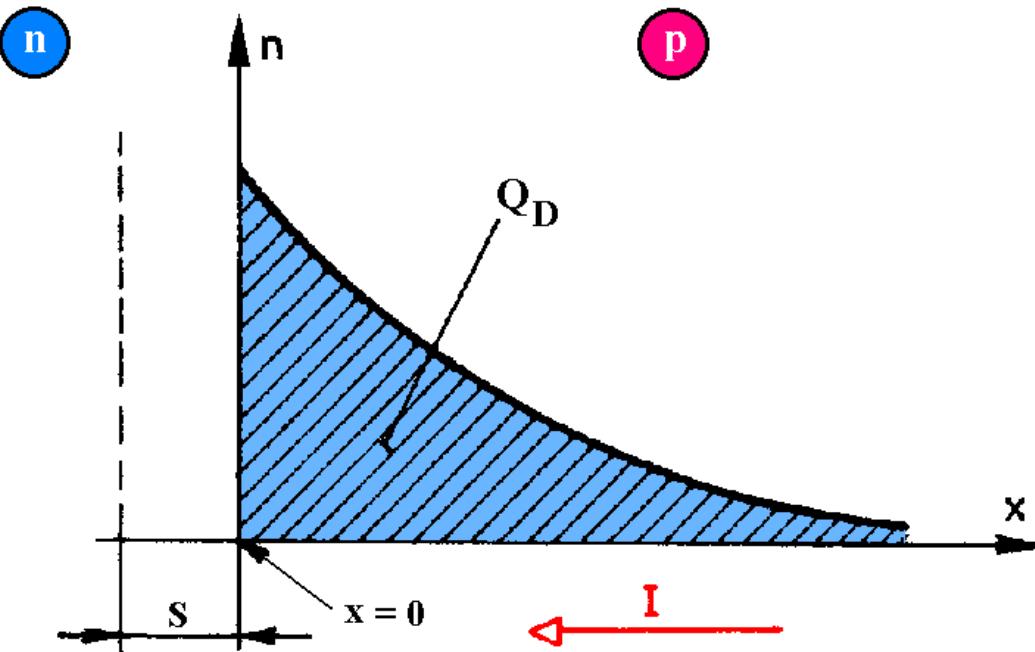
net charge

Transient behavior of the diode

The charge equation

$$Q_D = f(t)$$

$$I = \frac{Q_D}{\tau_{n(p)}} + \frac{dQ_D}{dT}$$



The current is spent on

maintaining recombination
depleting/supplying diffusion charge

Break-down phenomena

- Avalanche
- Zener

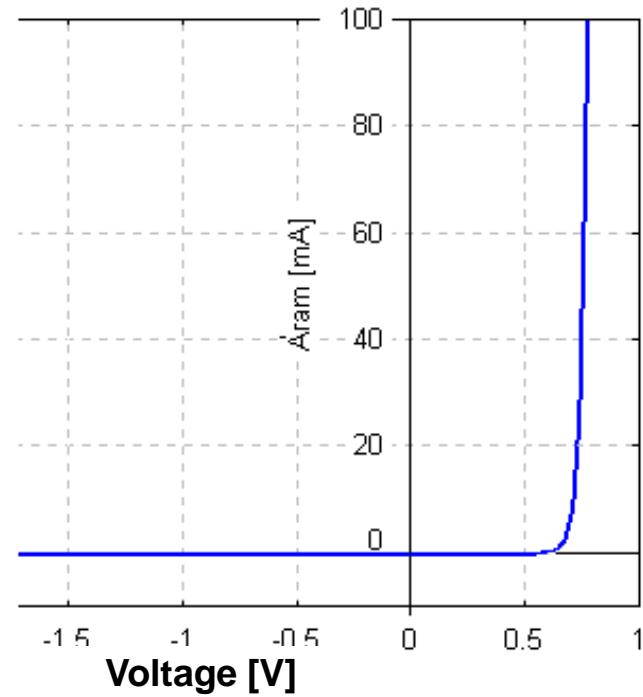
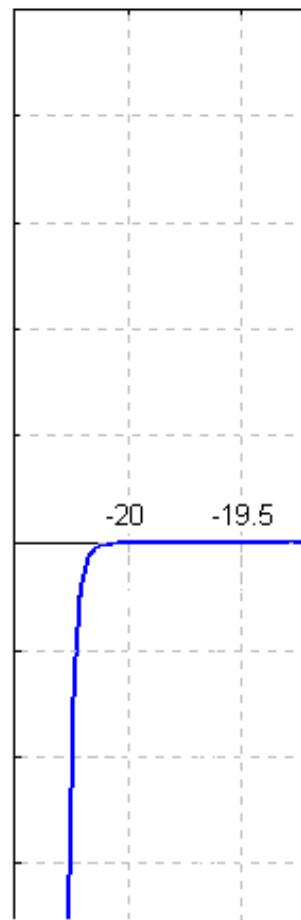
Break-down phenomena

Reason: either of

Avalanche
mechanism

Zener tunneling

Punch-through



Avalanche break-down

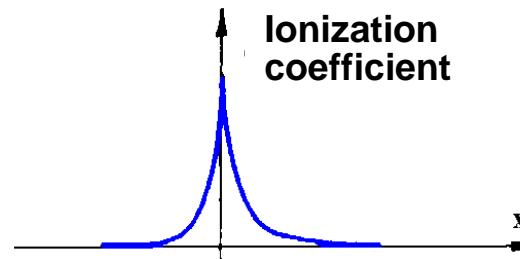
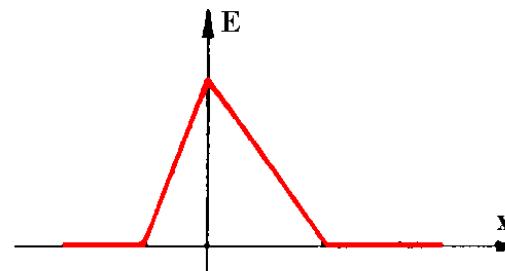
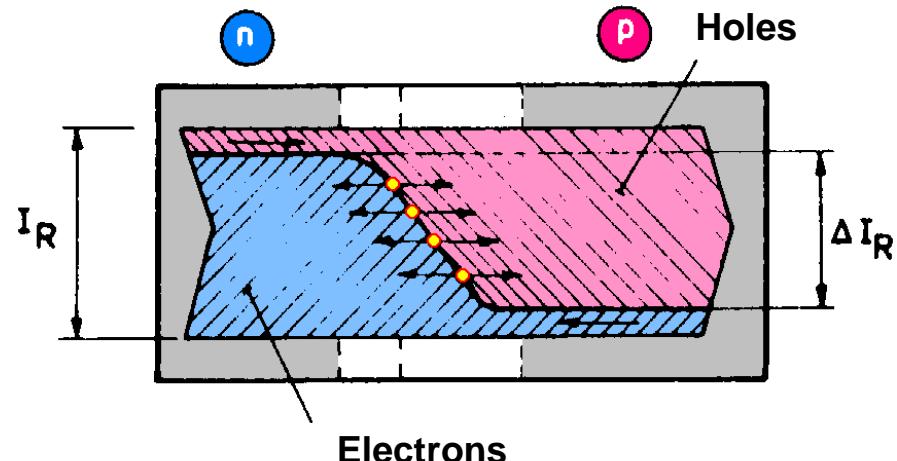
$$I_R = M(U_R) \cdot I_{R0}$$

M – multiplication factor

$$M = \frac{1}{1 - \left(\frac{-U}{U_L} \right)^m}$$

U_L depends on the less doped side:

$$U_L \sim N^{-0.7}$$

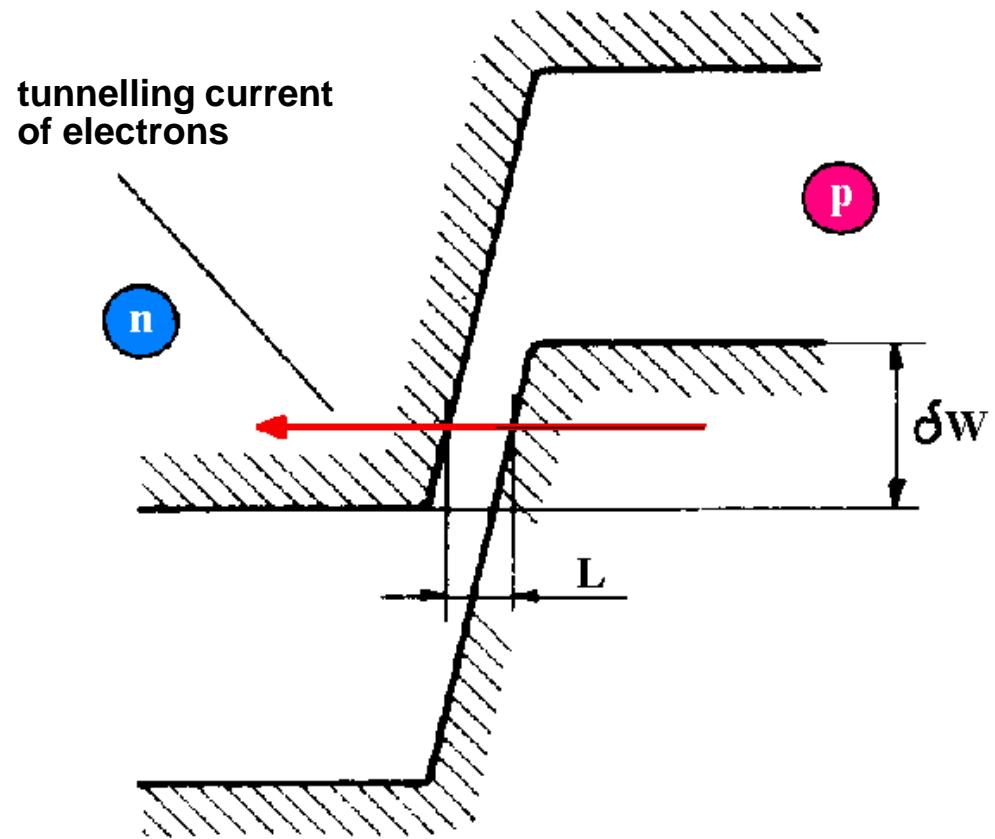


The Zener effect

Physical reason:

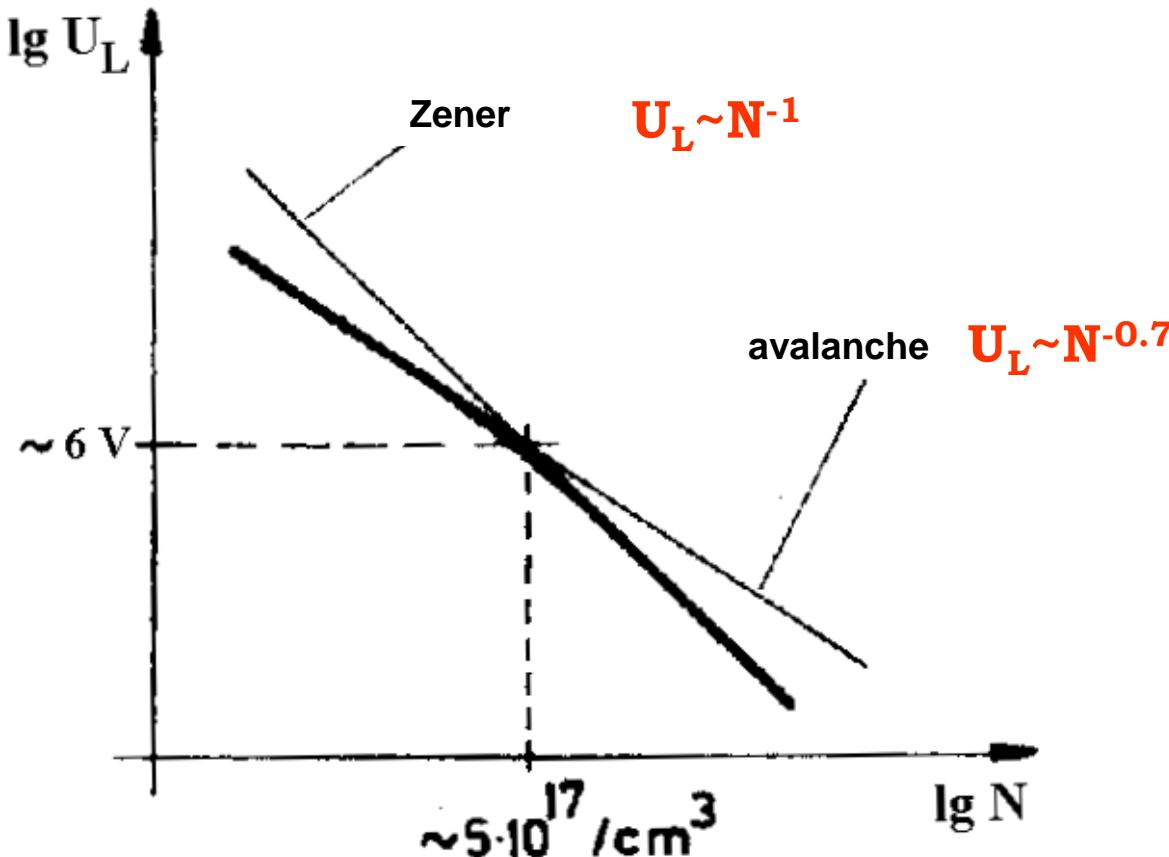
tunneling

$$U_L \sim N^{-1}$$



Break-down phenomena

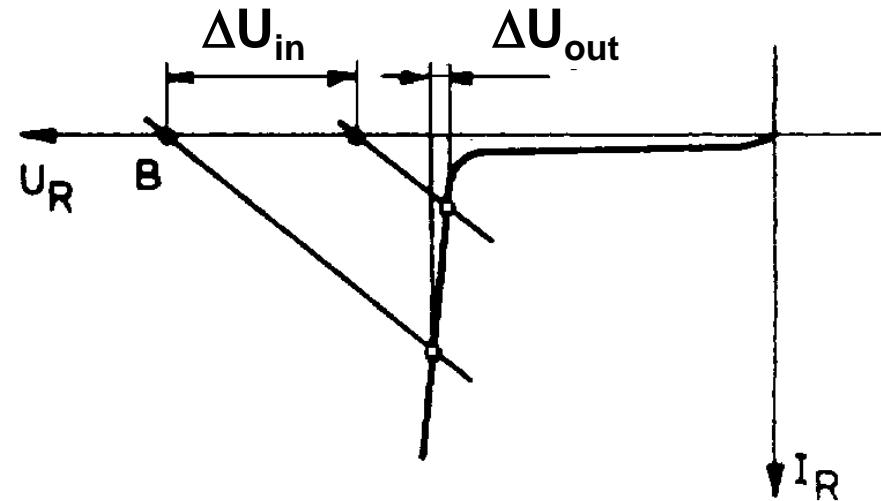
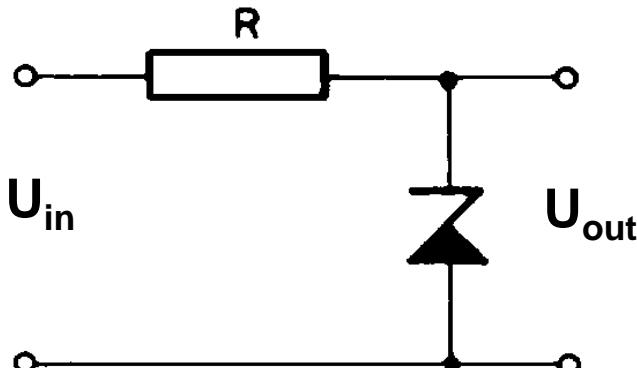
Comparison of the two phenomena



In case of Si: below 6V – Zener, above this – avalanche.

Break-down phenomena

Application: a Zener-diode



Voltage reference

Voltage stabilizer (at low power consumption)

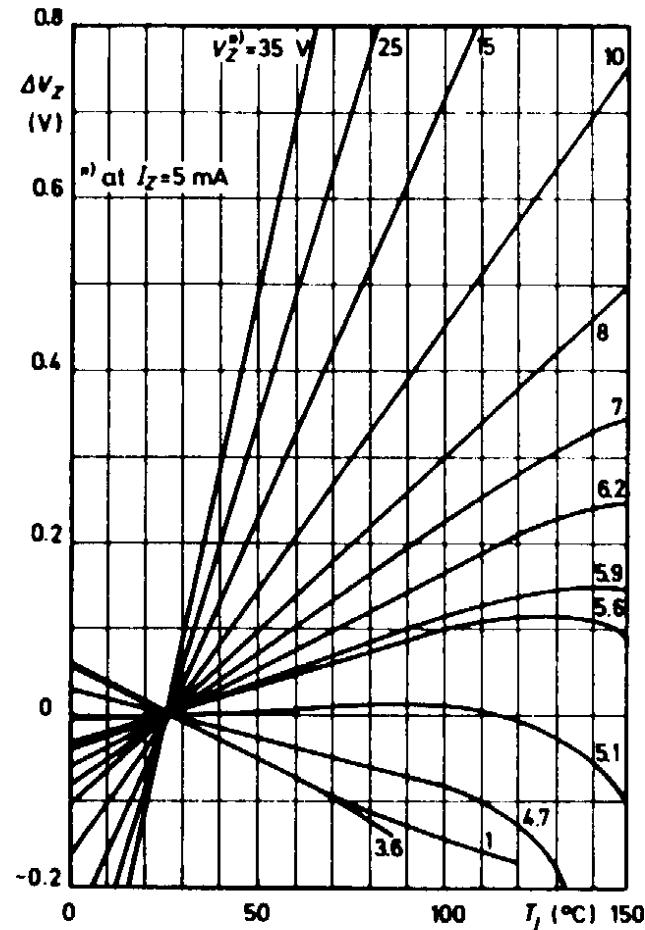
Break-down phenomena

Temperature dependence of Zener-diodes

The best: diodes around 5V

(Si diode)

Operating voltage variation versus junction temperature
 $\Delta V_Z = f(T_j); I_Z = 5 \text{ mA}$

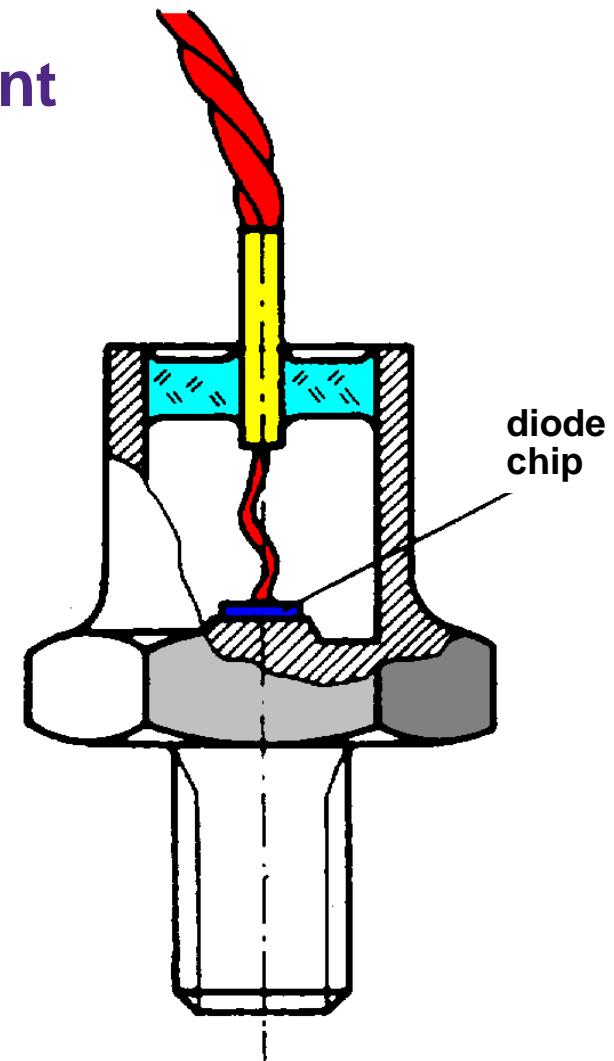


Practical issues

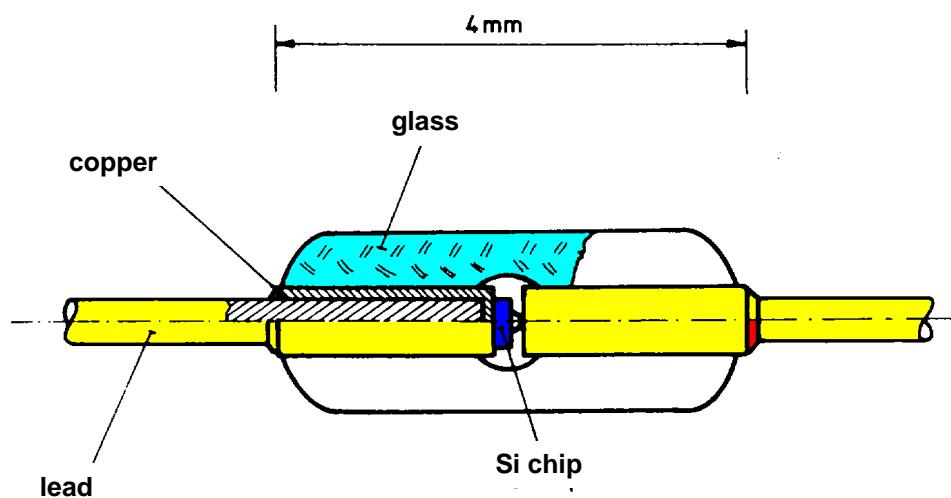
- Packaging
- Data sheets

Actual realization of diodes

Large current



Small current



(IC realizations will be discussed later)

Data sheets

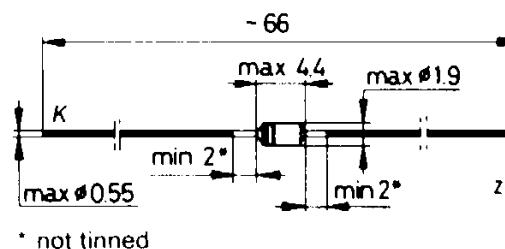
1N 4151 (BAY 95), 1N 4154 (BAY 94)

Silicon Epitaxial Planar Low-Capacitance Diodes

for very fast switching applications.

Dimensions in mm

Band: cathode



Case: DO-35

Mass: approx. 0.15 g

Absolute maximum ratings	1N 4151	1N 4154	
Reverse voltage	V_R	50	25
Peak reverse voltage	V_{RM}	75	35
Rectified current ¹	I_o	150	mA
Forward current	I_F	200	mA
Peak forward current	I_{FM}	450	mA
Surge peak forward current ²	I_{FSM}	2	A
Junction temperature	T_J	200	$^{\circ}C$
Storage temperature	T_s	-65 ... + 200	$^{\circ}C$
Total power dissipation	P_{tot}^4	440	mW
	P_{tot}^5	500	mW

Data sheets

Forward voltage $I_F = 30 \text{ mA}$	V_F^6	—	0.88 (≤ 1)	V
$I_F = 50 \text{ mA}$	V_F^6	0.88 (≤ 1)	—	V
Reverse current $V_R = 25 \text{ V}$	I_R^6	—	9 (≤ 100)	nA
$V_R = 50 \text{ V}$	I_R^6	14 (≤ 50)	—	nA
$V_R = 25 \text{ V}, T_{\text{amb}} = 150^\circ\text{C}$	I_R	—	≤ 100	μA
$V_R = 50 \text{ V}, T_{\text{amb}} = 150^\circ\text{C}$	I_R	≤ 50	—	μA
Breakdown voltage ⁶ $I_R = 5 \mu\text{A}$	$V_{(\text{BR})}$	≥ 75	≥ 35	V

Dynamic characteristics

$T_{\text{amb}} = 25^\circ\text{C}$ **1N 4151** **1N 4154**

Diode capacitance

$V_R = 0 \text{ V}, f = 1 \text{ MHz}, V_{HF} = 50 \text{ mV}$ C_D 1.7 (≤ 2) ≤ 4 pF

Reverse recovery time

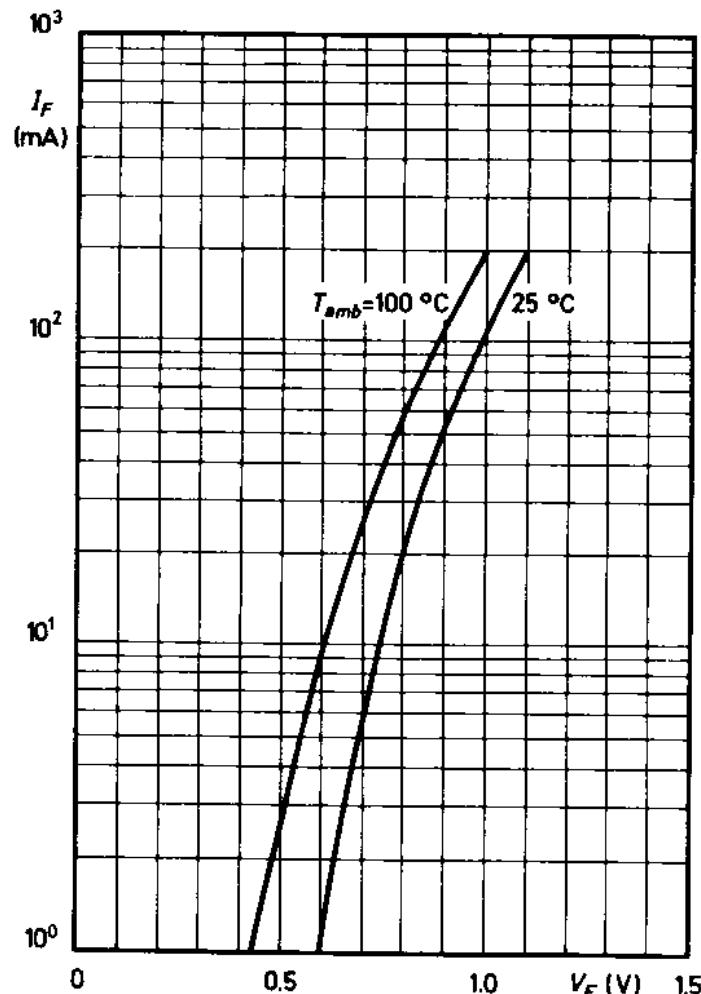
t_{rr}^1	≤ 4	≤ 4	ns
t_{rr}^2	≤ 2	≤ 2	ns

¹ measured at switching from $I_F = 10 \text{ mA}$ through $I_R = 10 \text{ mA}$ to $I_R = 1 \text{ mA}$

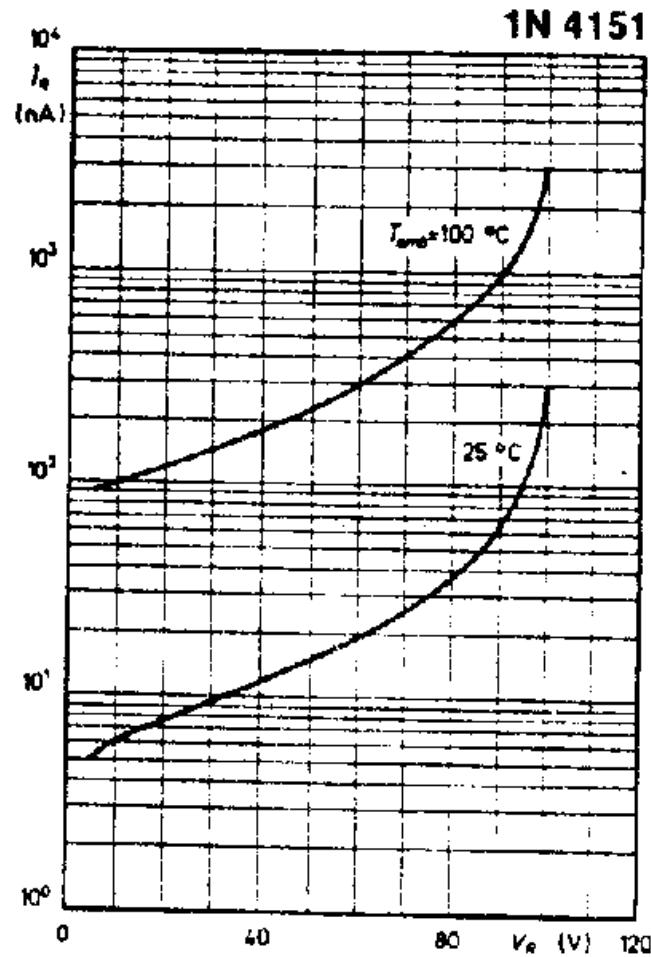
² measured at switching from $I_F = 10 \text{ mA}$ through $V_R = 6 \text{ V}$ to $I_R = 1 \text{ mA}$, R_L

Data sheets

Forward characteristics $I_F = f(V_F)$



Reverse characteristics



Data sheets

$0 \rightarrow 100 \text{ }^{\circ}\text{C}$

$6.5 \rightarrow 1200 \text{ nA}$

$$(1200/6.5)^{0.01} = 1.054$$

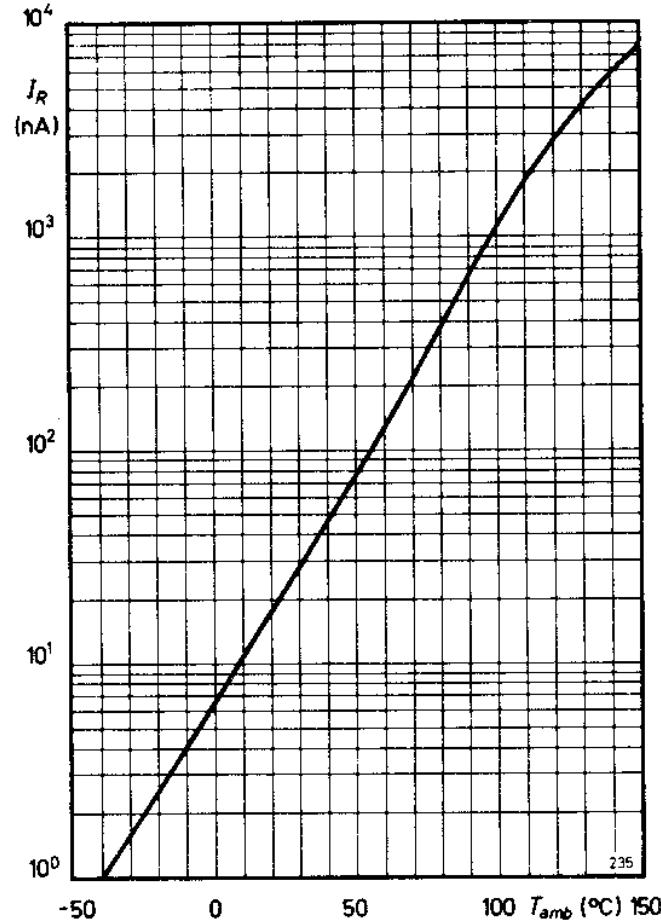
$5.4 \text{ \%}/\text{ }^{\circ}\text{C}$

**Reverse current versus
ambient temperature**

$$I_R = f(T_{\text{amb}})$$

$V_R = 50 \text{ V}: \mathbf{1N\ 4151}$

$V_R = 25 \text{ V}: \mathbf{1N\ 4154}$

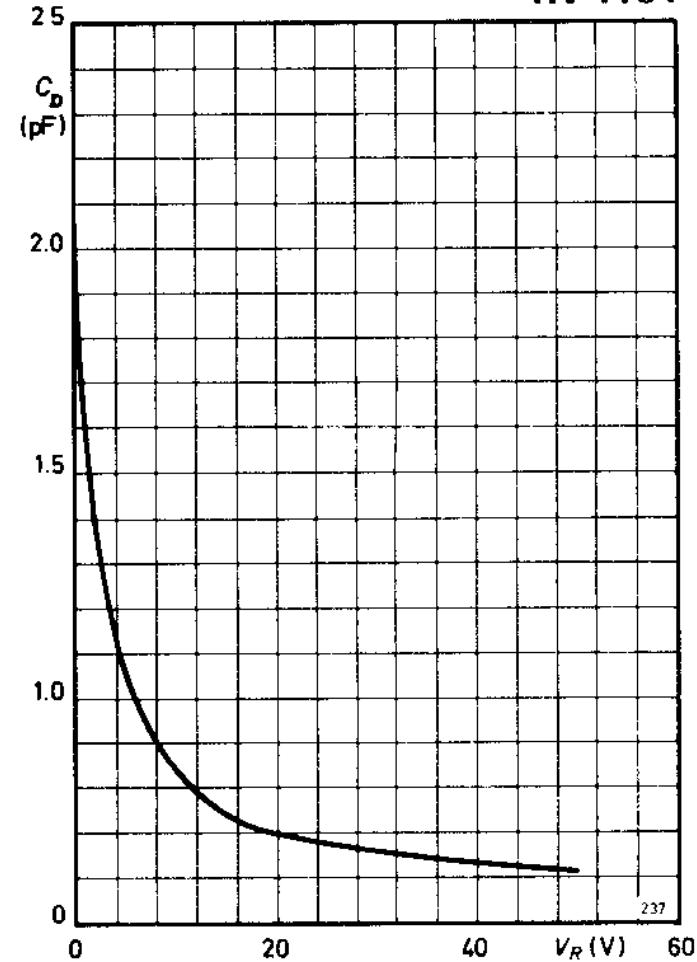


Data sheets

Diode capacitance versus
reverse voltage

$$C_D = f(V_R), f = 1 \text{ MHz}$$

1N 4151



Simulation model of the diode

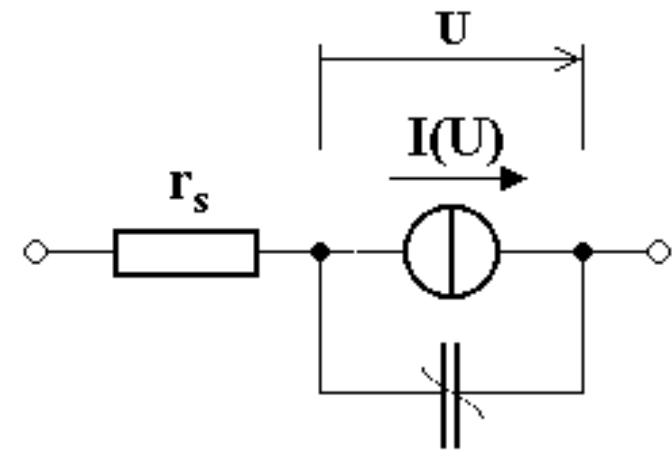
Large signal, non-linear model of the diode:

**Modelling equations are coded
in the simulation program!**

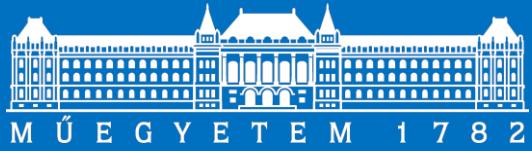
$$I = I_0 \left(\exp(U / mU_T) - 1 \right)$$

$$C_T = C_{T0} \left(\frac{U_D}{U_D - U} \right)^n$$

$$C_D = I \frac{\tau}{U_T}$$



**Modell parameter are from a
catalogue file!**

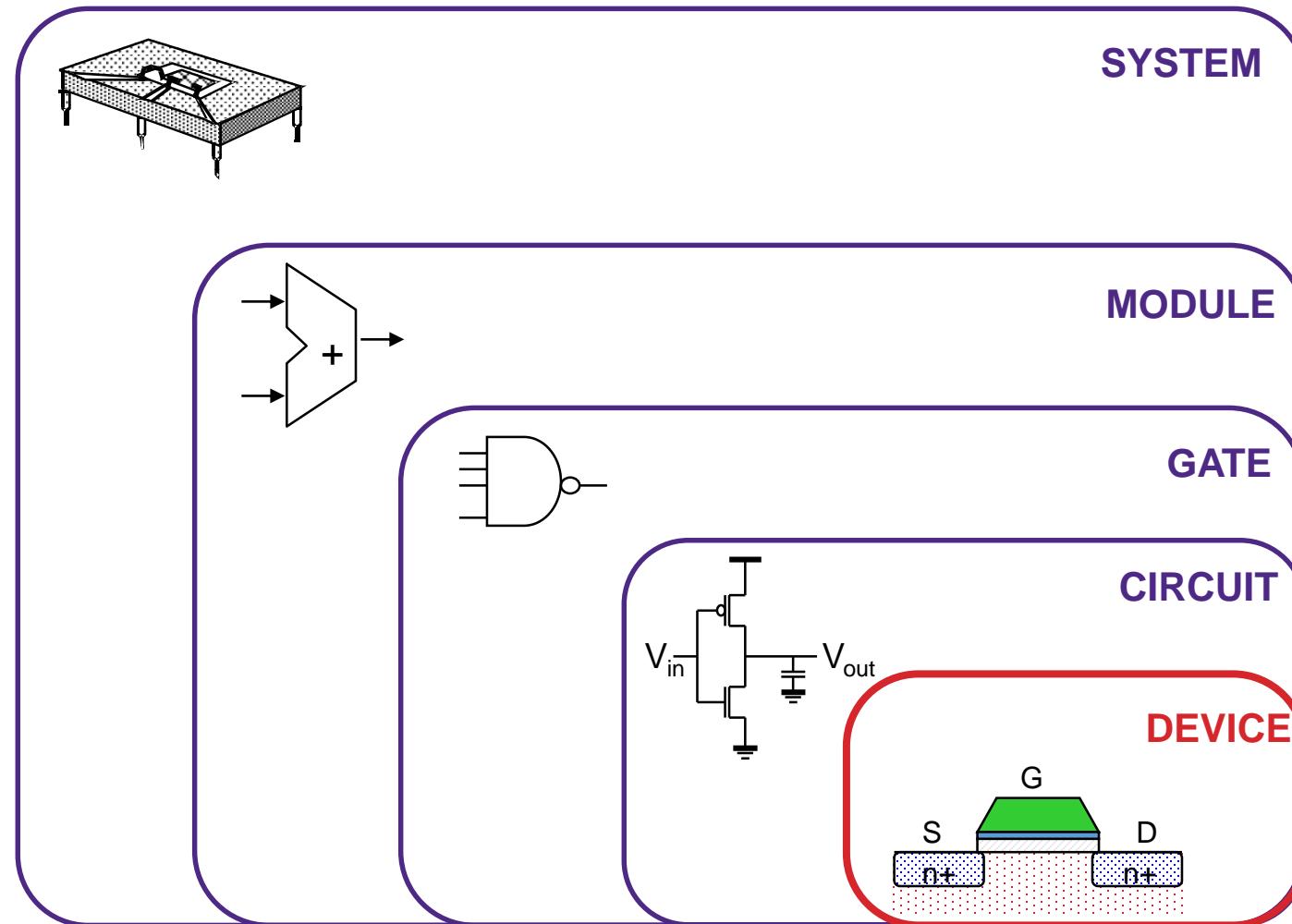


Budapest University of Technology and Economics
Department of Electron Devices

Microelectronics, BSc course

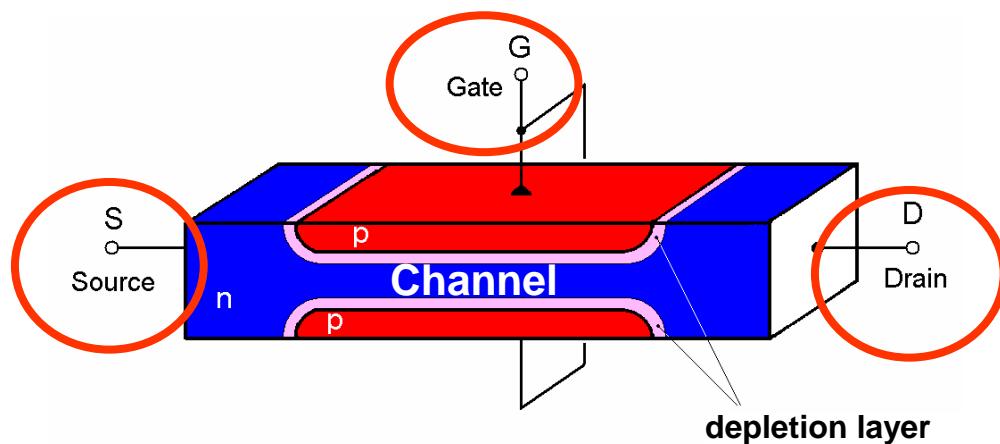
**Field effect transistors 2: The
MOSFETs**

The abstraction level of our study:



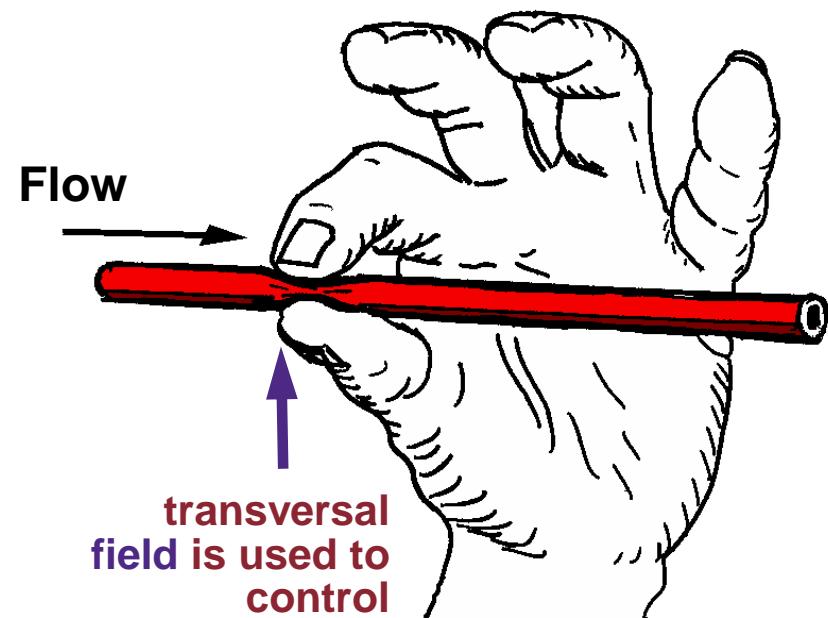
Field effect transistors 1

- FET = Field Effect Transistor – the flow of charge carriers is influenced by electric field



JUNCTION FET: depletion layers of pn-junctions close the channel

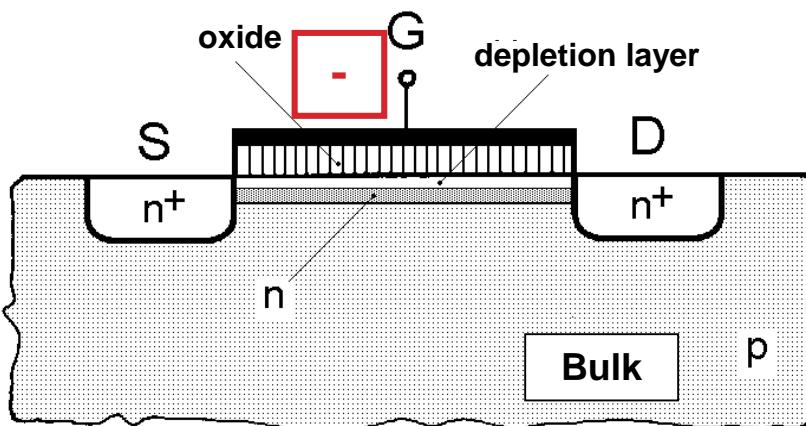
Most important parameter:
 U_0 pinch-off voltage



- Unipolar device: current is conducted by majority carriers
- Power needed for controlling the device ≈ 0

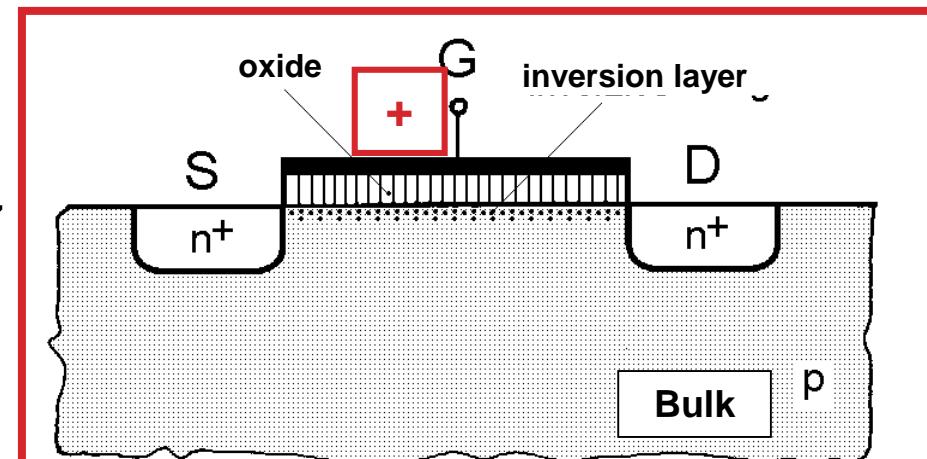
Field effect transistors 2

- MOSFET: Metal-Oxide-Semiconductor FET



First type: depletion mode device

Most important parameter:
 U_0 pinch off voltage



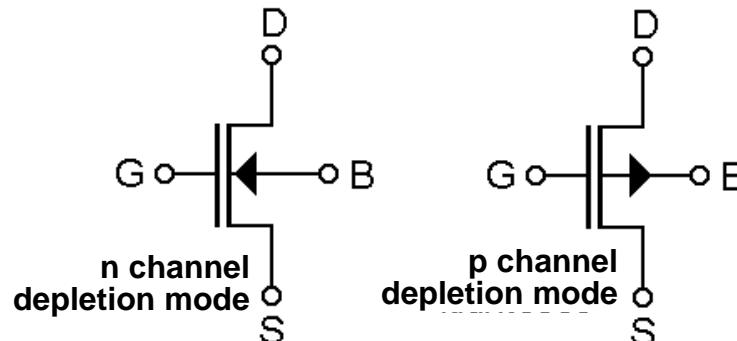
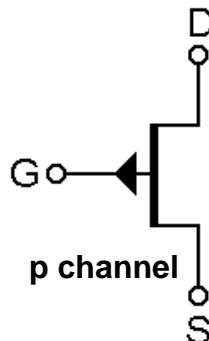
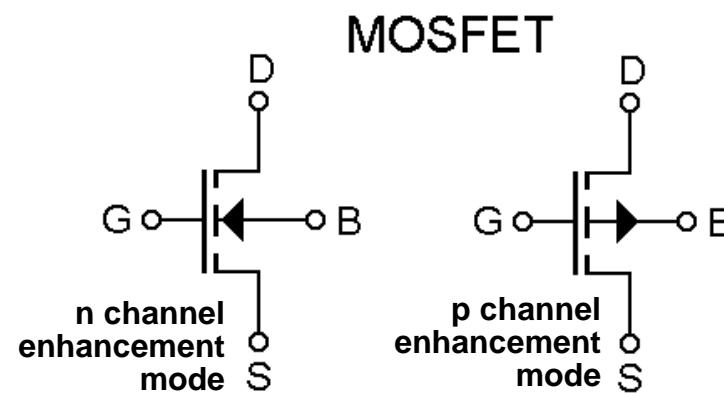
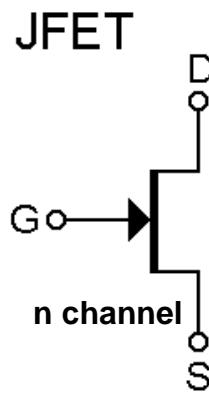
Second type: enhancement mode device

Most important parameter:
 V_T threshold voltage

Most frequently used today

Field effect transistors 3

- Symbols:



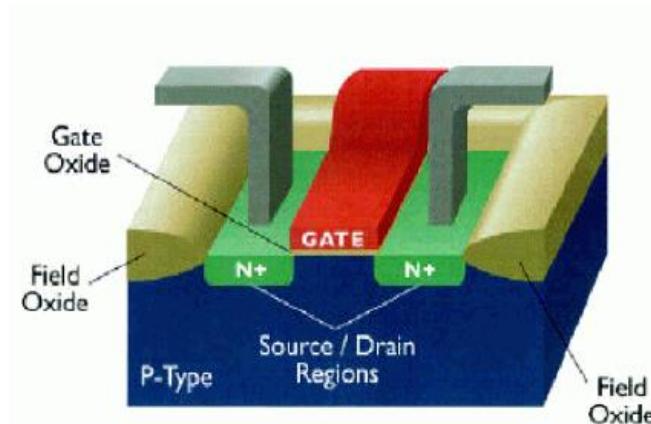
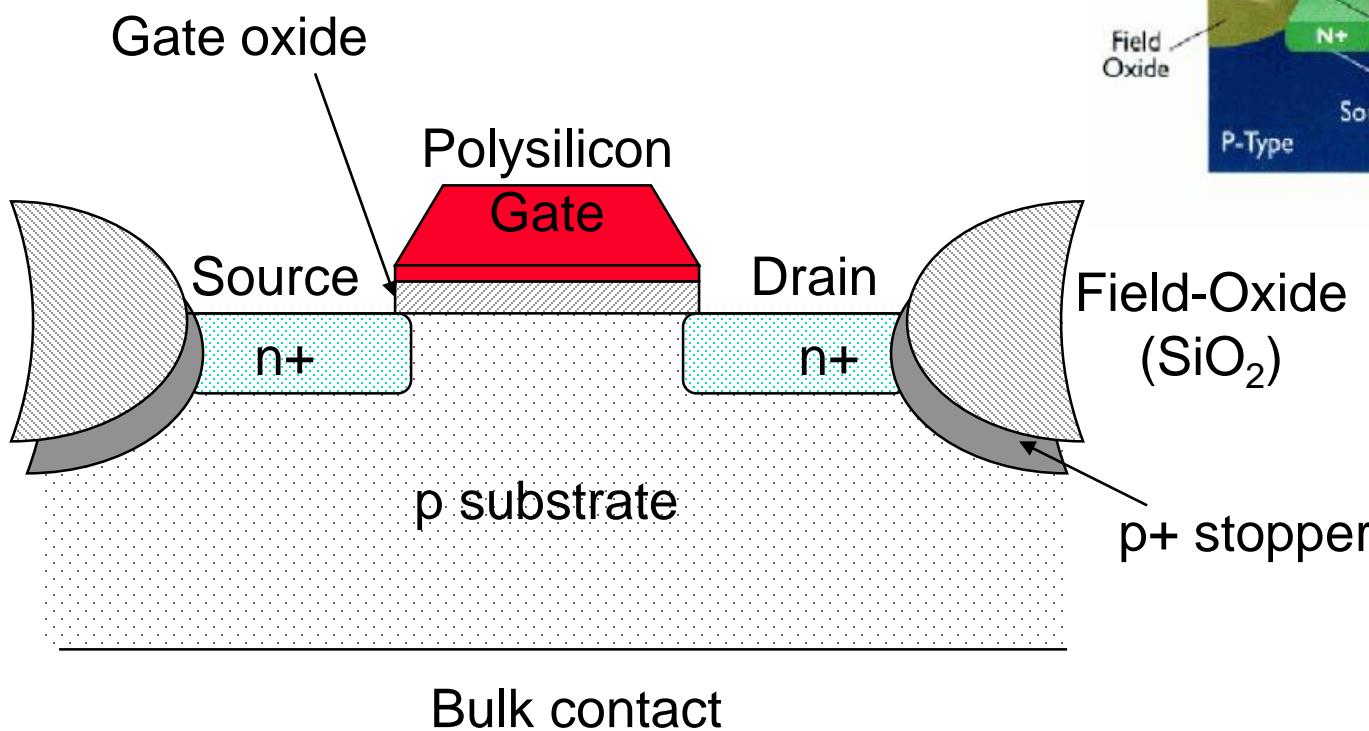
a.)

b.)

c.)

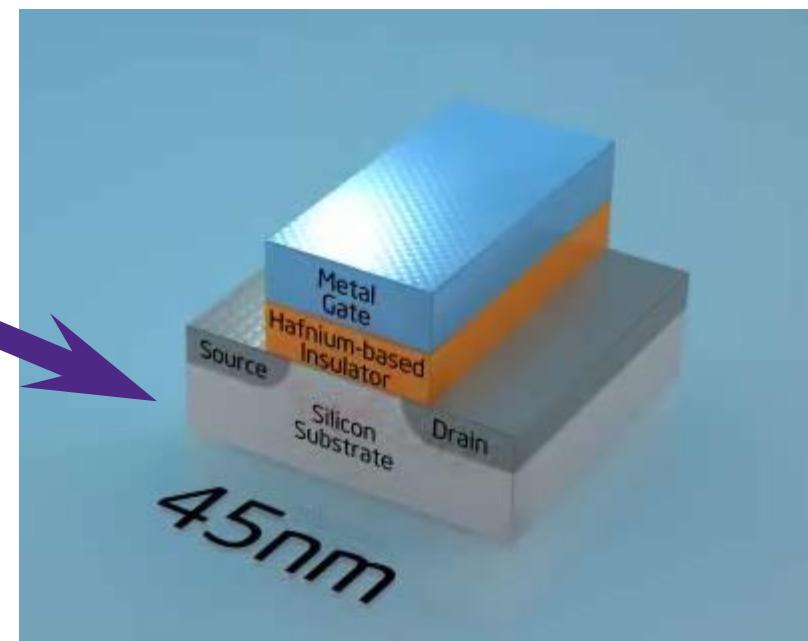
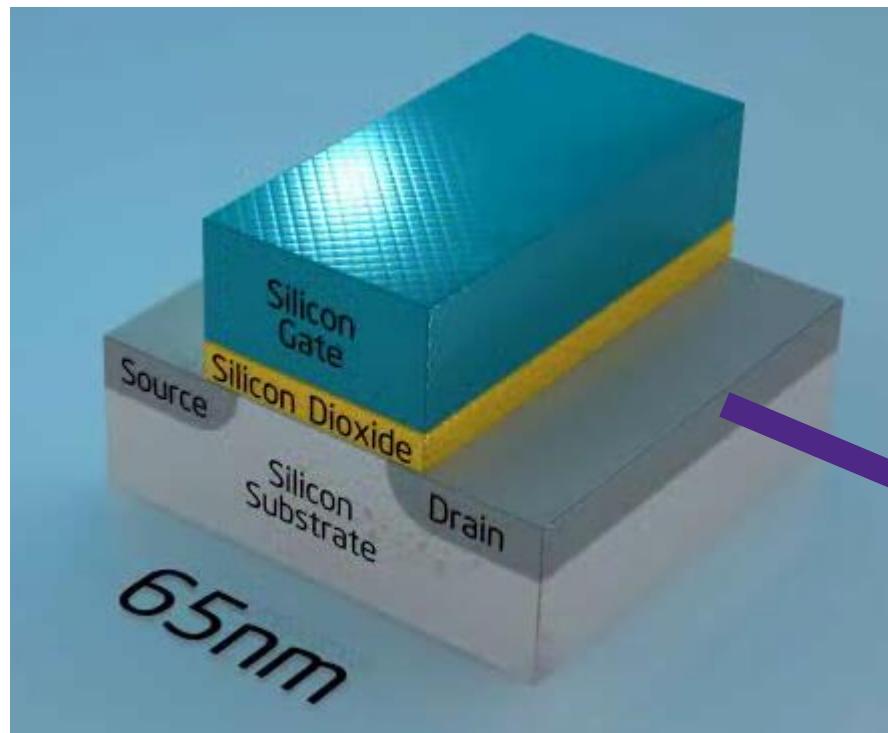
MOSFETs

- More realistic cross-sectional view of enhancement mode MOSFETs:



The most modern MOSFETs:

- 2007/2008 ... Intel:

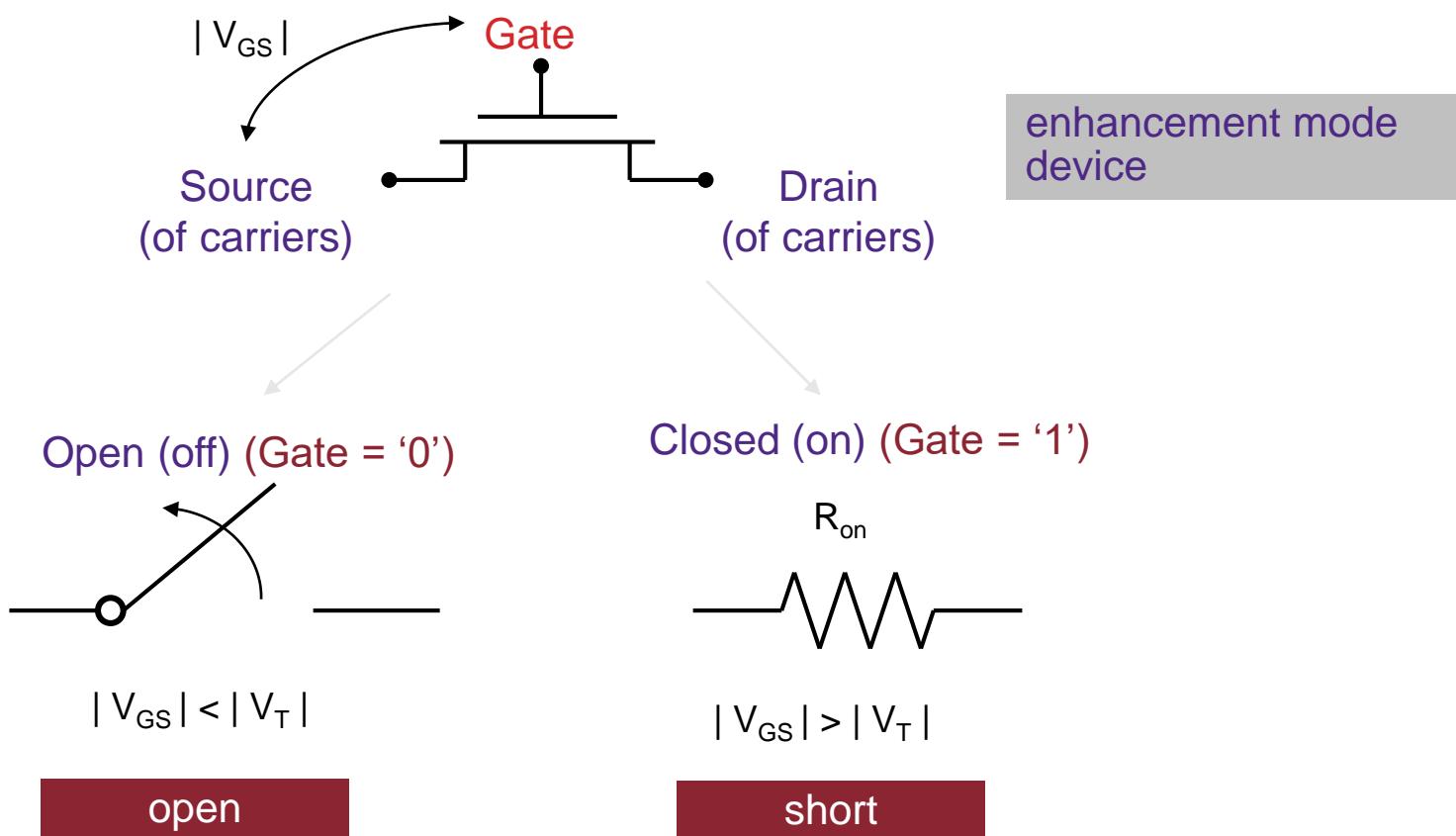


Further topics:

- Overview of operation of MOS transistors
- Characteristics
- Secondary effects
- Models

Operation of MOSFETs

- The simplest (logic) model:
 - open (off) / short (on)



Operation of MOSFETs

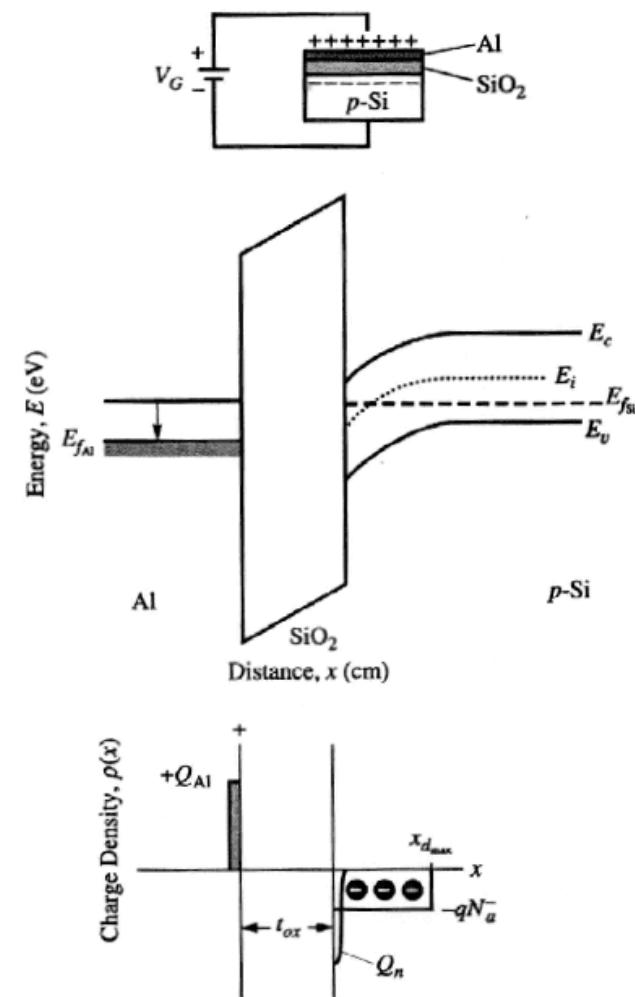
- n-channel device:
 - electrons are flowing
- p-channel device:
 - holes are flowing
 - same operation, change of the signs
- **Normally OFF device:** at 0 gate (control) voltage the are "open" (enhancement mode device)
- **Normally ON device:** at 0 gate (control) voltage the are "short" (depletion mode device)

Overview of MOSFET types

Type	Circuit Symbol	Cross Section	Output Characteristics	Transfer Characteristics
n-Channel Enhancement (Normally Off)				
n-Channel Depletion (Normally On)				
p-Channel Enhancement (Normally Off)				
p-Channel Depletion (Normally On)				

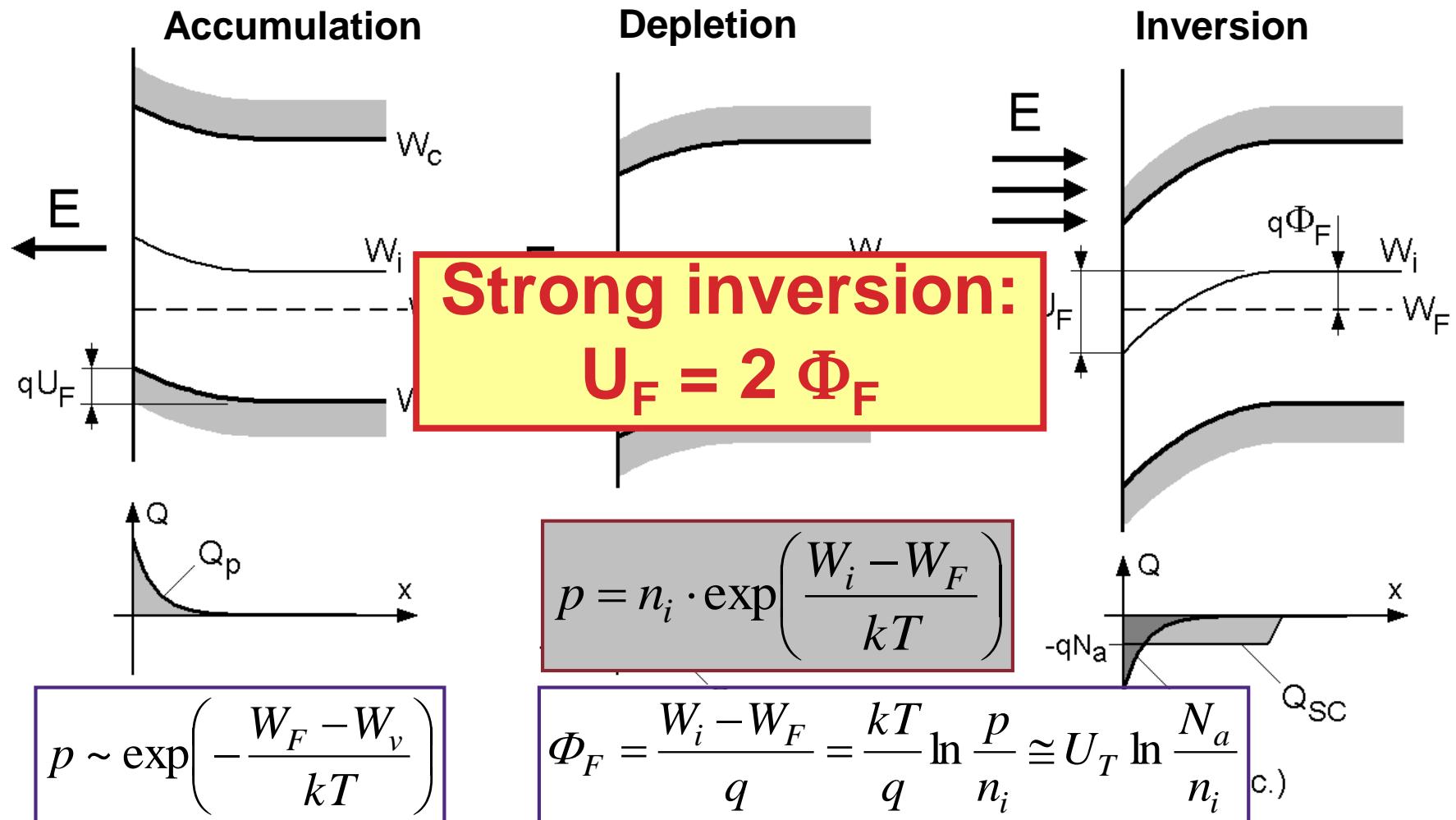
Overview of the operation

- The operation is based on the so called MOS capacitance:
- As a result of electrical field perpendicular to the gate surface
 - positive charges accumulate at the metal (gate)
 - in the p-type semiconductor
 - first the positive charges are "swept" out and a **depletion layer** is formed
 - further increasing the electric field, negative carriers are collected from the bulk under the metal
 - if the voltage at the surface exceeds a threshold value, the type of the semiconductor gets „**inverted**“: an **inversion layer** is formed
- V_T threshold voltage – the minimal voltage needed to form the inversion layer; depends on:
 - the energy levels of the semiconductor material
 - the thickness and the dielectric constant of the oxide (SiO_2)
 - the doping level and dielectric constant of the semiconductor (Si)



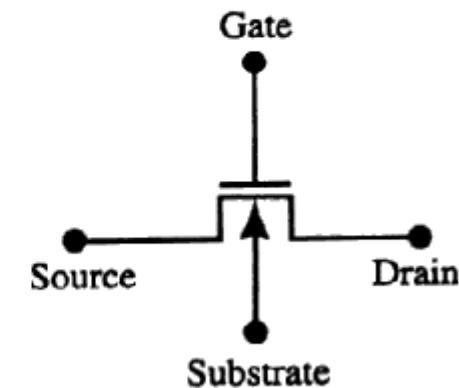
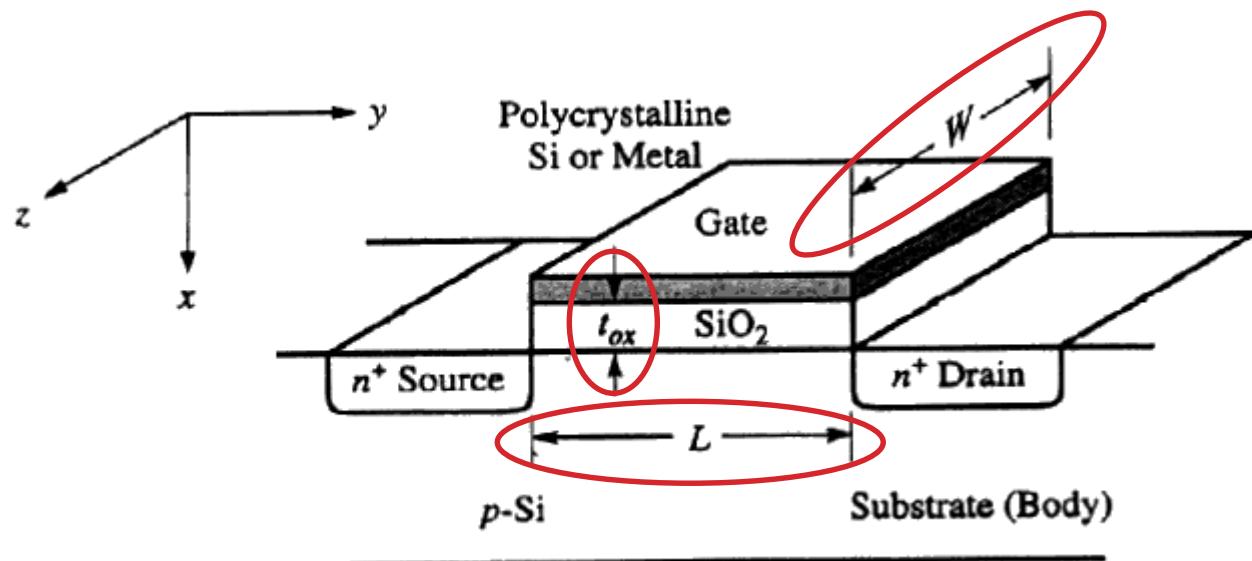
Overview of the operation

- Surface phenomena in case of the MOS capacitance



The MOS transistor

- MOS capacitance completed by two electrodes at its two sides:

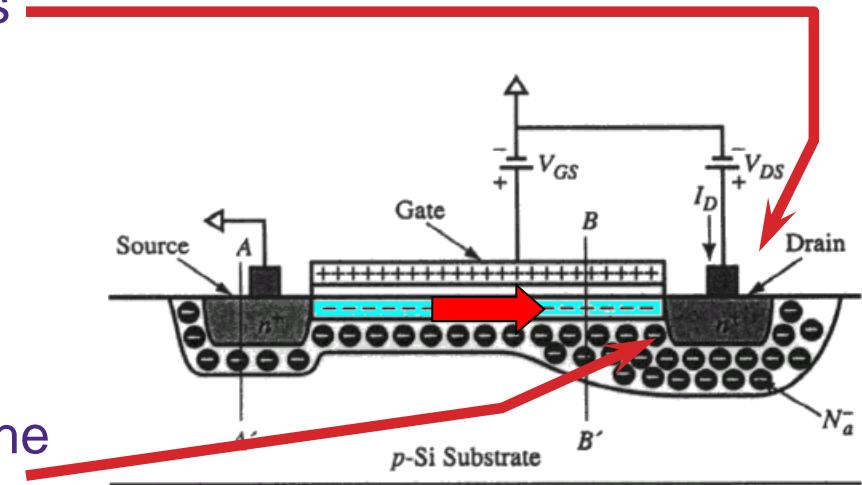
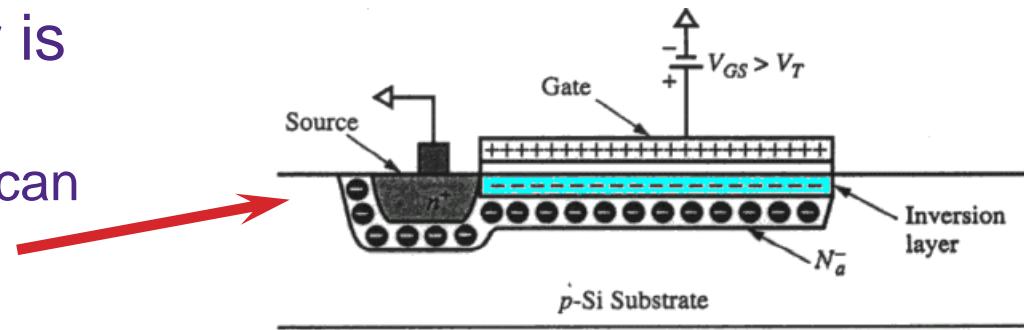


- n-channel device:** current conducted by **electrons**
- p-channel device:** current conducted by **holes**

Qualitative operation of the MOSFET

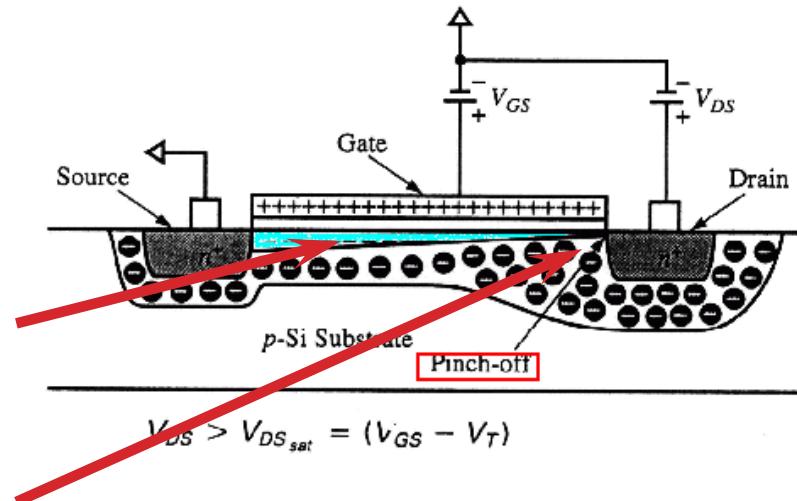
- If $V_{GS} > V_T$, inversion layer is formed

- the n+ region at the *source* can inject electrons into the inversion channel
- the positive potential at the *drain* induces flow of electrons in the channel,
- the positive potential of the drain *reverse biases* the pn junction formed there
- the electrons drifted there are all sank in the n+ region and the circuit is closed



Qualitative operation of the MOSFET

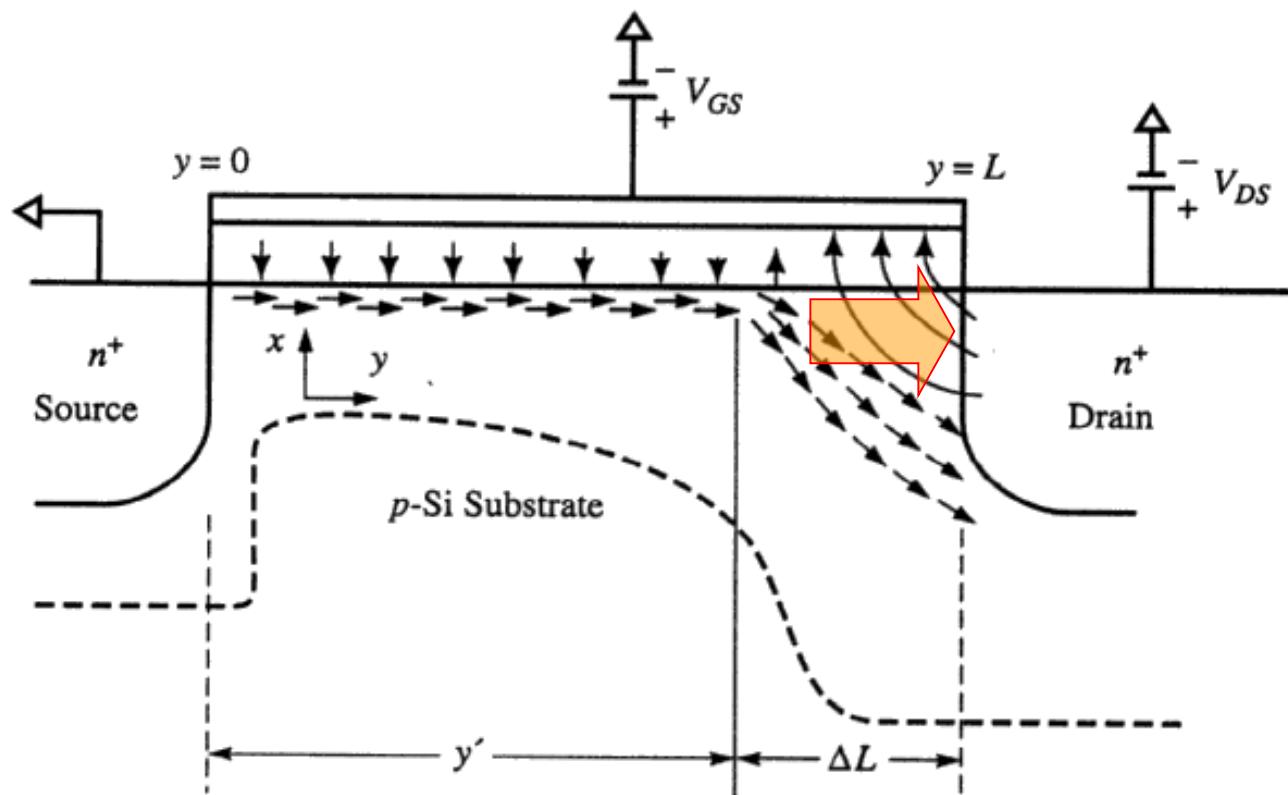
- the charge density in channel depends on the V_{GS} voltage
- there is a *voltage drop* in the channel, thus, the thickness of the inversion layer will diminish along the channel
- at a given V_{DSsat} *saturation voltage* the thickness will reach 0, this is the so called **pinch-off**



$$V_{DS_{sat}} = V_{GS} - V_T$$

After this voltage is reached, the MOSFET operates ***in saturation mode***, ***the drain voltage does not influence the drain current any longer***.

Qualitative operation of the MOSFET



In the *pinch-off* region the charge transport takes place by means of diffusion current.

I-V characteristics

- output characteristics: $I_D = f(U_{DS})$, parameter: U_{GS}
- input characteristics: $I_D = f(U_{GS})$

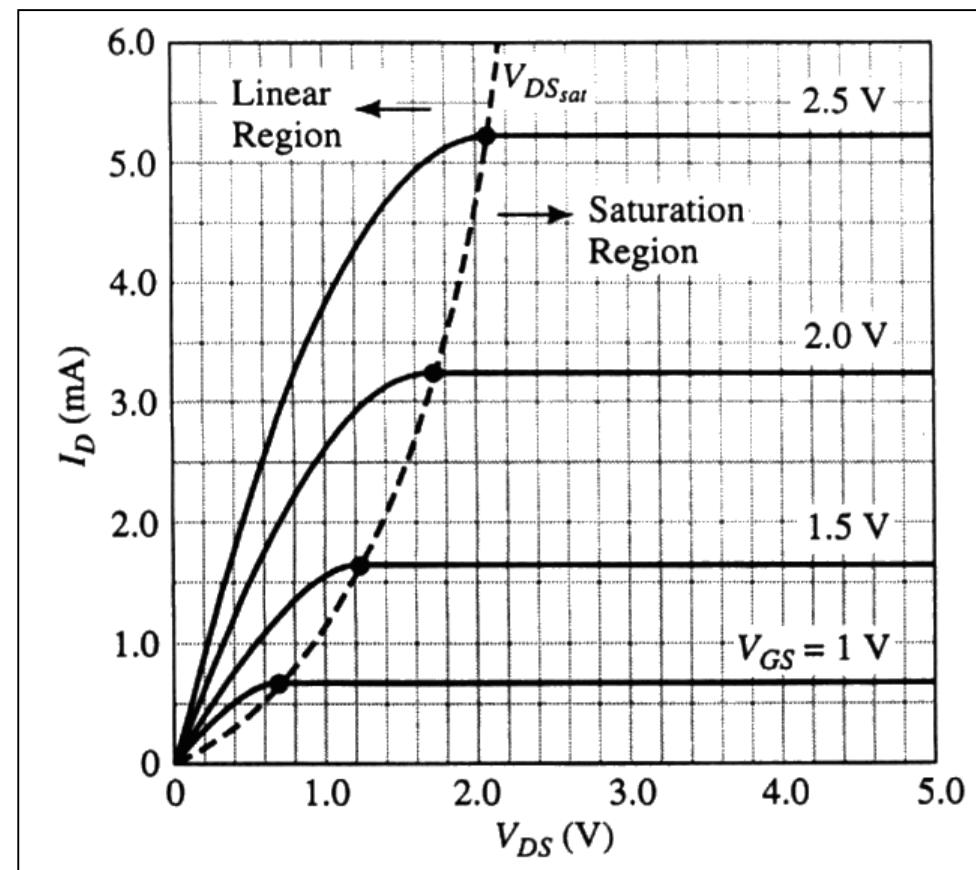
Output characteristics:

In saturation:

$$I_D = \frac{W}{L} \frac{\mu_n}{2} \frac{\epsilon_{ox}}{t_{ox}} (V_{GS} - V_T)^2$$

$$K = \frac{\mu_n \epsilon_{ox}}{t_{ox}} \quad \text{current constant}$$

The circuit designer can change the geometry only:
the **W** width and the **L** length



Example

Calculate the saturation current of a MOSFET for $U_{GS}=5V$ if

$$K = \frac{\mu_n \epsilon_{ox}}{t_{ox}} = 110 \mu A/V^2$$

$V_T = 1V$, and the geometry

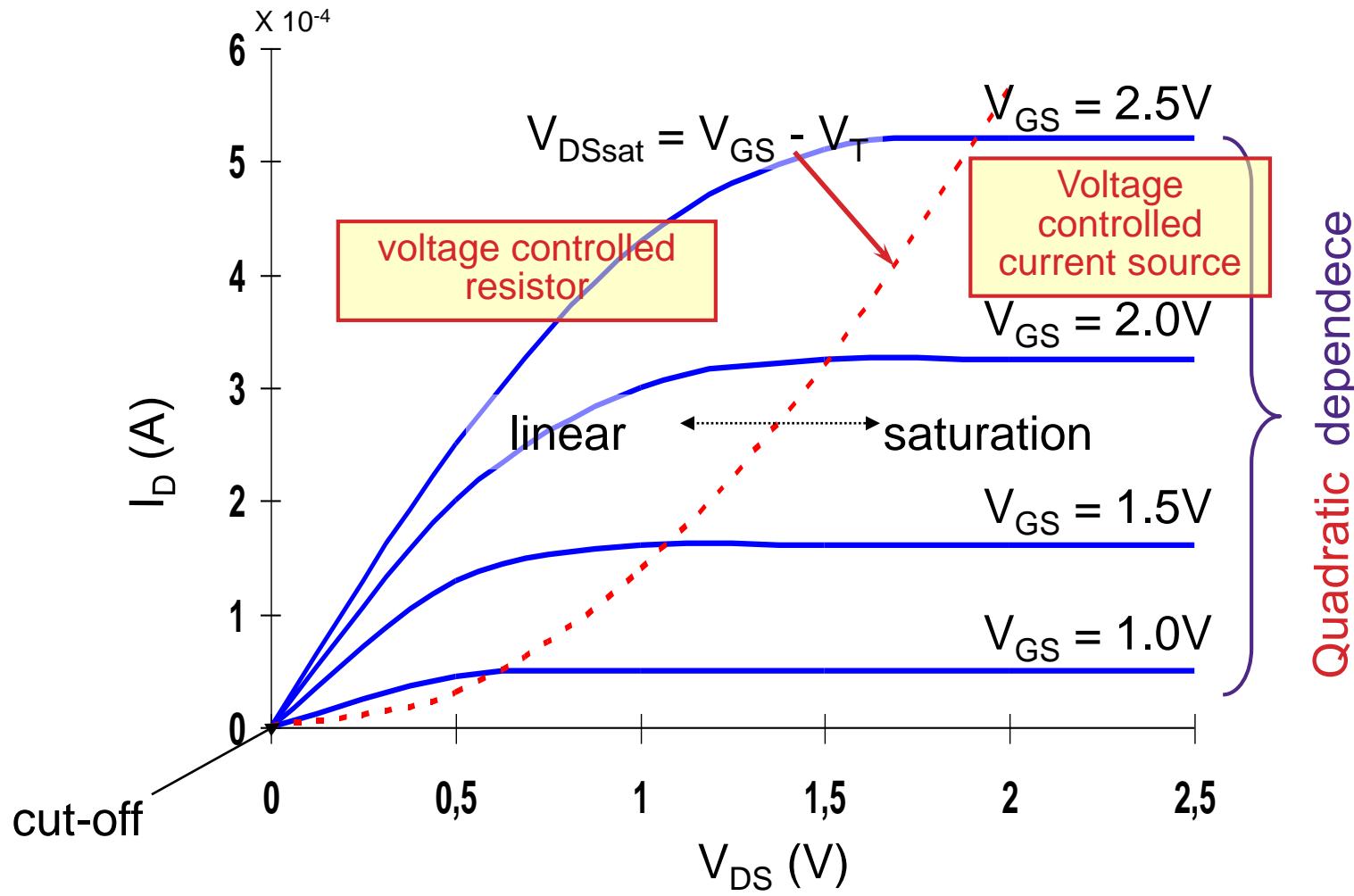
- a) $W= 5\mu m$, $L=0.4\mu m$,
 - b) $W= 0.8\mu m$, $L=5\mu m$!
-

a) $I_D = \frac{W}{L} \frac{K}{2} (U_{GS} - V_T)^2 = \frac{5}{0.4} \frac{110}{2} 10^{-6} (5 - 1)^2 = 11 \cdot 10^{-3} A = \underline{11mA}$

b) $I_D = \frac{W}{L} \frac{K}{2} (U_{GS} - V_T)^2 = \frac{0.8}{5} \frac{110}{2} 10^{-6} (5 - 1)^2 = 141 \cdot 10^{-6} A = \underline{141 \mu A}$

By changing the **W/L ratio** the drain current can be changed by orders of magnitude

I-V characteristics



nMOS transistor, $0.25\mu\text{m}$, $L_d = 10\mu\text{m}$, $W/L = 1.5$, $V_{DD} = 2.5\text{V}$, $V_T = 0.4\text{V}$

Overview of the physics:

- Charges and potentials at the surface
- The threshold voltage
- The characteristics
- Secondary effects

Potentials of the MOS structure

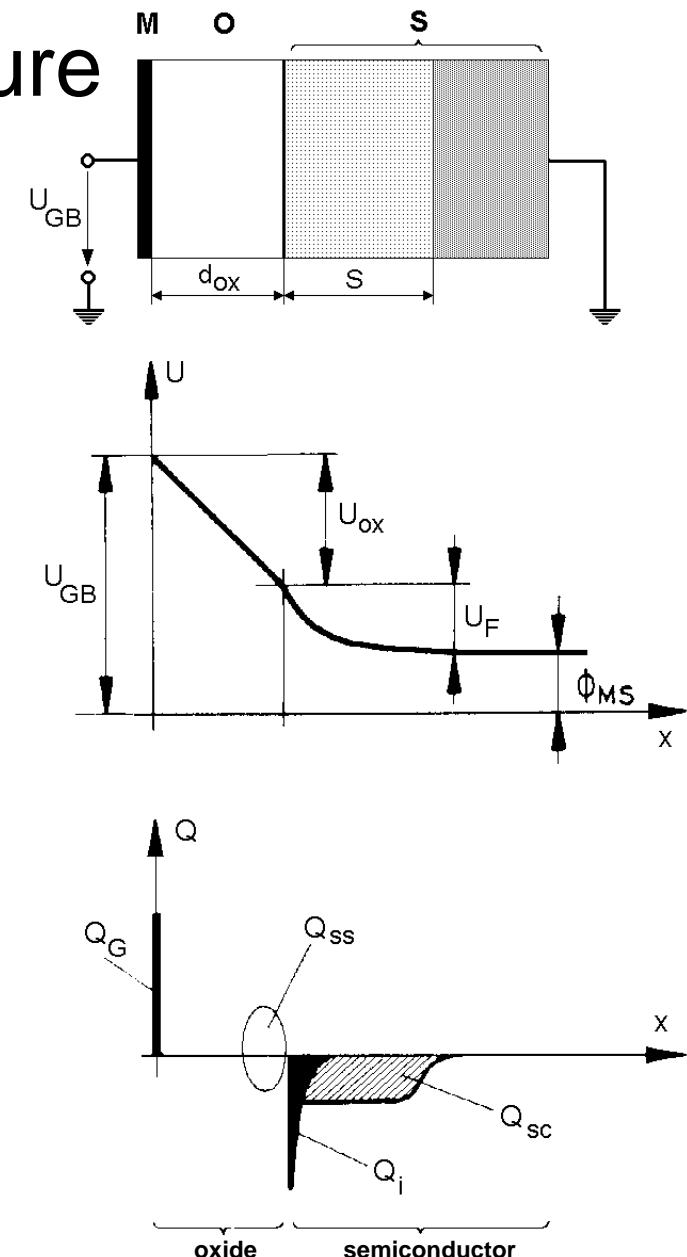
$$U_{GB} = U_{ox} + U_F + \Phi_{MS}$$

$$Q_G = Q_{SC} - Q_{SS} + Q_i$$

$$C_0 = \frac{\epsilon_{ox}}{d_{ox}}$$

$$Q_G = C_0 U_{ox}$$

$$Q_{SC} = qN_a S$$



Potentials of the MOS structure

$$U_{GB} = U_{ox} + U_F + \Phi_{MS}$$

$$Q_G = Q_{SC} - Q_{SS} + Q_i$$

$$Q_G = C_0 U_{ox}$$

$$Q_{SC} = qN_a S$$

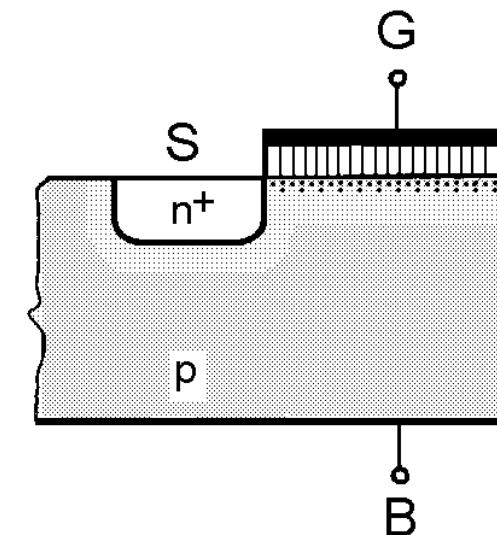
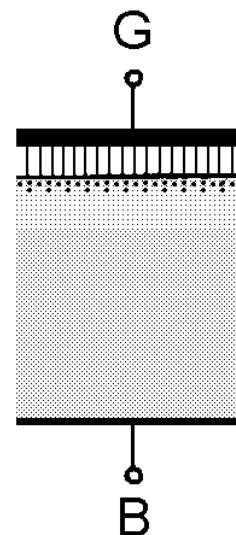
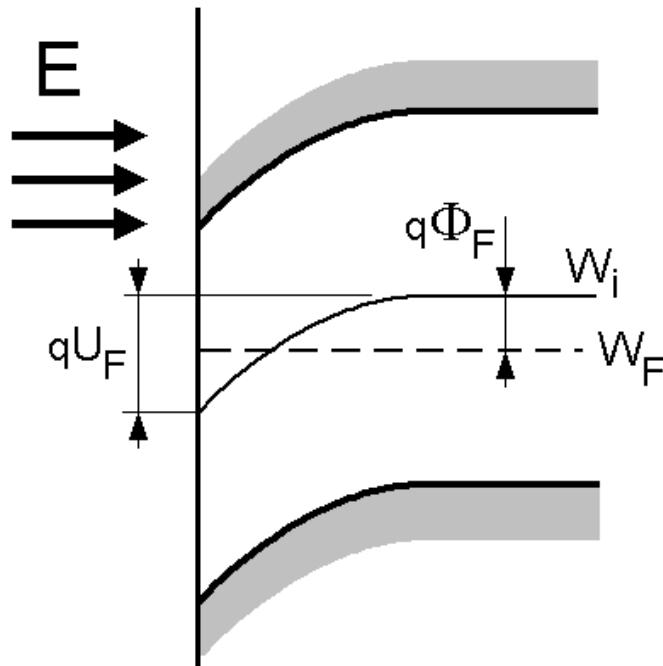
$$\begin{aligned} Q_i &= Q_G - Q_{SC} + Q_{SS} = \\ &= C_0 U_{ox} - \sqrt{2\epsilon_s q N_a} \sqrt{U_F} + Q_{SS} \end{aligned}$$

$$\begin{aligned} Q_i &= C_0 (U_{GB} - U_F - \Phi_{MS}) - \\ &\quad - \sqrt{2\epsilon_s q N_a} \sqrt{U_F} + Q_{SS} \end{aligned}$$

$$Q_{SC} = qN_a \sqrt{\frac{2\epsilon_s}{qN_a}} \sqrt{U_F} = \sqrt{2\epsilon_s q N_a} \sqrt{U_F}$$

The threshold voltage of the MOSFET

Inversion



$$U_F = 2\Phi_F$$

$$U_F = 2\Phi_F + U_{SB}$$

The threshold voltage of the MOSFET

$$Q_i = C_0(U_{GB} - 2\Phi_F - U_{SB} - \Phi_{MS}) - \sqrt{2\varepsilon_s q N_a} \sqrt{2\Phi_F + U_{SB}} + Q_{SS}$$

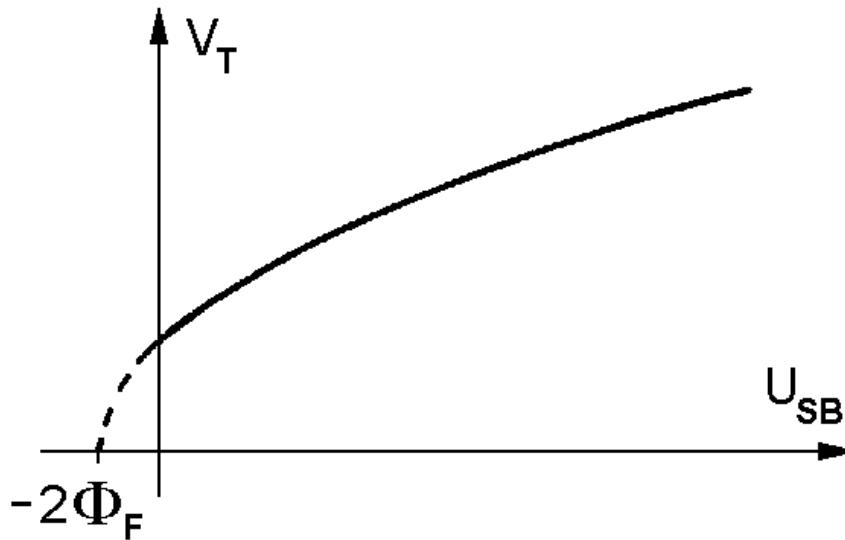
$$V_T = U_{GS} \Big|_{Q_i=0}$$

$$Q_i \cong C_0(U_{GS} - V_T)$$

$$V_T = 2\Phi_F + \Phi_{MS} - \frac{Q_{SS}}{C_0} + \frac{\sqrt{2\varepsilon_s q N_a}}{C_0} \sqrt{2\Phi_F + U_{SB}}$$

The threshold voltage of the MOSFET

$$V_T = 2\Phi_F + \Phi_{MS} - \frac{Q_{SS}}{C_0} + \frac{\sqrt{2\varepsilon_s q N_a}}{C_0} \sqrt{2\Phi_F + U_{SB}}$$



Flat-band potential:

$$\Phi_{FB} = \Phi_{MS} - \frac{Q_{SS}}{C_0}$$

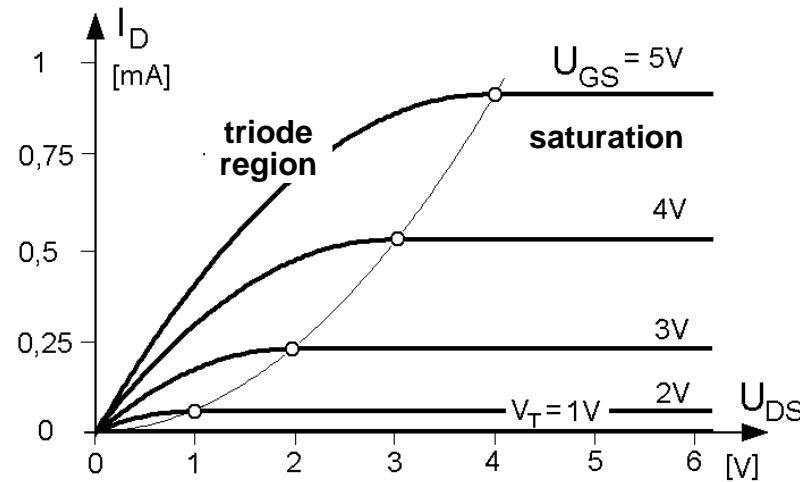
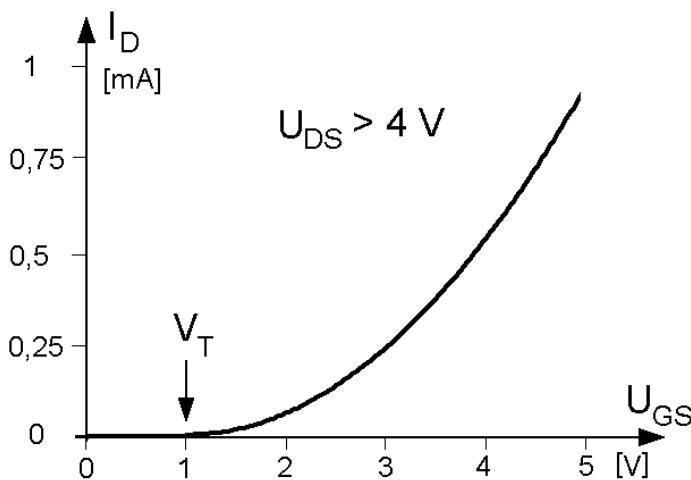
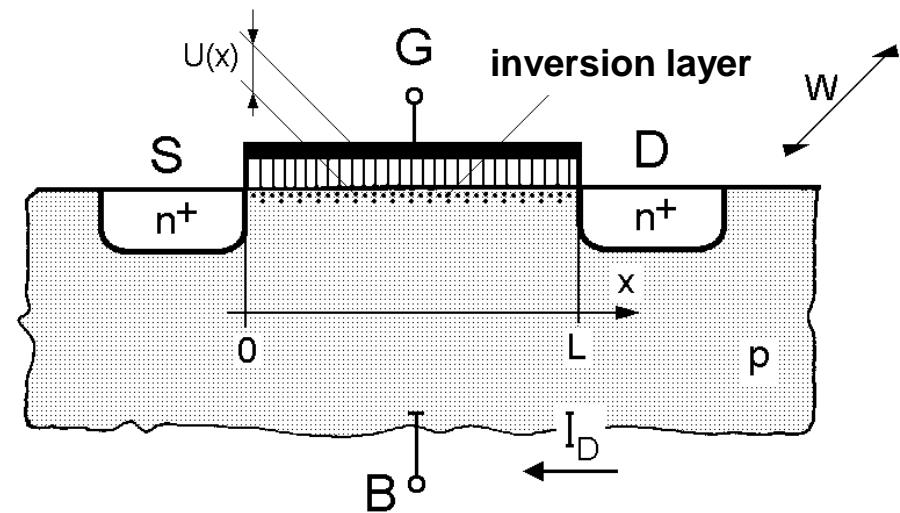
Bulk constant:

$$P = \frac{\sqrt{2\varepsilon_s q N_a}}{C_0}$$

$$V_T = 2\Phi_F + \Phi_{FB} + P \sqrt{2\Phi_F + U_{SB}}$$

The char. of an enhancement mode MOSFET

Later we shall calculate these!

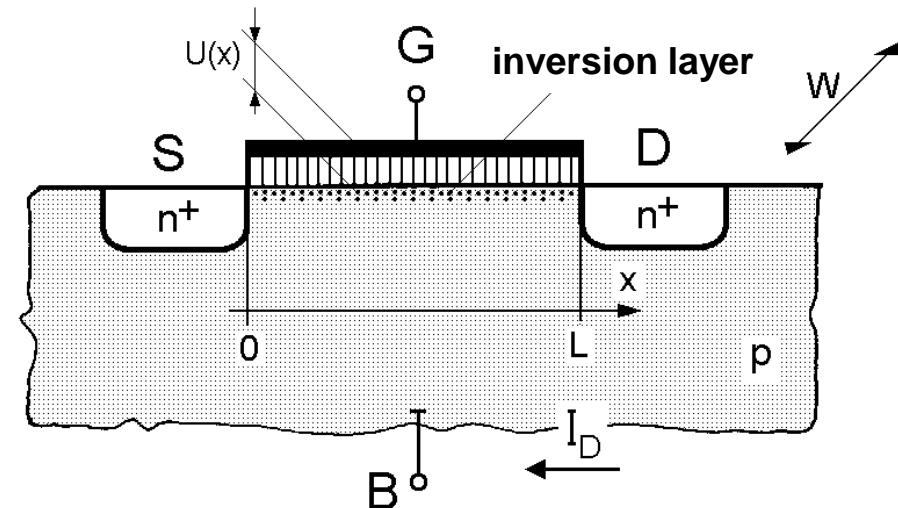


Derivation of the characteristic

$$U(0) = U_{GS}, \quad U(L) = U_{GD}$$

$$Q_i(U) = Q_i[U(x)]$$

$$I_D = Q_i W v$$



$$v = -\mu E = -\mu \frac{dU}{dx}$$

$$I_D = -Q_i(U)W\mu \frac{dU}{dx}$$

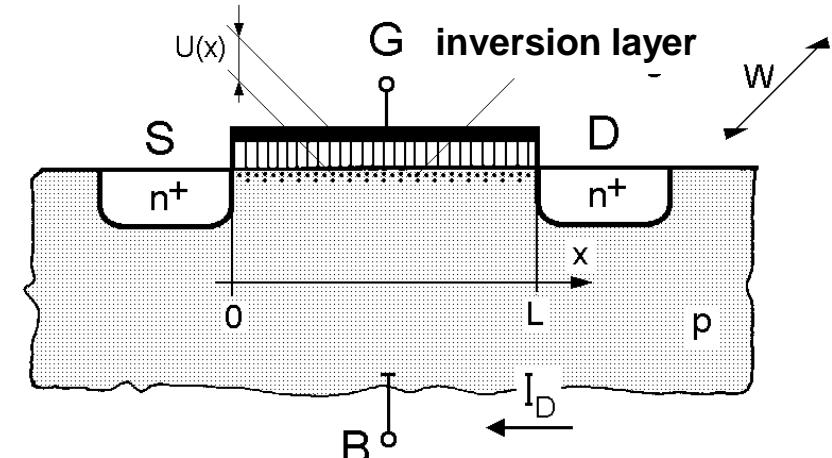
$$\int_0^L I_D dx = -W\mu \int_0^L Q_i \frac{dU}{dx} dx$$

Derivation of the characteristic

$$\int_0^L I_D dx = -W\mu \int_0^L Q_i \frac{dU}{dx} dx$$

$$I_DL = -W\mu \int_{U_{GS}}^{U_{GD}} Q_i(U) dU$$

$$Q_i = C_0(U(x) - V_T)$$

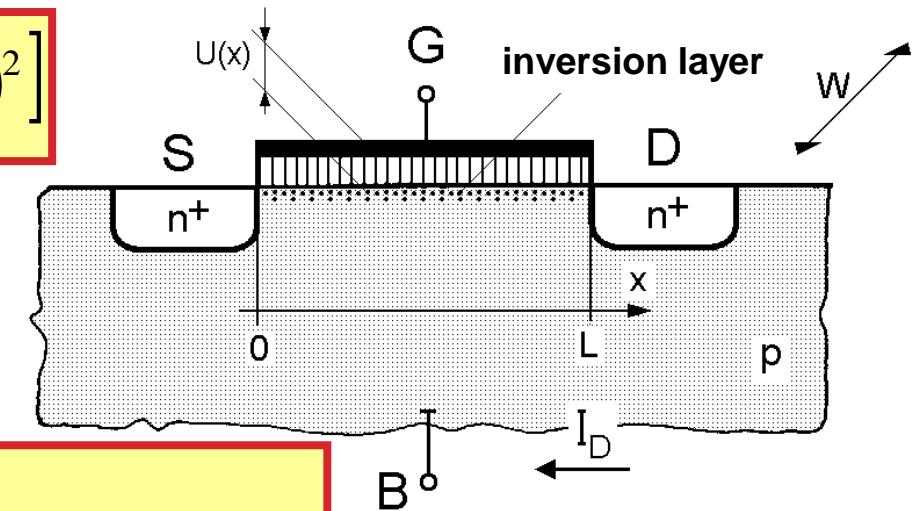


$$I_D = -\frac{W}{L}\mu \int_{U_{GS}}^{U_{GD}} C_0(U - V_T) dU = \frac{W}{L} \frac{\mu C_0}{2} (U - V_T)^2 \Big|_{U_{GS}}^{U_{GD}}$$

$$I_D = \frac{W}{L} \frac{\mu C_0}{2} \left[(U_{GS} - V_T)^2 - (U_{GD} - V_T)^2 \right]$$

Derivation of the characteristic

$$I_D = \frac{W}{L} \frac{\mu C_0}{2} [(U_{GS} - V_T)^2 - (U_{GD} - V_T)^2]$$



$$I_D = \frac{W}{L} \frac{\mu C_0}{2} [F(U_{GS}) - F(U_{GD})]$$

$$F(U) = \begin{cases} (U - V_T)^2 & \text{if } U > V_T \\ 0 & \text{if } U \leq V_T \end{cases}$$

For all regions of operation!

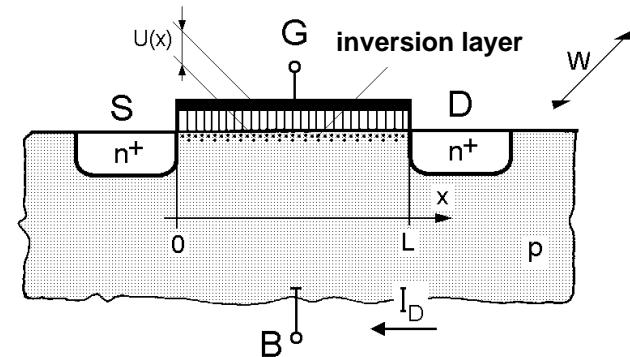
The saturation region

$$I_D = \frac{W}{L} \frac{\mu C_0}{2} [F(U_{GS}) - F(U_{GD})]$$

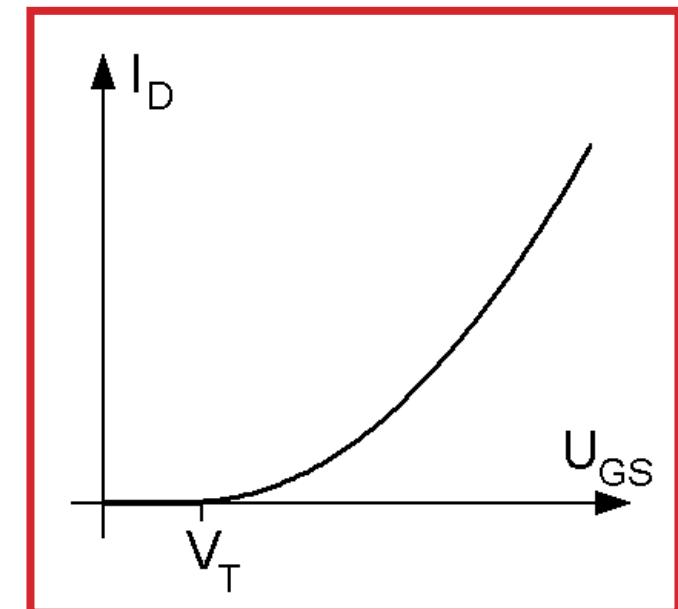
$$F(U) = \begin{cases} (U - V_T)^2 & \text{ha } U > V_T \\ 0 & \text{ha } U \leq V_T \end{cases}$$

For all regions of operation!

$$I_D = \frac{W}{L} \frac{\mu C_0}{2} (U_{GS} - V_T)^2$$



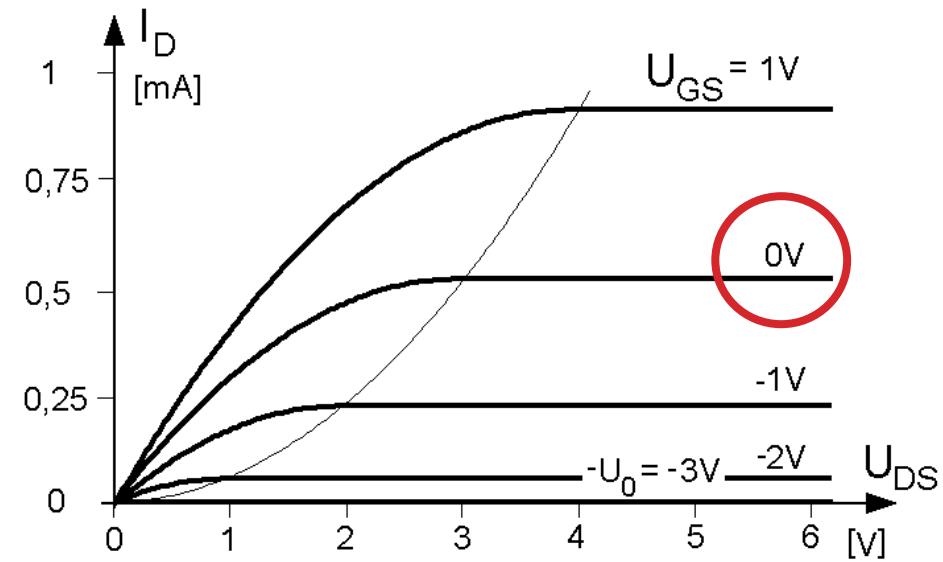
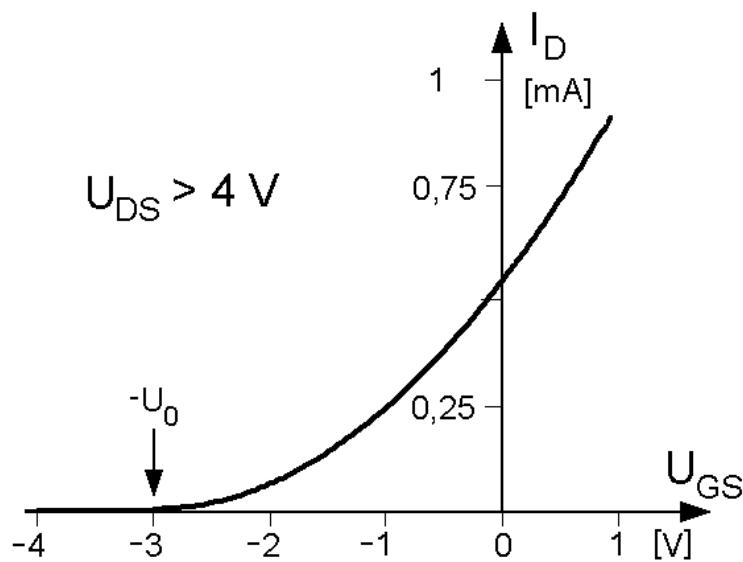
Saturation: $U_{GD} < V_T$



Overview of all types of MOSFETs

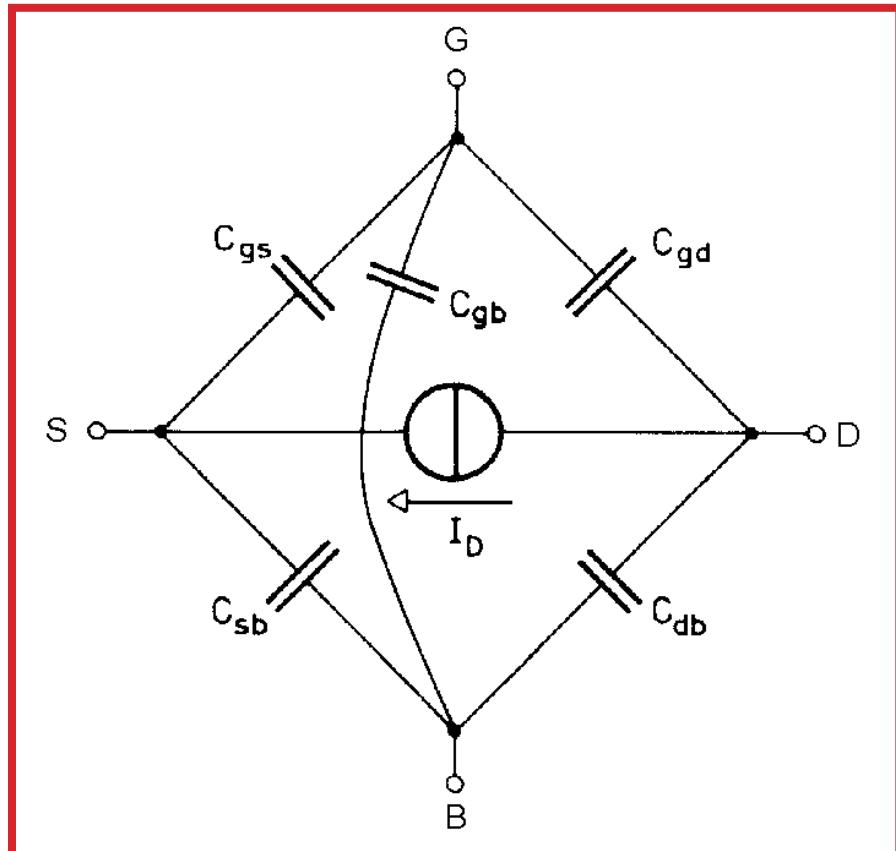
Type	Circuit Symbol	Cross Section	Output Characteristics	Transfer Characteristics
n-Channel Enhancement (Normally Off)				
n-Channel Depletion (Normally On)				
p-Channel Enhancement (Normally Off)				
p-Channel Depletion (Normally On)				

Depletion mode MOSFET

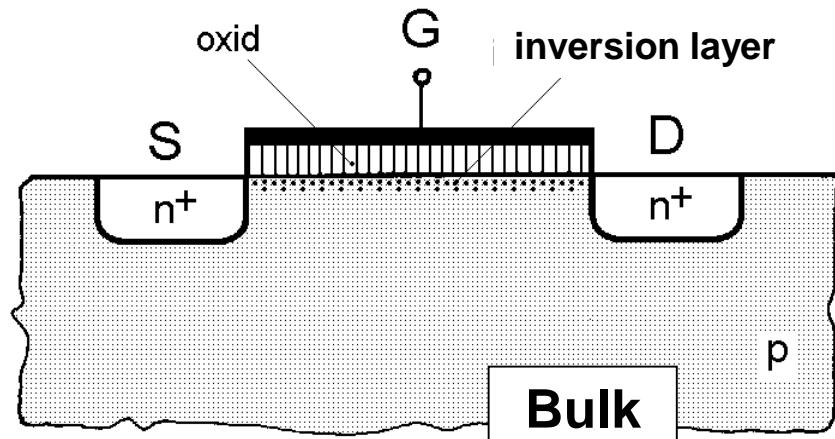


Like an enhance mode MOSFET with a negative threshold voltage

Capacitances of the MOSFET



S/D – B capacitance: reverse biased PN junction

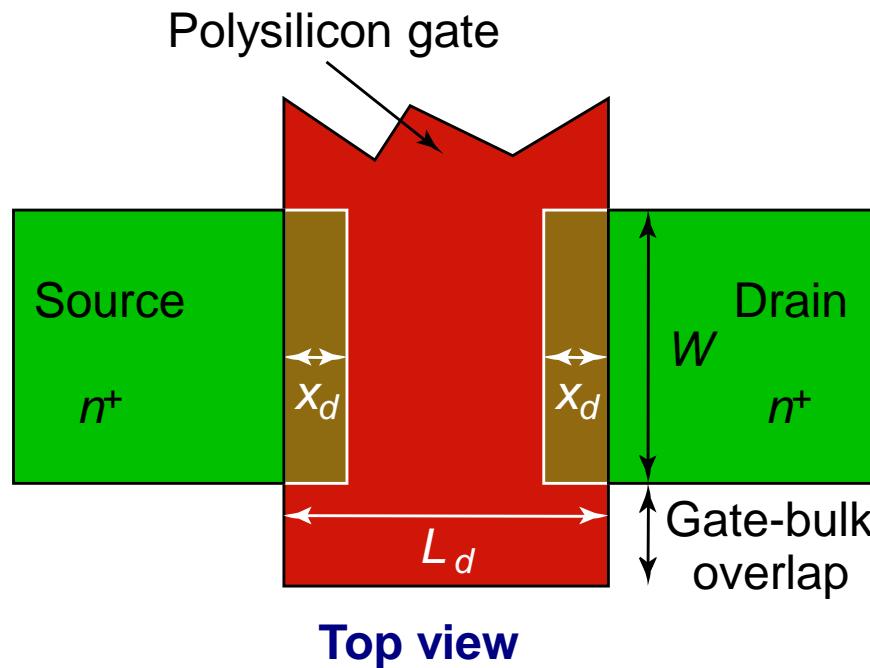


$$Q_G = f_G(U_{GS}, U_{GD}, U_{GB})$$

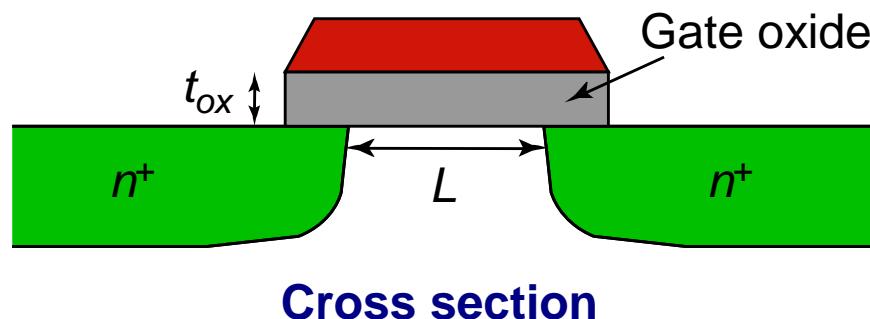
$$Q_i = f_i(U_{GS}, U_{GD}, U_{GB})$$

$$C_{gs} = \frac{\partial Q_G}{\partial U_{GS}}$$

The gate capacitance:



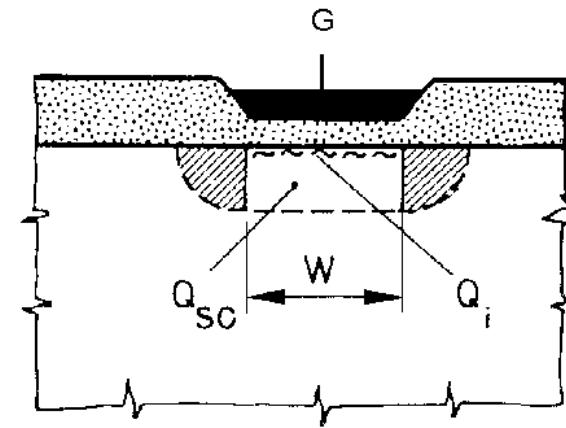
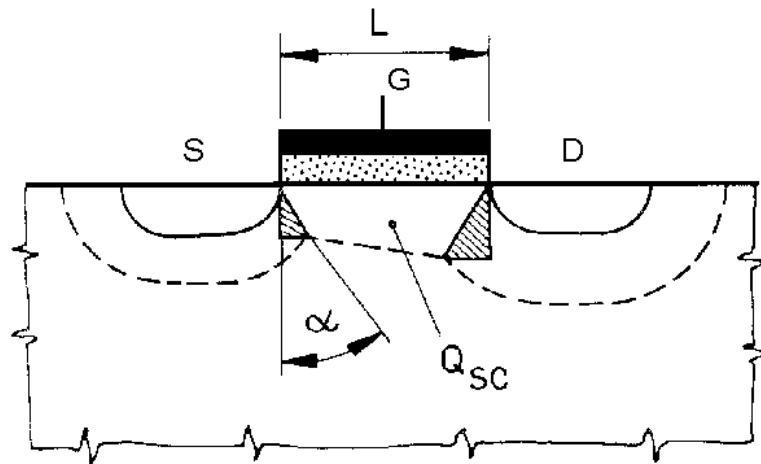
$$C_{gate} = \frac{\epsilon_{ox}}{t_{ox}} WL$$



Secondary effects

- Short and narrow-channel effects
- Velocity saturation
- Channel length modulation
- Temperature dependence
- Subthreshold current

Dependence of threshold voltage on geometry

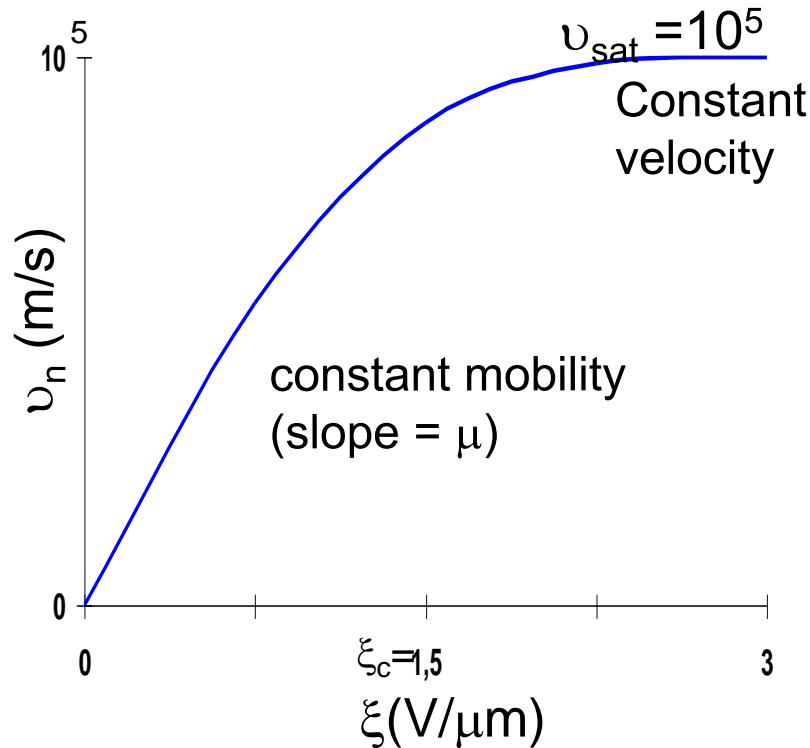


Short channel: V_T decreases

Narrow channel: V_T increases

Velocity saturation

- Influences the operation of short channel devices

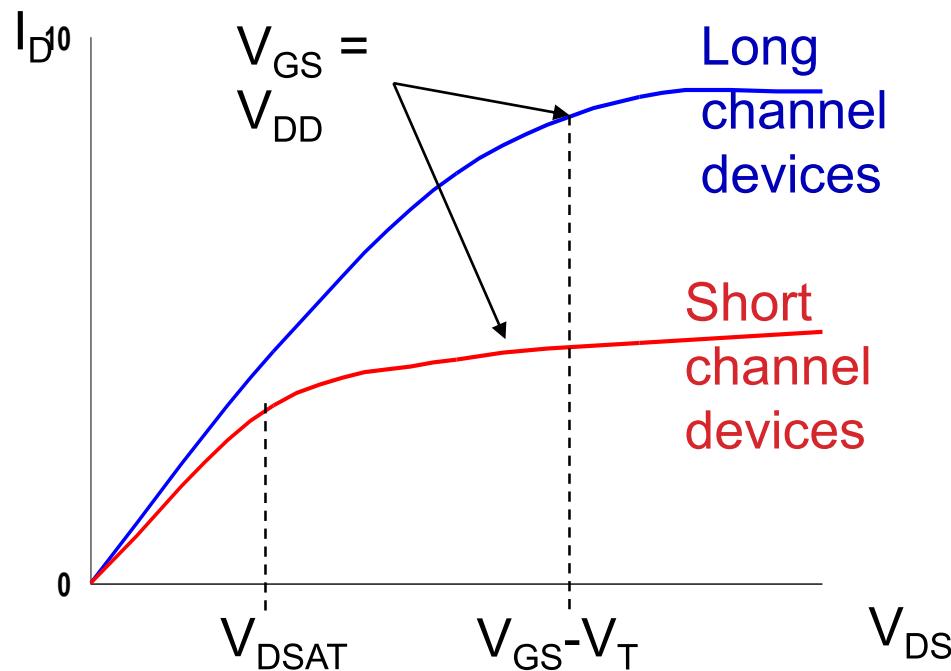


Velocity saturation the speed of carriers (due to the collisions) becomes constant

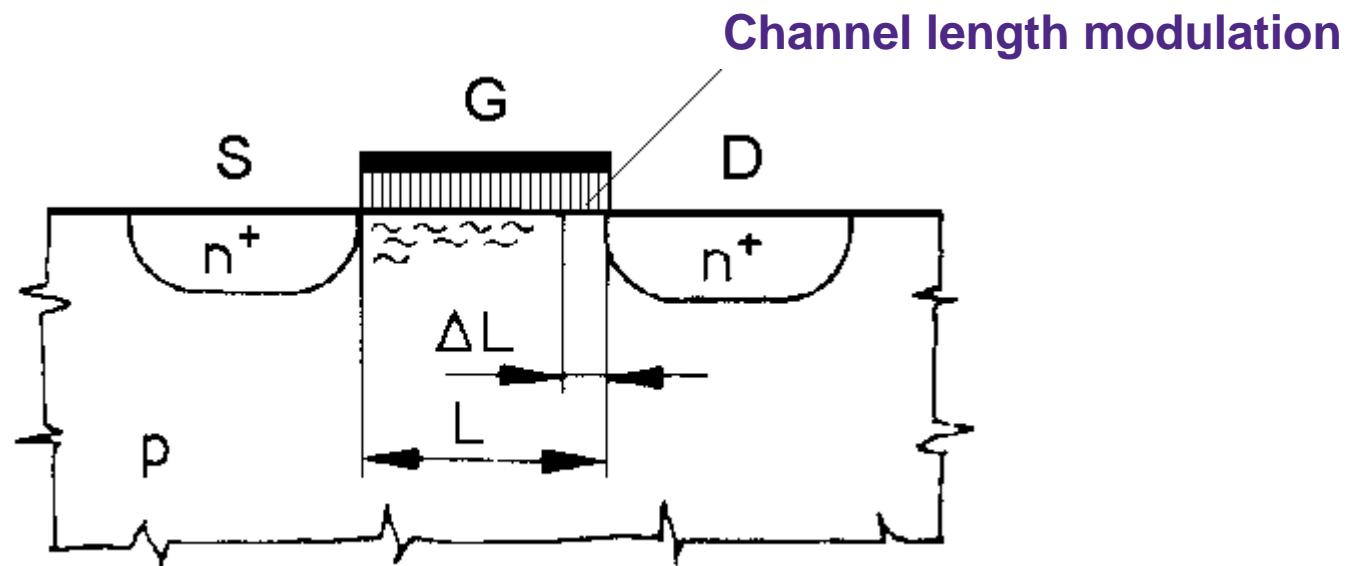
In a $L = 0.25\mu\text{m}$ channel device a few Volts of D-S voltage may already result in velocity saturation.

Velocity saturation

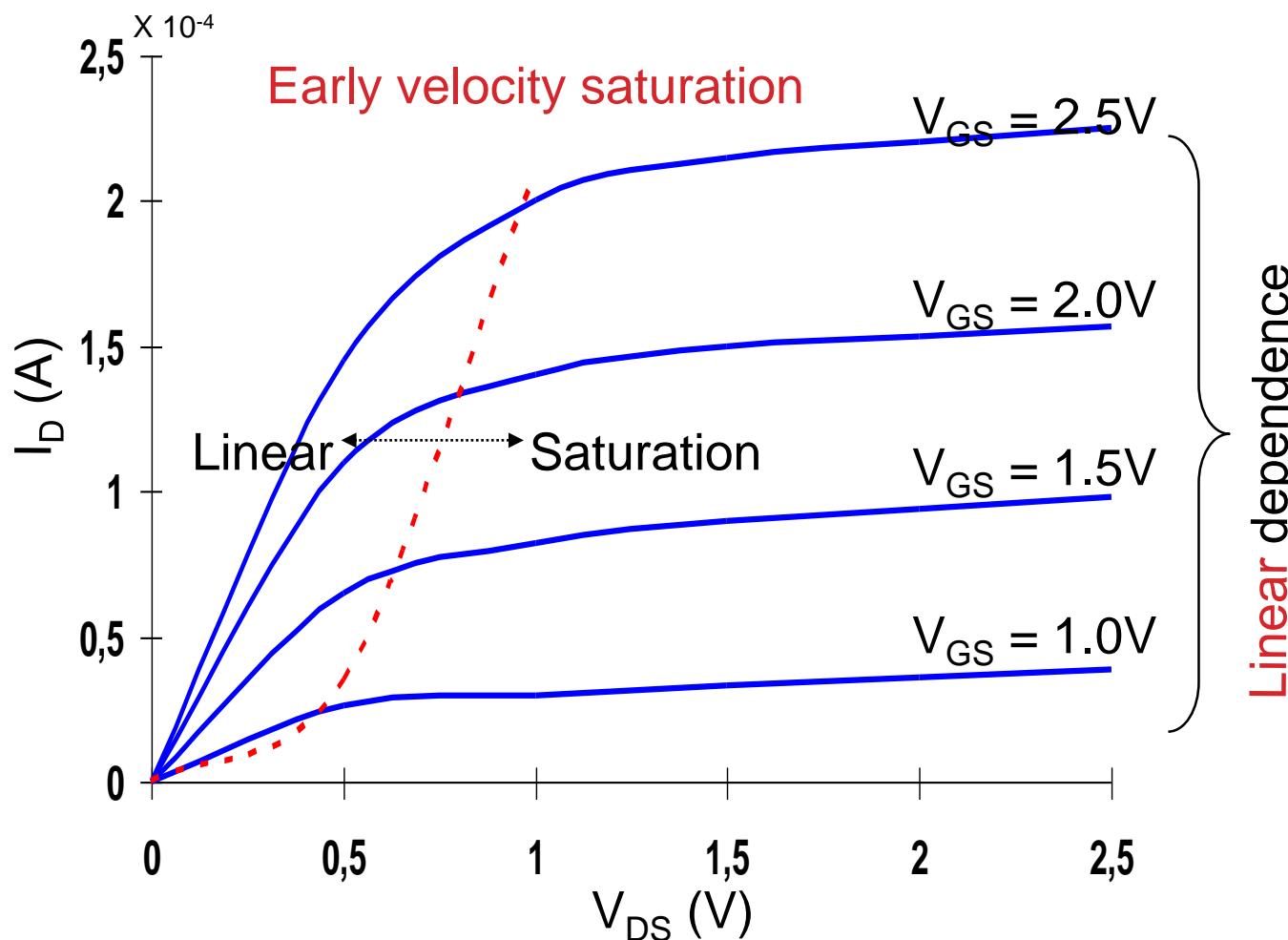
- In short channel device velocity saturation takes place sooner (at lower voltage)



Short channel characteristics



Short channel characteristics



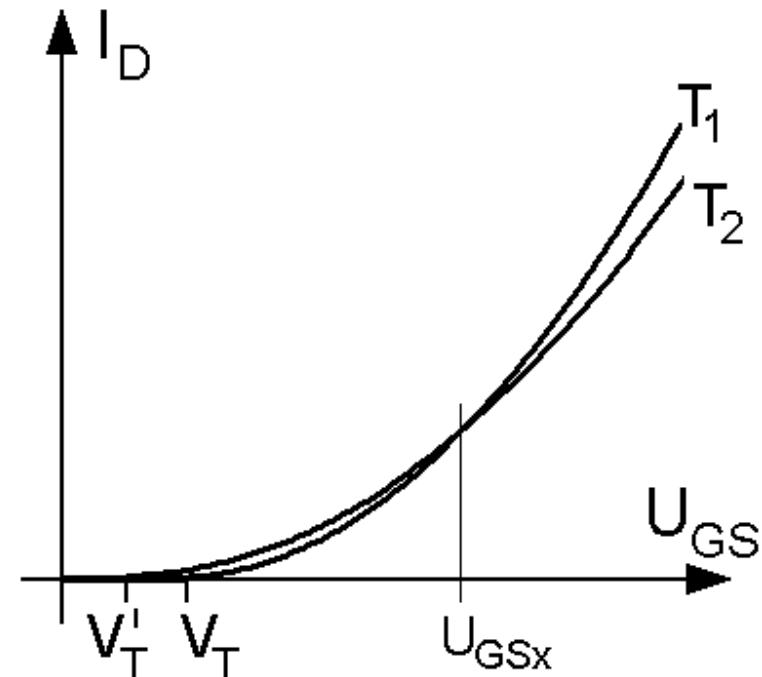
nMOS transistor, $0.25\mu m$, $L_d = 10\mu m$, $W/L = 1.5$, $V_{DD} = 2.5V$, $V_T = 0.4V$

Temperature dependence

$$I_D = \frac{W}{L} \frac{\mu C_0}{2} (U_{GS} - V_T)^2$$

$$\frac{1}{\mu} \frac{d\mu}{dT} = -0,003 \dots - 0,006 \text{ } /^\circ C$$

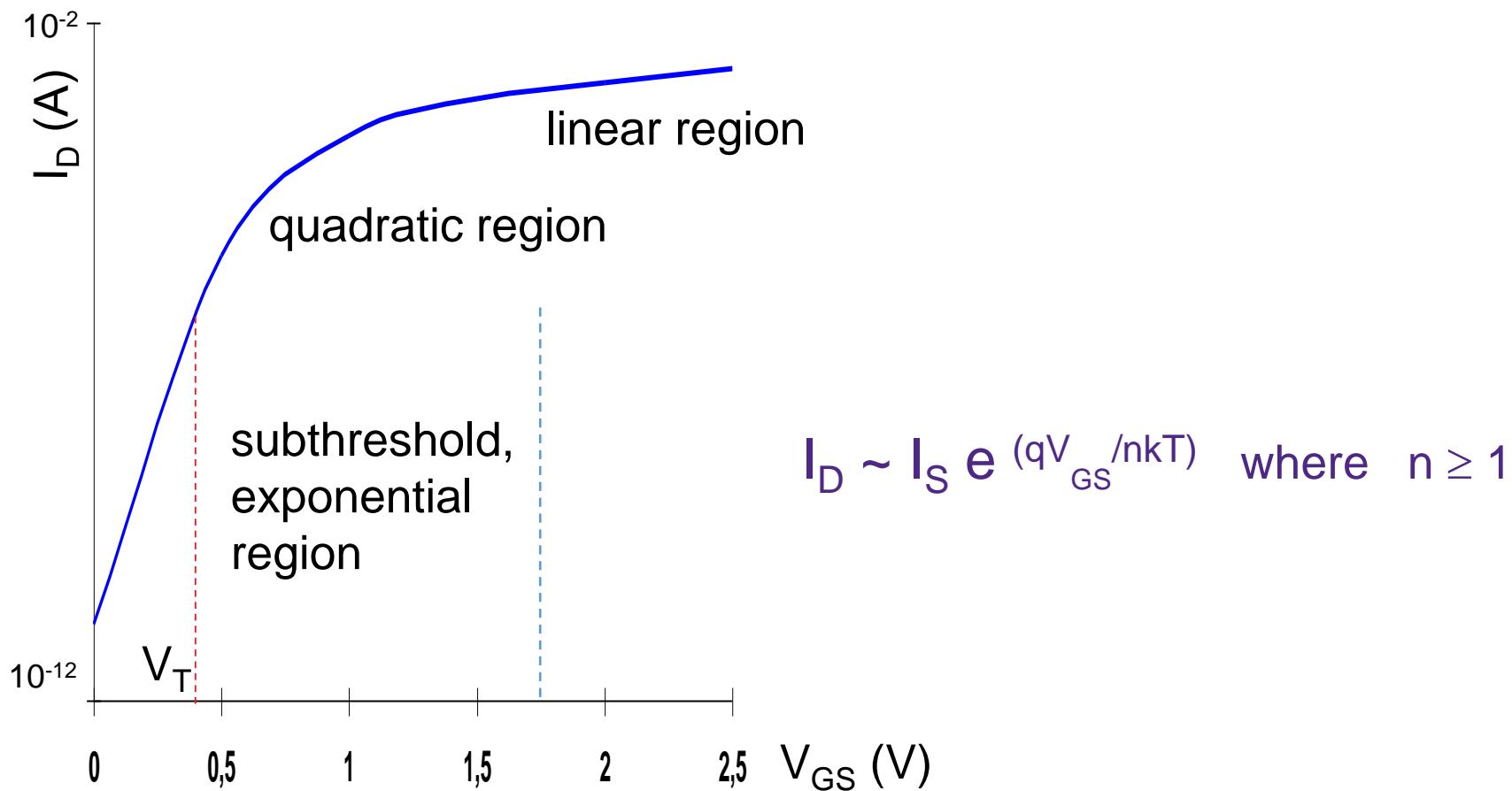
$$\frac{\partial V_T}{\partial T} = -1,5 \dots - 4 \text{ mV } /^\circ C$$



Zero Temperature Coefficient (ZTC) bias point

Subthreshold current

- Assuming a given V_T is rough model; in reality the current vanishes exponentially with the gate voltage:



Subthreshold current

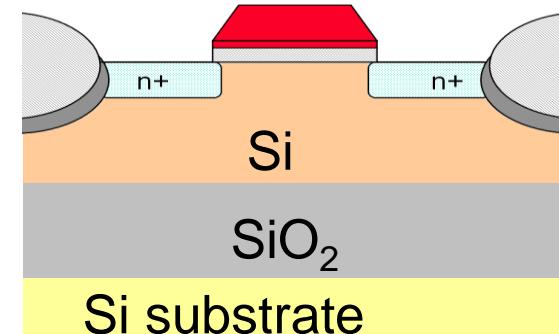
- Continuous transition between the ON and OFF states
 - Subthreshold is undesired: strong deviation from the **switch** model

$$I_D \sim I_0 e^{\frac{qV_{GS}}{nkT}}, \quad n = 1 + \frac{C_D}{C_{ox}}$$

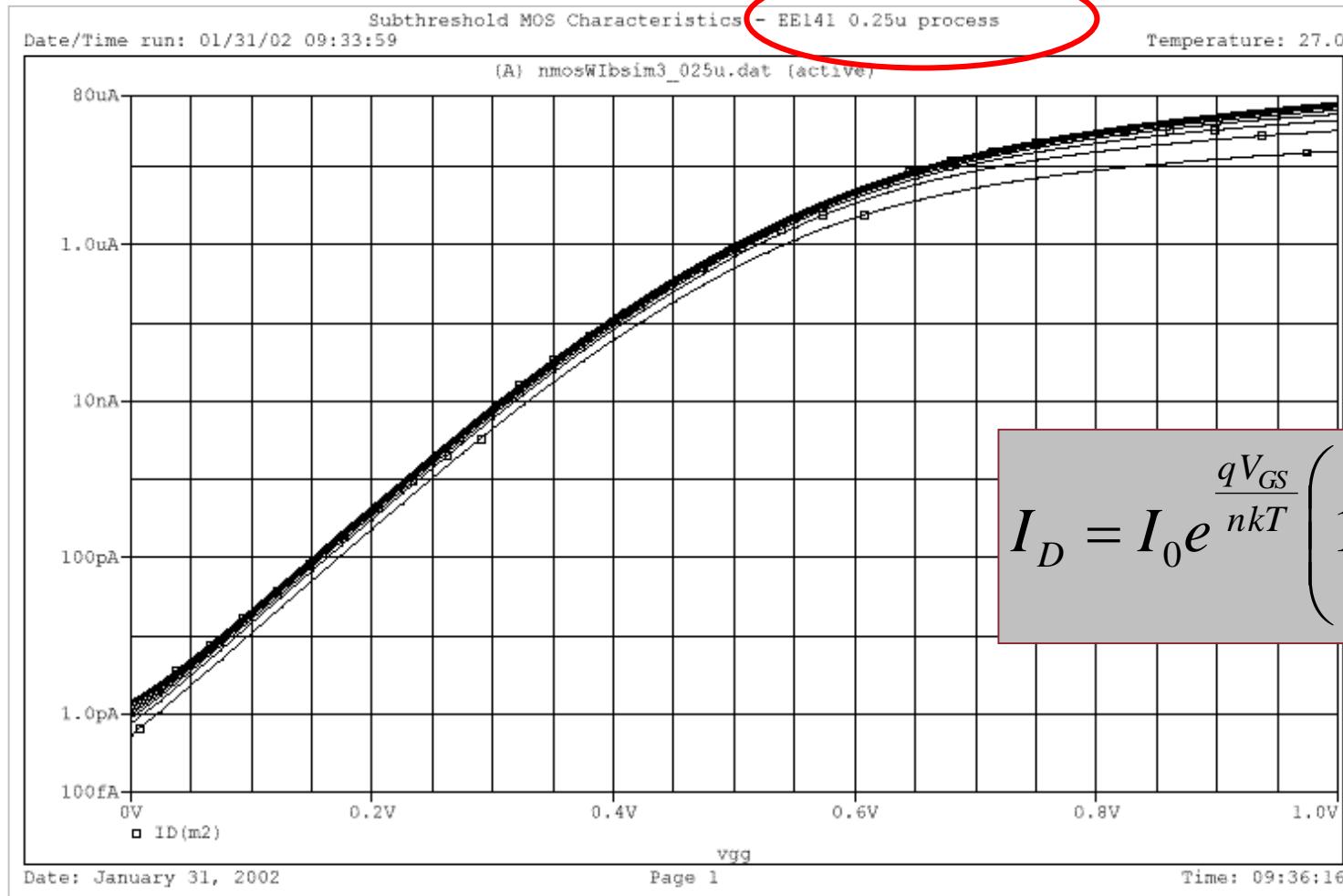
- I_0 , n – empirical parameters, n is typically 1.5
- Slope factor: $S = n (kT/q) \ln (10)$
(typically: 60 ..100 mV/decade) – the smaller the better, depends on.

Can be reduced by SOI:

e.g. *SiMOX* process



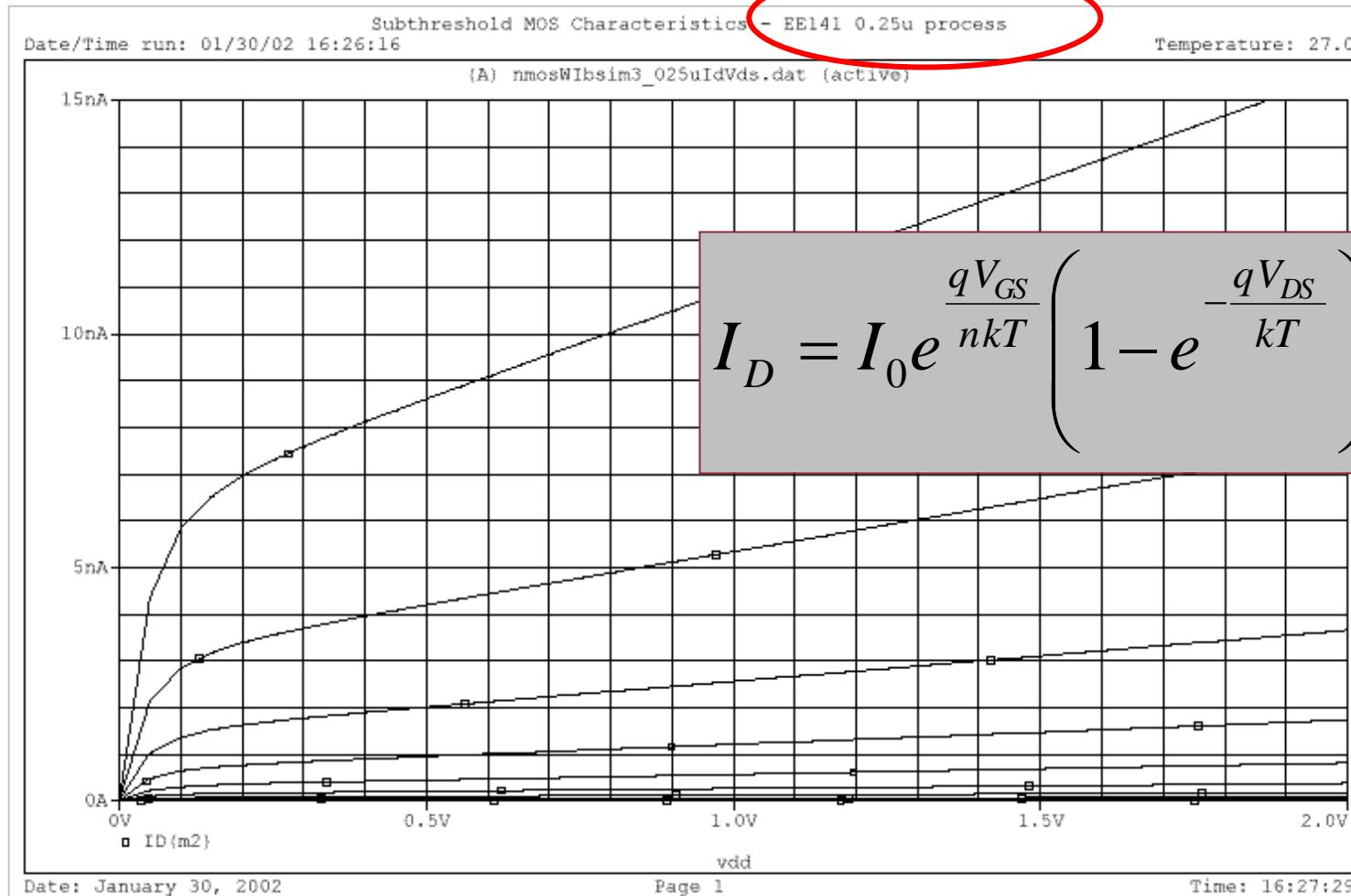
Subthreshold $I_D(V_{GS})$ characteristic



$V_{DS} : 0 .. 0.5V$

$$I_D = I_0 e^{\frac{qV_{GS}}{nkT}} \left(1 - e^{-\frac{qV_{DS}}{kT}} \right)$$

Subthreshold $I_D(V_{DS})$ characteristic



$V_{GS} : 0 .. 0.3V$

$$I_D = I_0 e^{\frac{qV_{GS}}{nkT}} \left(1 - e^{-\frac{qV_{DS}}{kT}} \right) (1 + \lambda \cdot V_{DS})$$

MOS transistor models

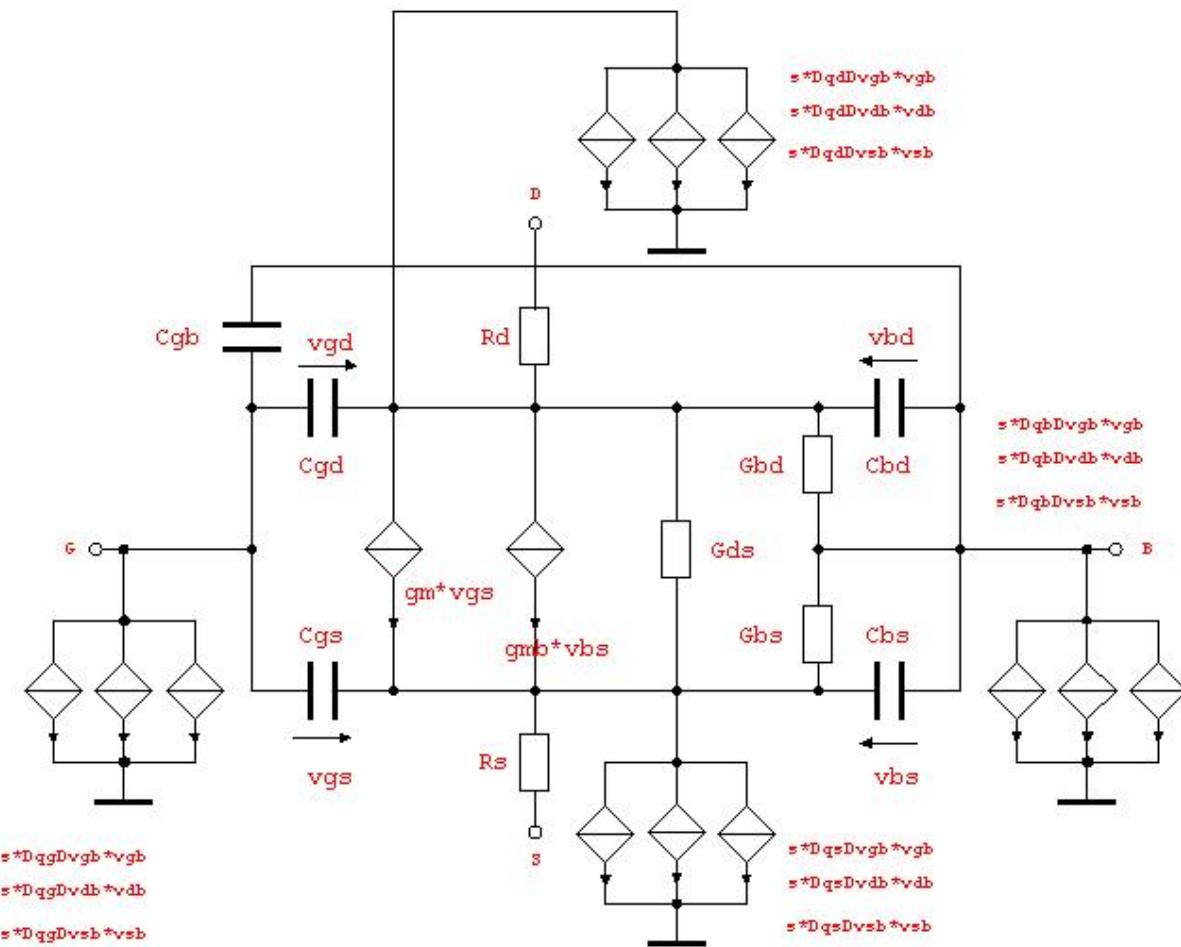
- Needed for circuit simulators (SPICE, TRANZ-TRAN, ELDO, SABER, etc.)
- Different levels of complexity:
 - level0, 1, 2, ...n,
 - EKV,
 - BSIM3, BSIM4

TABLE 8.1 SPICE2 and PSpice MOSFET DC Model Parameters.

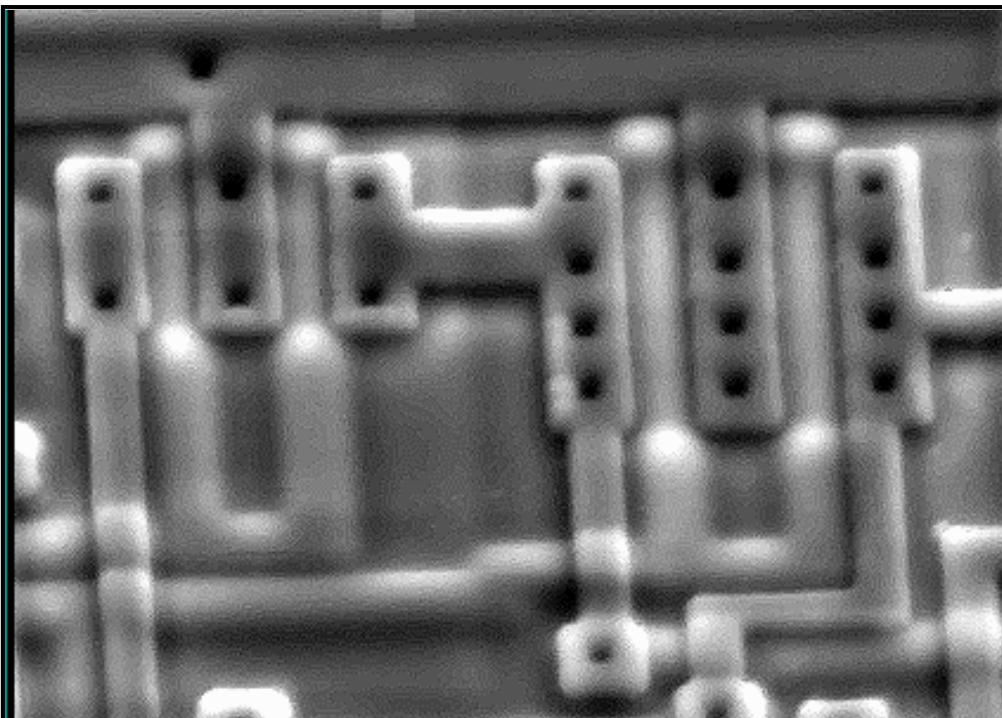
No.	Text Symbol	SPICE Keyword	Level	Parameter Name	Default Value	Units
1	—	LEVEL	1-3	SPICE model 1, 2 or 3	1	—
2	V_T	VTO	1-3	Zero-bias threshold voltage	0.0	V
3	γ	GAMMA	1-3	Bulk space-charge parameter	0.0	$V^{0.5}$
4	ψ_s	PHI	1-3	Surface potential	0.6	V
5	KP	KP	1-3	Transconductance parameter	2.0E-5	A/V^2
6	λ	LAMBDA	1, 2	Channel-length modulation	0	V^{-1}
7	t_{ox}	TOX	1-3	Gate-oxide thickness	1.0E-7	meter
8	N_b	NSUB	1-3	Substrate doping	0.0	cm^{-3}
9	N_f	NSS	2, 3	Fixed oxide charge	0.0	cm^{-2}
10	N_{it}	NFS	2, 3	Interface-trapped charge	0.0	cm^{-2}
11	—	TPG	2, 3	Type of gate material +1 opp. to substrate -1 same as substrate 0 Al gate	1	—
12	μ	UO	1-3	Surface mobility	600	cm^2/Vs
13	U_c	UCRIT	2	Critical electric field for mobility	1E4	V/cm
14	U_e	UEXP	2	Exponential coefficient for mobility	0.0	—
15	U_t	UTRA	2	Transverse field coefficient	0.0	—
16	x_j	XJ	2, 3	Source or drain junction depth	0.0	meters
17	x_{jl}	LD	1-3	Lateral diffusion	0.0	meters
18	v_{max}	VMAX	2, 3	Maximum carrier drift velocity	0.0	$meters/s$
19	N_{eff}	NEFF	2	Total channel charge coefficient	1	—
20	δ	DELTA	2, 3	Width effect on threshold voltage	0.0	—
21	η	ETA	3	Static feedback on threshold voltage	0.0	—
22	V_{bi}	PB	1-3	Source and drain junction built-in potential	0.80	V
23	θ	THETA	3	Mobility modulation	0.0	—
24	κ	KAPPA	3	Saturation field factor	0.2	—

MOS transistor models

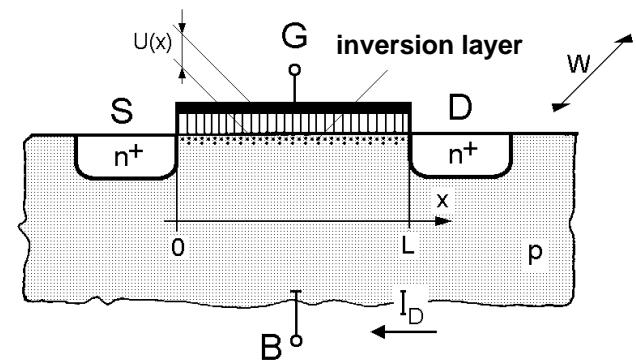
- Needed for circuit simulators (SPICE, TRANZ-TRAN, ELDO, SABER, etc.)
- Different levels of complexity:
 - level0, 1, 2, ...n,
 - EKV,
 - BSIM3, BSIM4



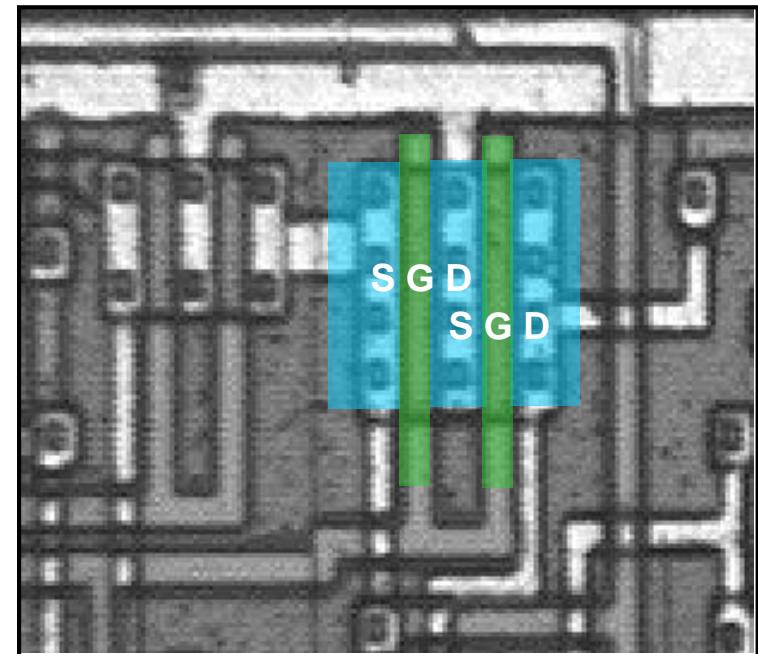
Examples for MOSFETs



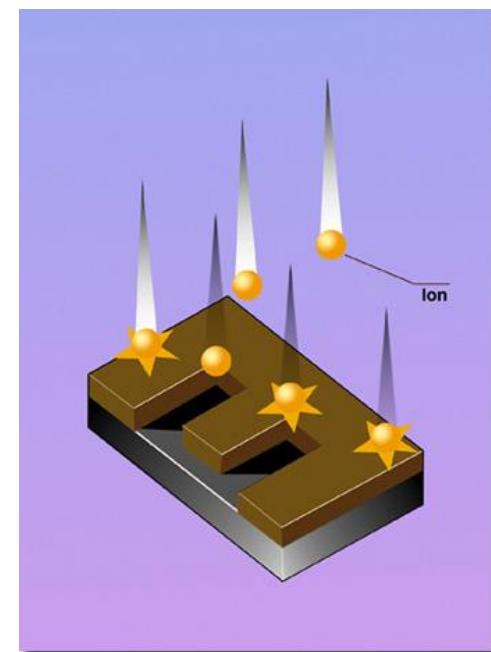
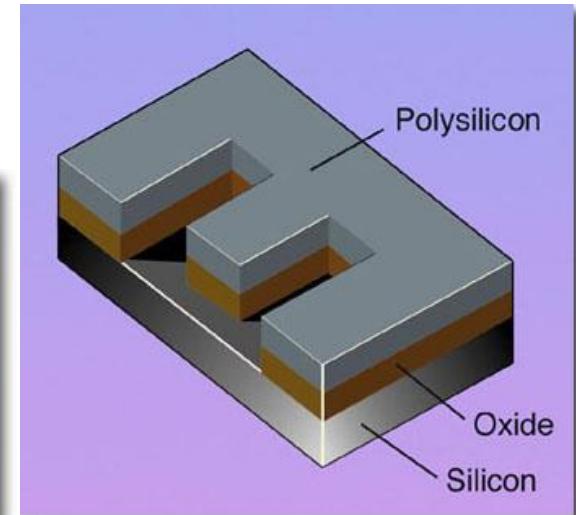
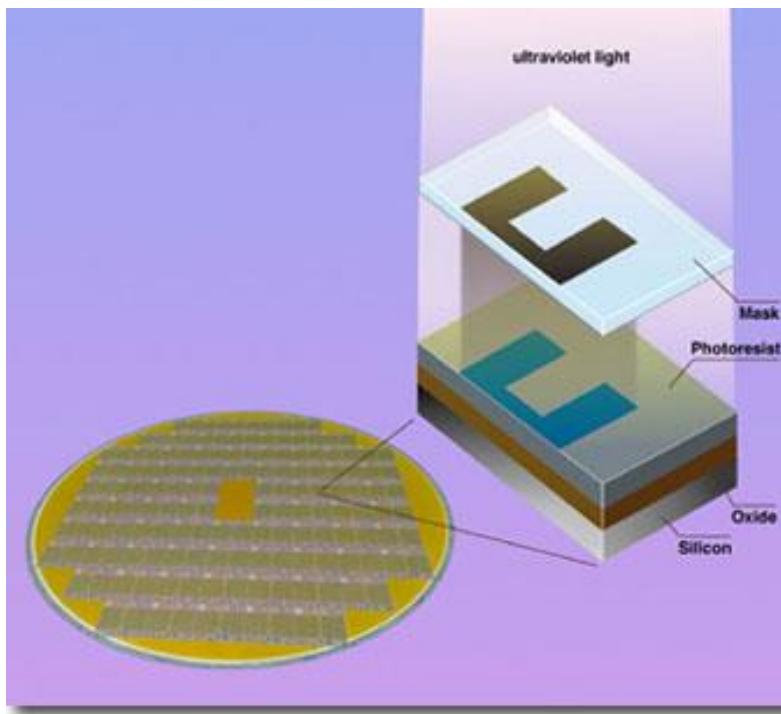
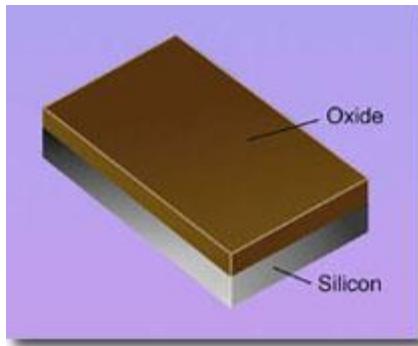
Micro-photograph by SEM



Photograph by
optical microscope

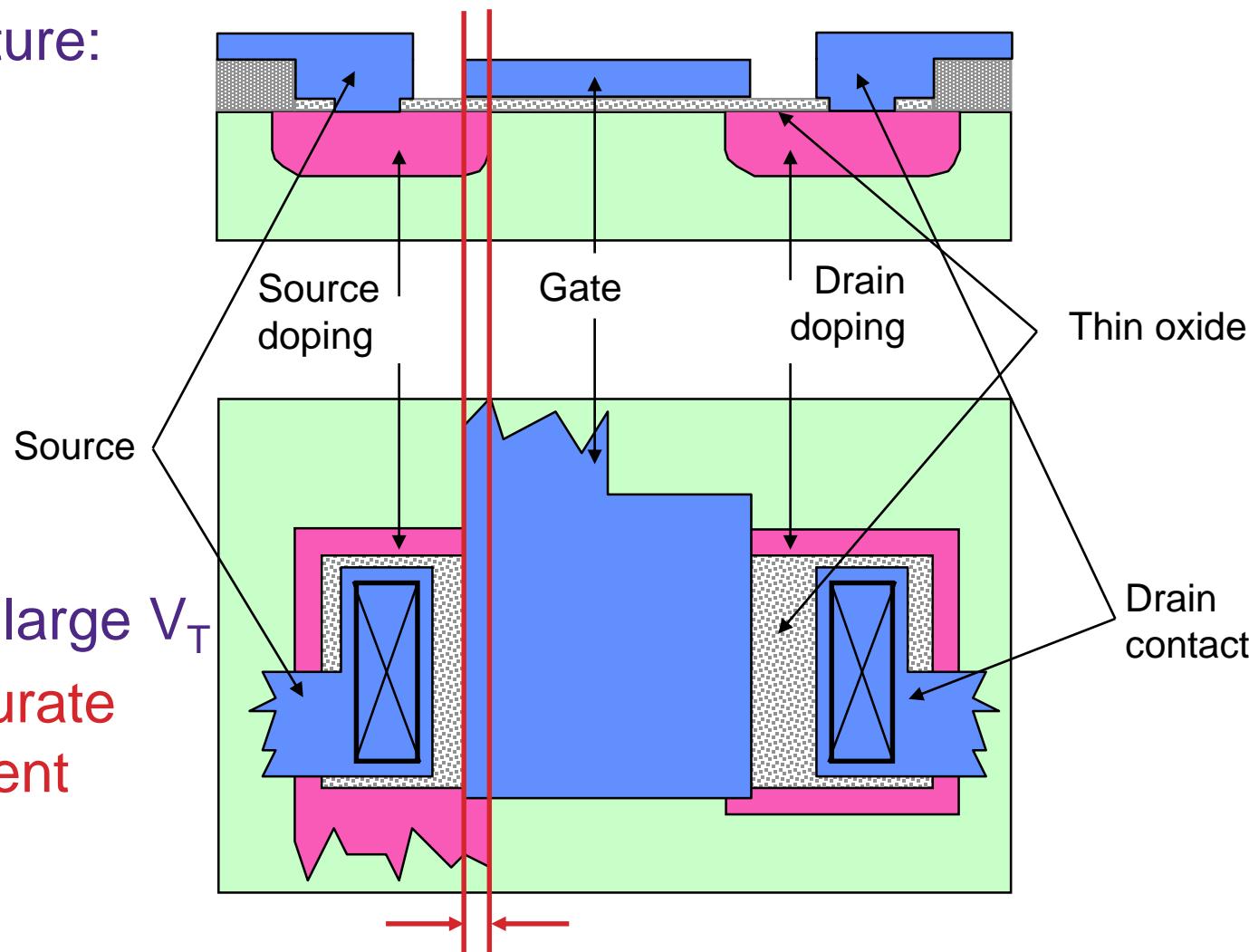


How is it manufactured?



Metal gate MOS transistor

In-depth structure:



Layout view:

Problems:

- metal gate – large V_T
- requires accurate mask alignment

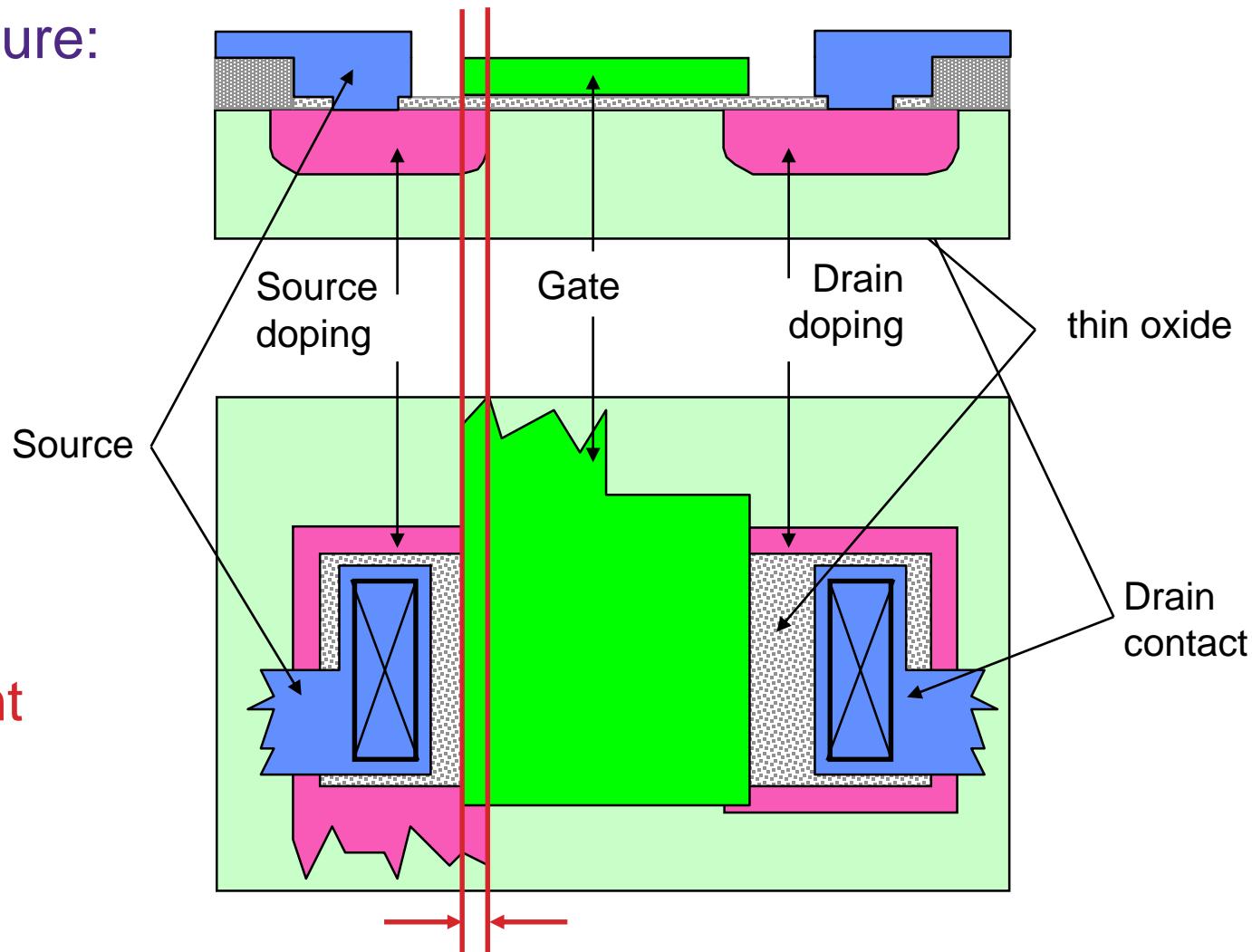
Poly-Si gate MOS transistor

In-depth structure:

Layout view:

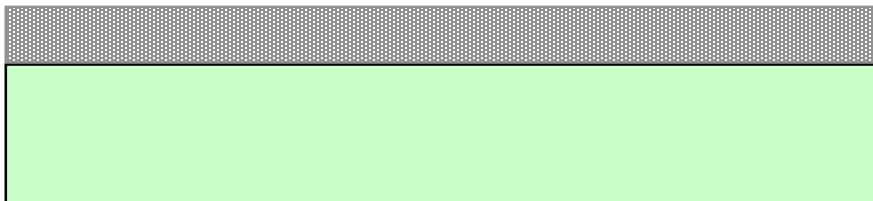
Advantages

- smaller V_T
- self alignment



A poli-Si gate nMOS process

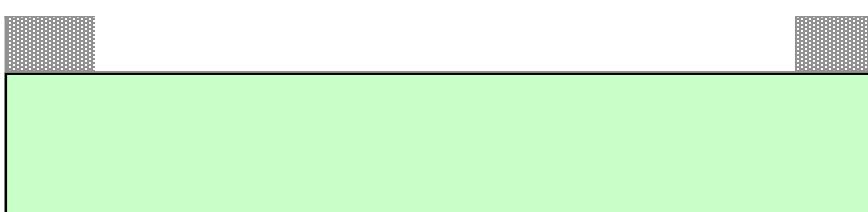
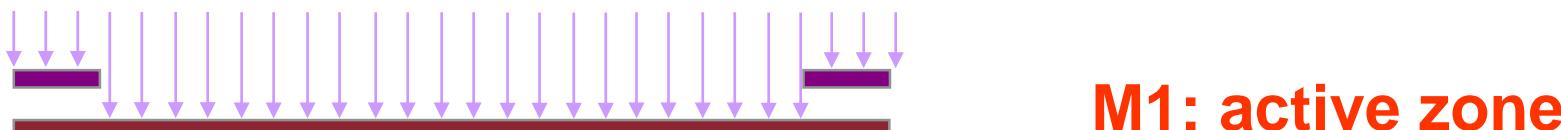
- Start with: p type substrate (Si wafer)
 - cleaing,
 - grow thick SiO_2 – this is called *field oxide*



The poli-Si gate nMOS process

- Create the active zone with photolithography

- coat with resist,
 - expose to UV light through a mask,
 - development, removal of exposed resists
 - etching of SiO_2 removal of the resist

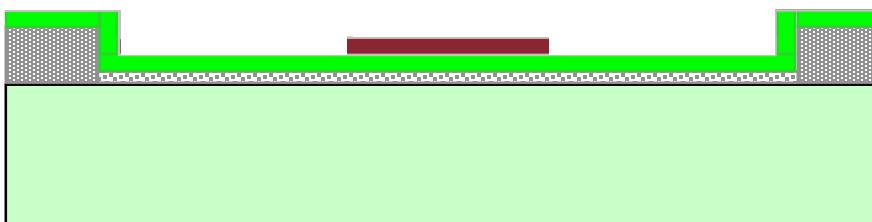


The poli-Si gate nMOS process

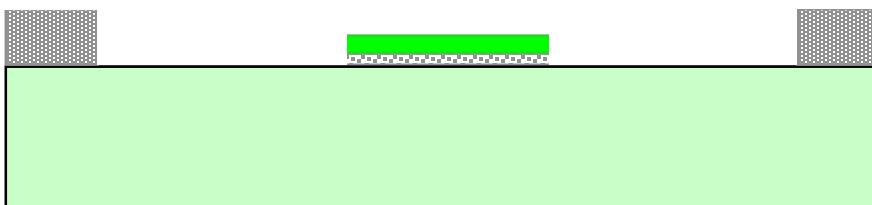
- Create the gate structure:

- growth of thin oxide
- deposit poly-Si
- pattern poly-Si with photolithography
- etch poly-Si, etch thin oxide

(resist, exposure, develop)



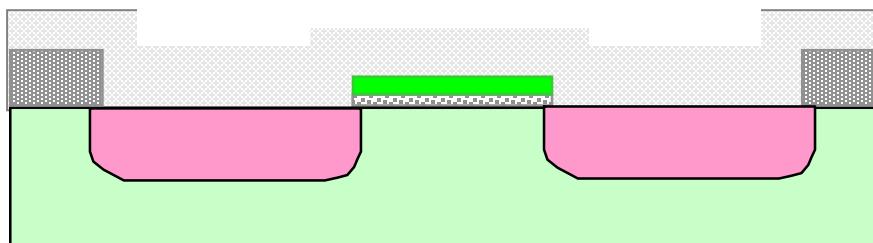
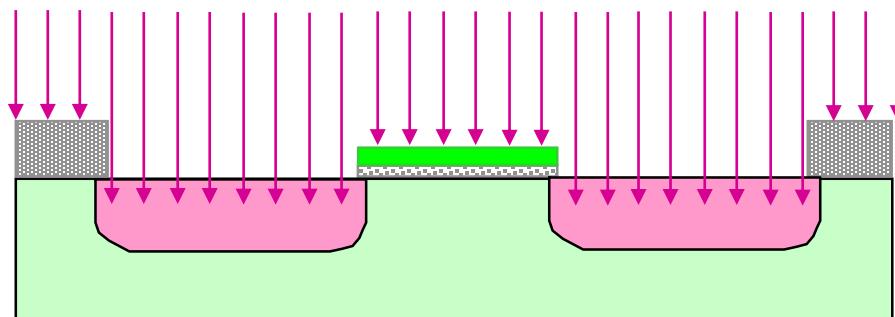
M2: poly-Si pattern



The poli-Si gate nMOS process

- S/D doping (implantation)
 - the oxide (thin, thick) masks the dopants
 - this way the self-alignment of the gate is assured

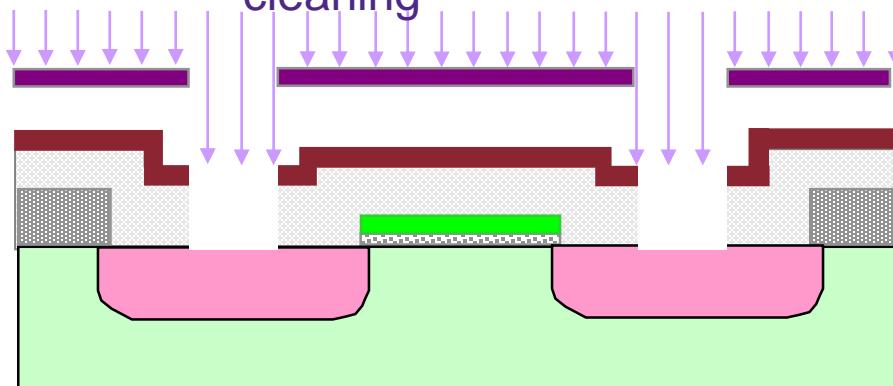
► Passivation: deposit PSG



The poli-Si gate nMOS process

- Open contact windows through PSG

- photolithography (resist, expose pattern, develop)
 - etching (copy the pattern)
 - cleaning

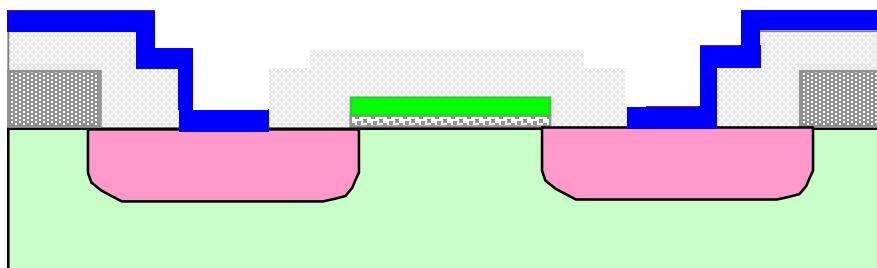


M3: contact window pattern

The poli-Si gate nMOS process

- Metallization

- Deposit Al
 - photolithography, etching, cleaning

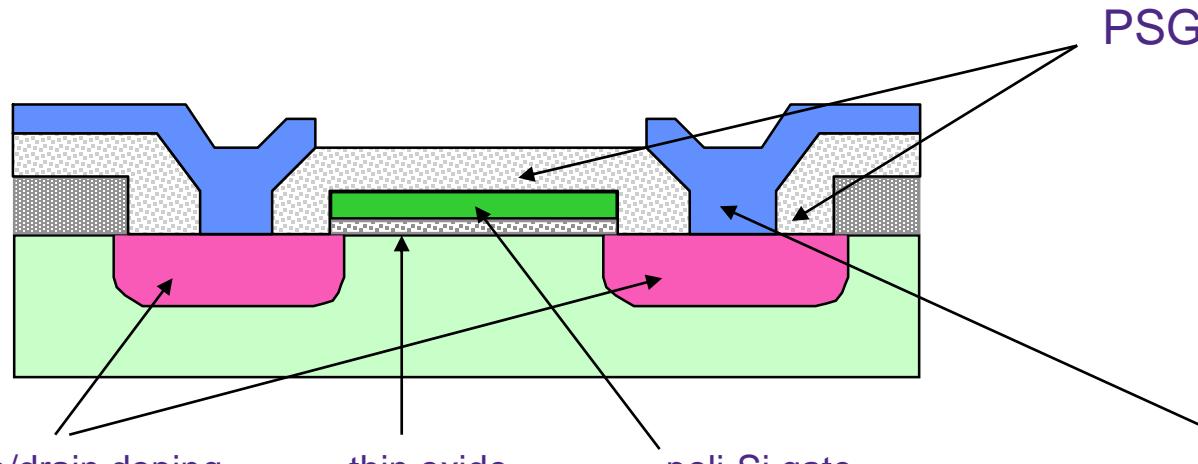


M4: metallization pattern

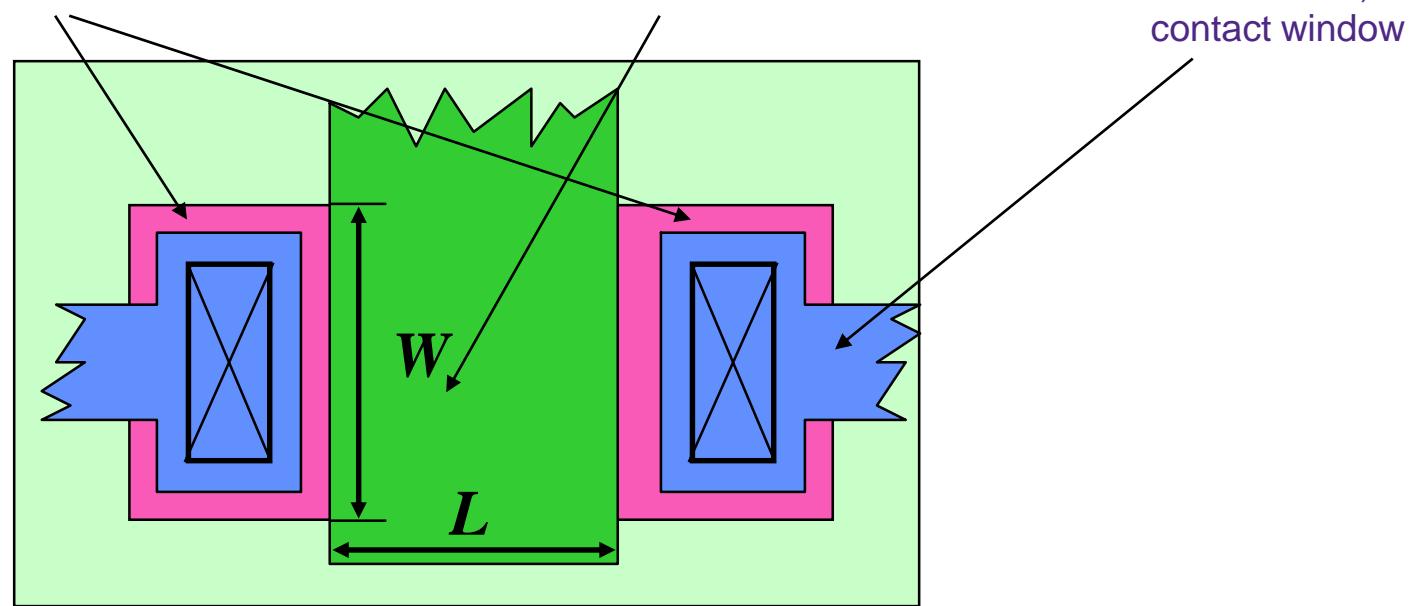
- ▶ The recipe of the process is given, the in-depth structure is determined by the sequence of the masks
- ▶ One needs to specify the shapes on the masks
 - ***The set of shapes on subsequent masks is called layout***

Poli-Si gate self-aligned device

Structure:



Layout:



Steps of the self-aligned poli-Si gate process

- 1) Open window for the active region
 - photolithography, field oxide etching
- 2) Growth of thin oxide
- 3) Window for hidden contacts
 - Contacts the poli-Si gate (yet to be deposited) with the active region (after doping).
- 4) Deposit poli-Si
- 5) Patterning of poli-Si
- 5) Open window through the thin oxide (etching only)

M

M

M

Steps of the self-aligned poli-Si gate process

6) n+ doping:

Form source and drain regions as well as wiring by diffusion lines. Through the hidden contact poli-Si gate will also be connected to diffused lines.

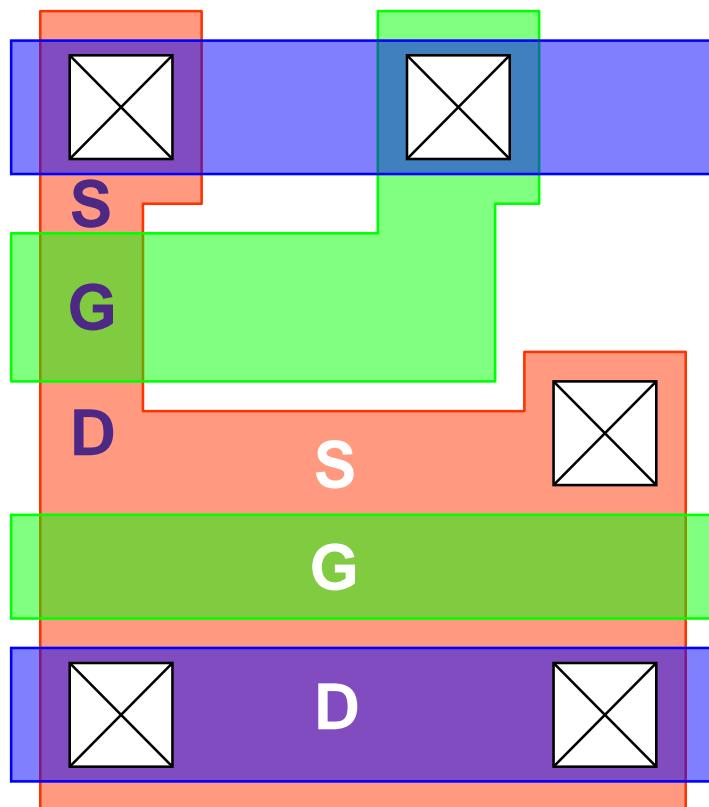
7) Deposit phosphor-silica glass (PSG) as insulator

8) Open contact windows through PSG-n M

9) Metallization

10) Patterning metallization layer

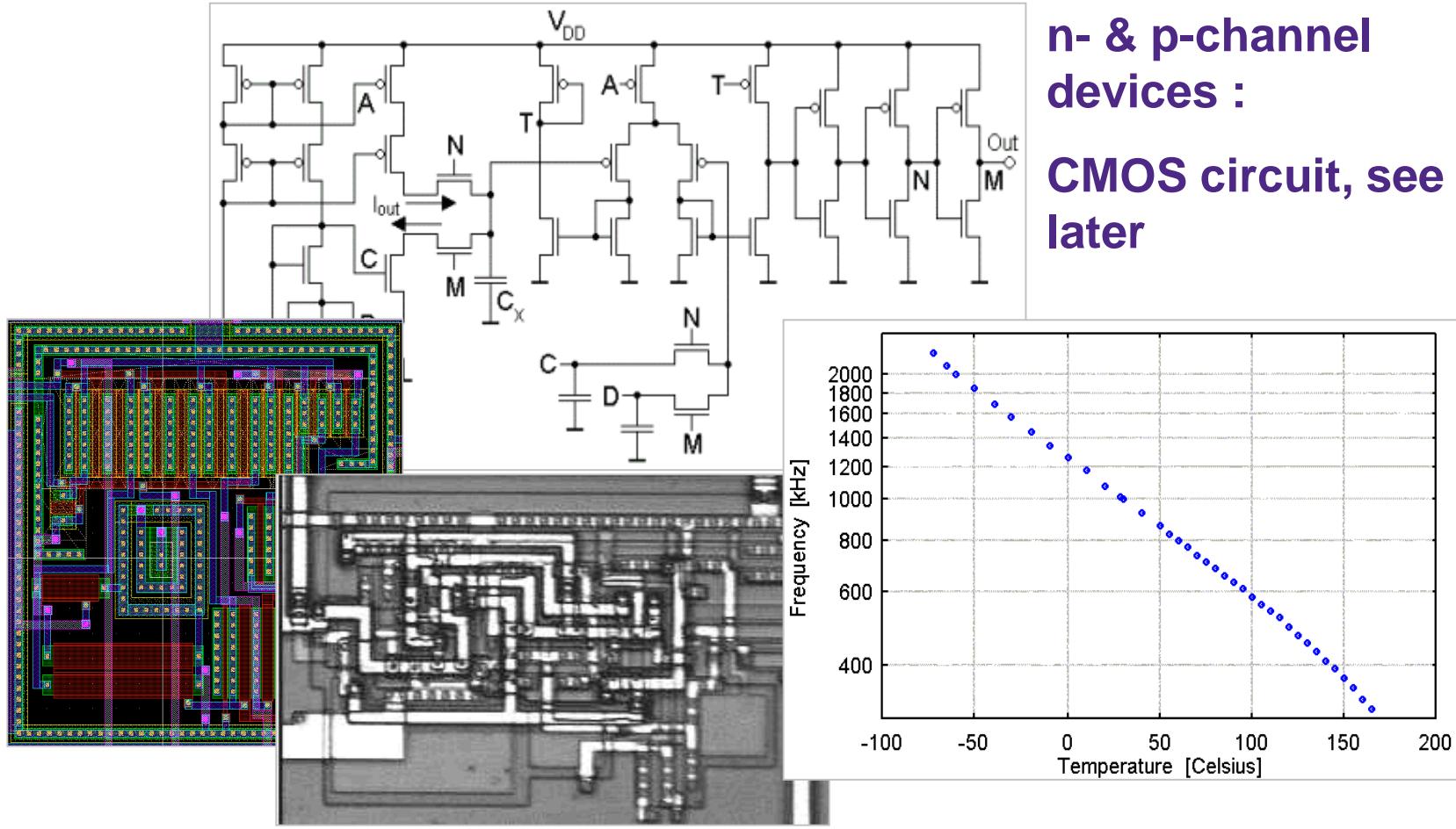
Layout of a depletion mode inverter



- Layout == set of 2D shapes on subsequent masks
- Masks are color coded:
 - active zone: red
 - poly-Si: green
 - contact windows: black
 - metal: blue
- Mask == layout layer

Where is a transistor? Channel between two doped regions:
CHANNEL = ACTIVE AND POLY

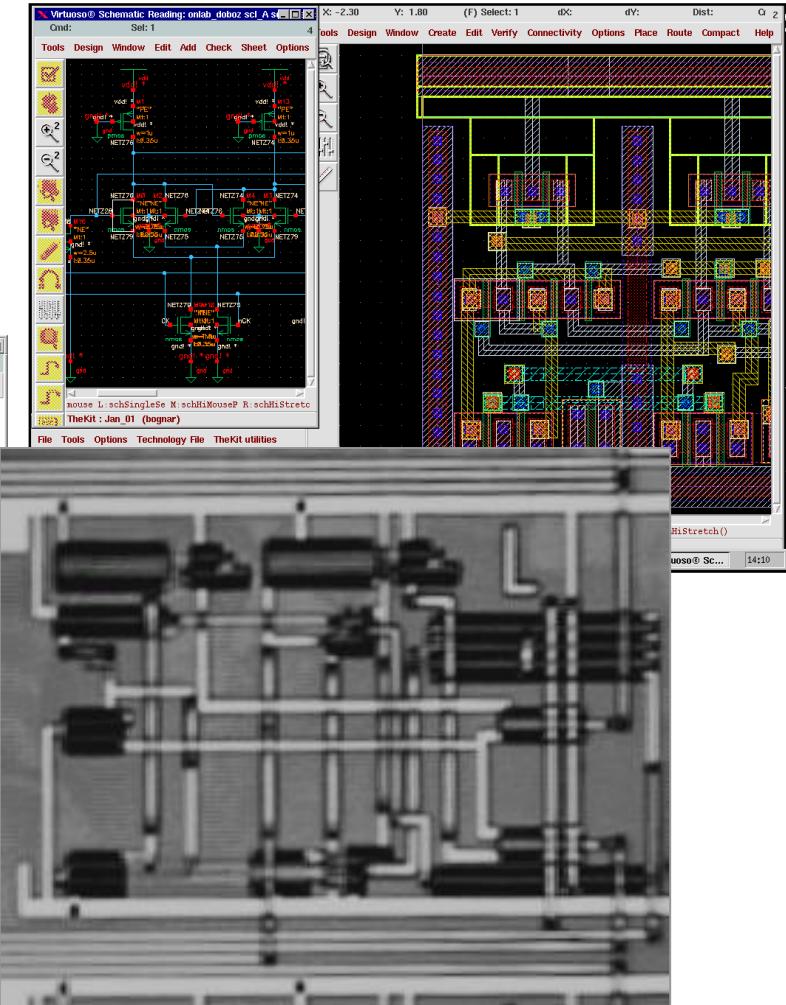
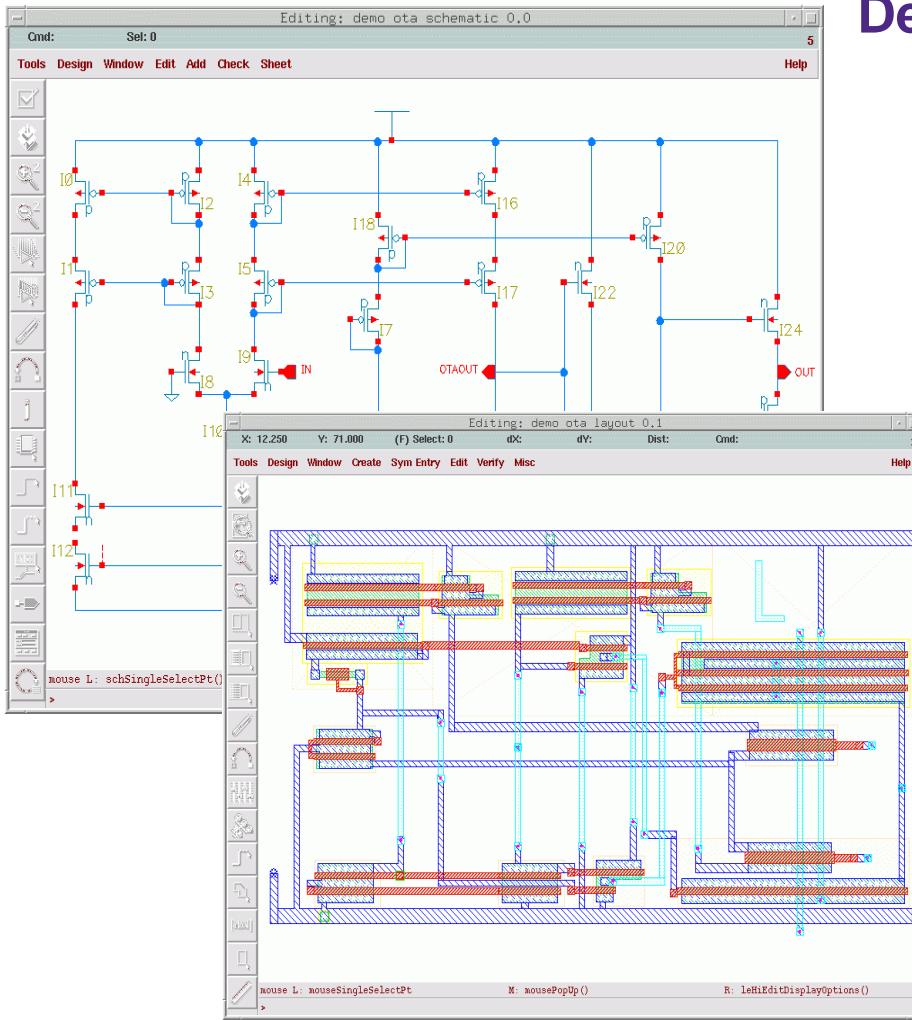
Some more complex MOS circuits



n- & p-channel devices :
CMOS circuit, see later

Some more complex MOS circuits

Designed by CAD tools



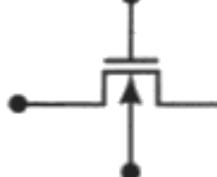
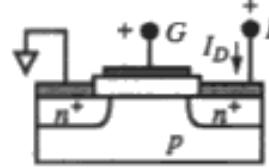
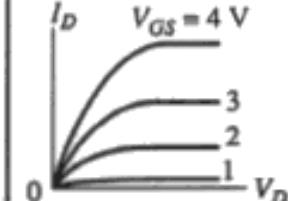
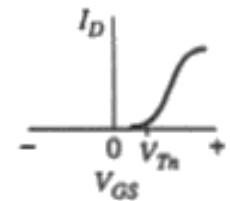
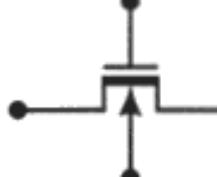
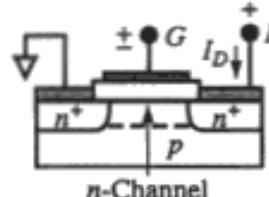
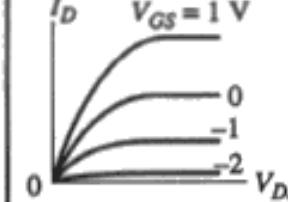
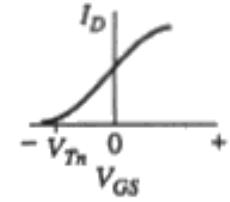
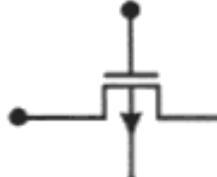
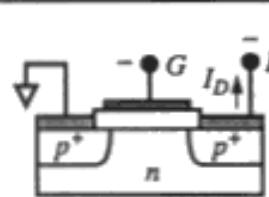
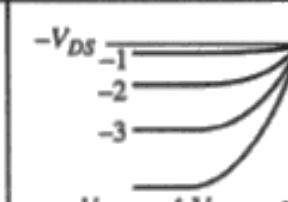
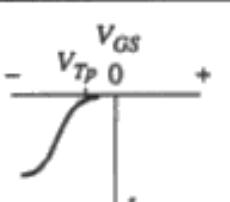
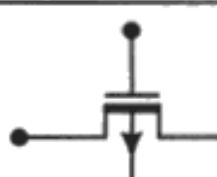
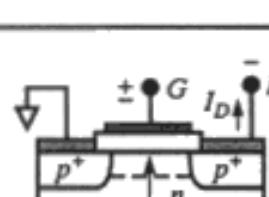
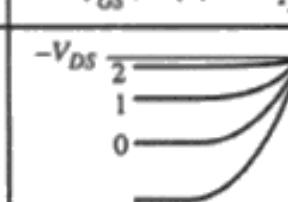
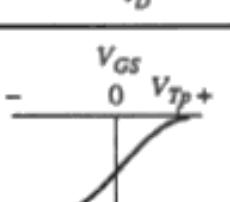


Budapest University of Technology and Economics
Department of Electron Devices

Microelectronics, BSc course

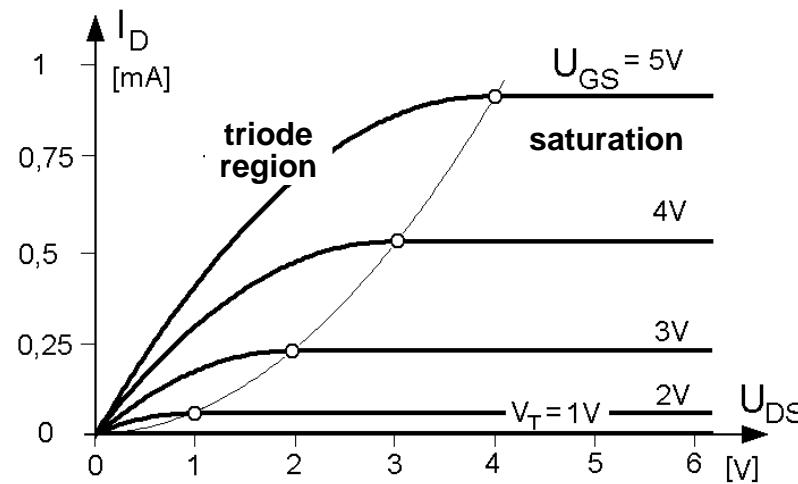
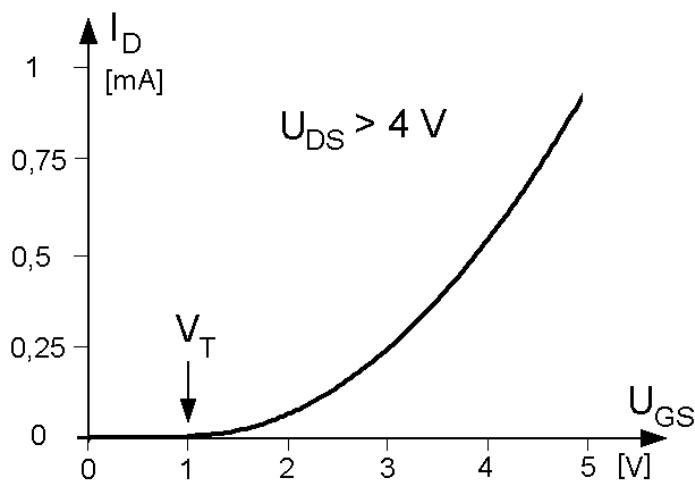
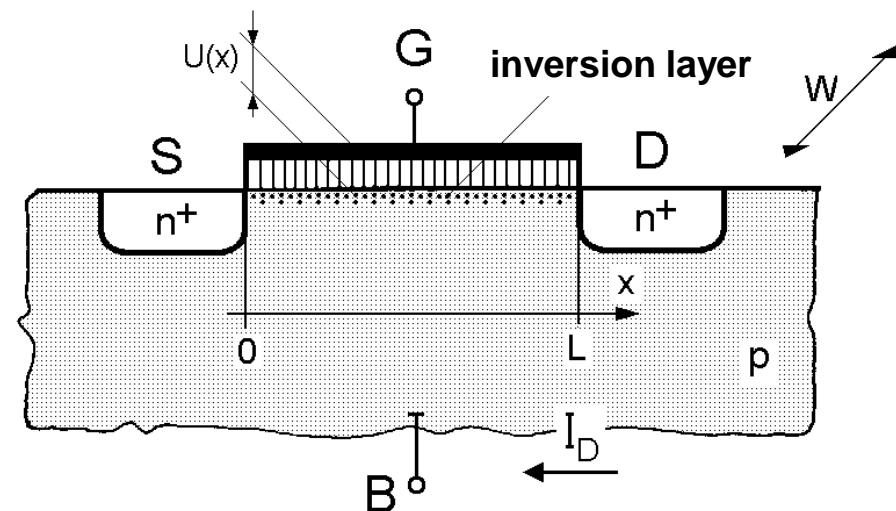
nMOS/CMOS Logic Gates

Overview of MOSFET types

Type	Circuit Symbol	Cross Section	Output Characteristics	Transfer Characteristics
n-Channel Enhancement (Normally Off)				
n-Channel Depletion (Normally On)		 n-Channel		
p-Channel Enhancement (Normally Off)				
p-Channel Depletion (Normally On)		 p-Channel		

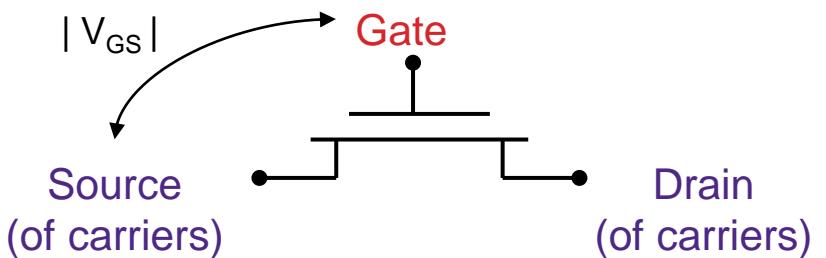
Char. of enhancement mode n type MOSFETs

Now we calculate with this!



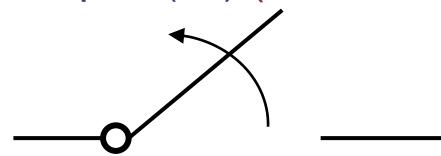
Simple model of MSOFETs

- The simplest (logic) model:
 - no conduction (off) / conduction (on)



enhancement mode device

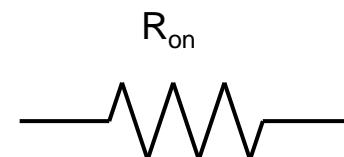
Open (off) (Gate = '0')



$$|V_{GS}| < |V_T|$$

"open"

Closed (on) (Gate = '1')

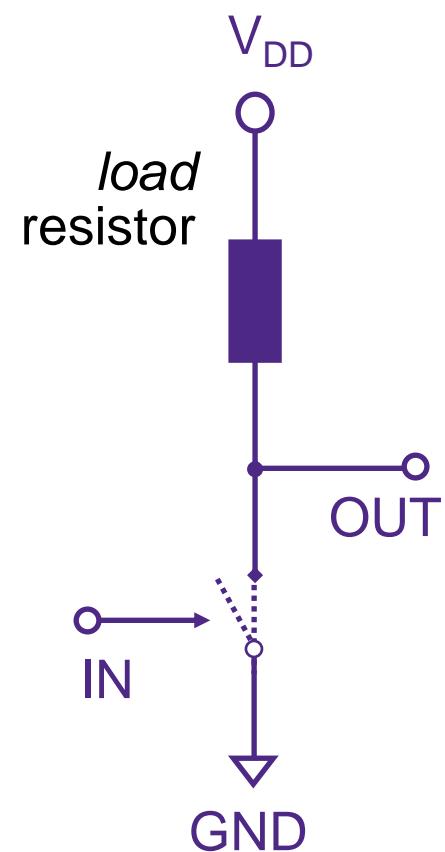


$$|V_{GS}| > |V_T|$$

"short"

Let's construct an inverter!

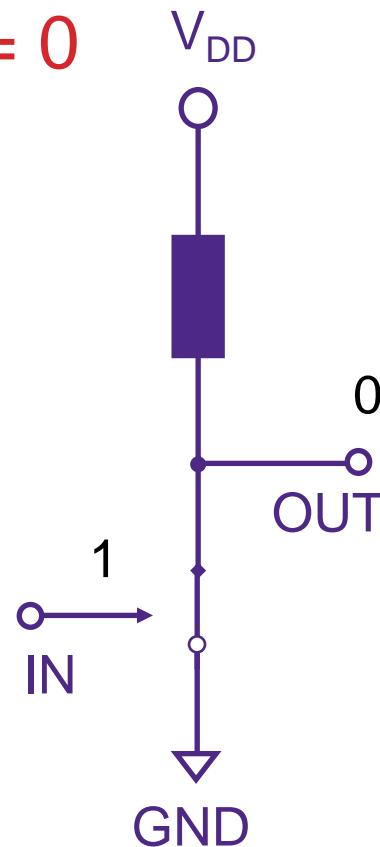
- Resistor at supply voltage (V_{DD})
- Other end connected to ground (GND) through a switch
- Switch controlled by a logic signal:
 - 1 (V_{DD} level) – "short"
 - 0 (GND level) – "open"
- The output is the common node of the switch and the resistor



Let's construct an inverter!

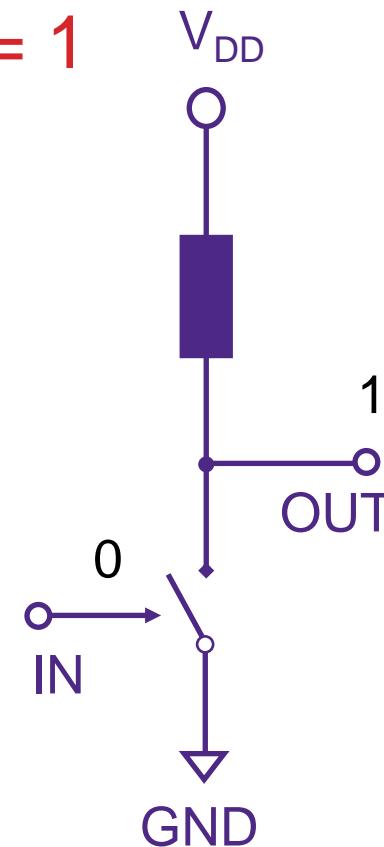
► IN = 1

- switch "on"
- output connected to GND
- OUT = 0

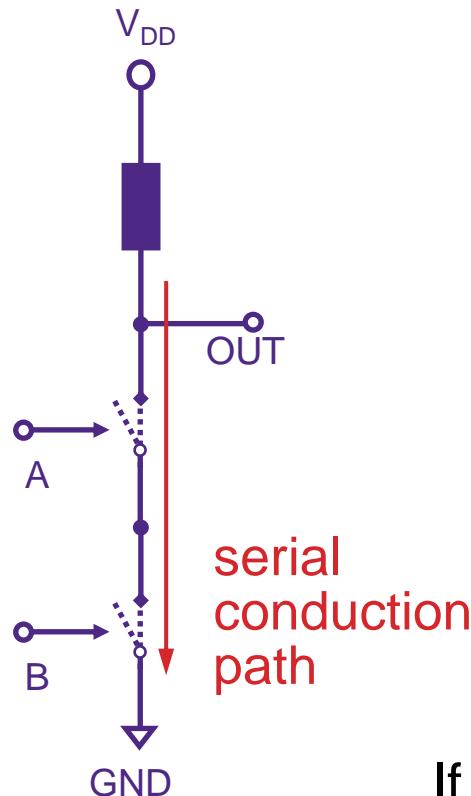


► IN = 0

- switch "off"
- output floating at V_{DD}
- OUT = 1



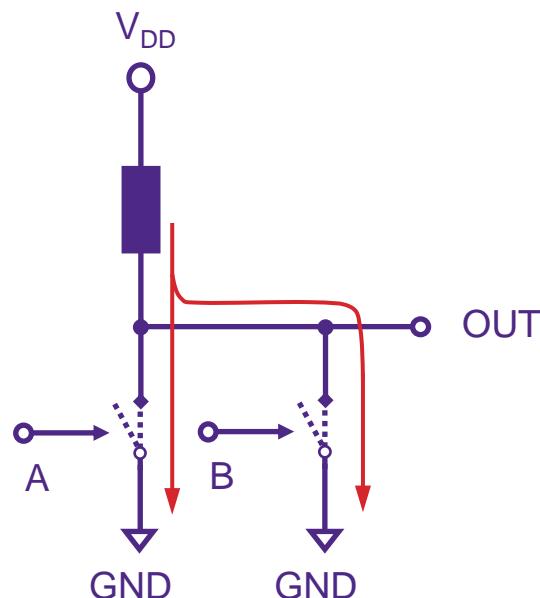
Two switches in series: NAND gate



- ▶ If A and B equal to 1, then $OUT=0$
 - ▶ This is the NOT (A AND B) function, i.e. NAND
- In practice with max. 3..4 inputs.

If there are *parallel* conductive paths then we get the *NOR function*

The scheme of the NOR gate:



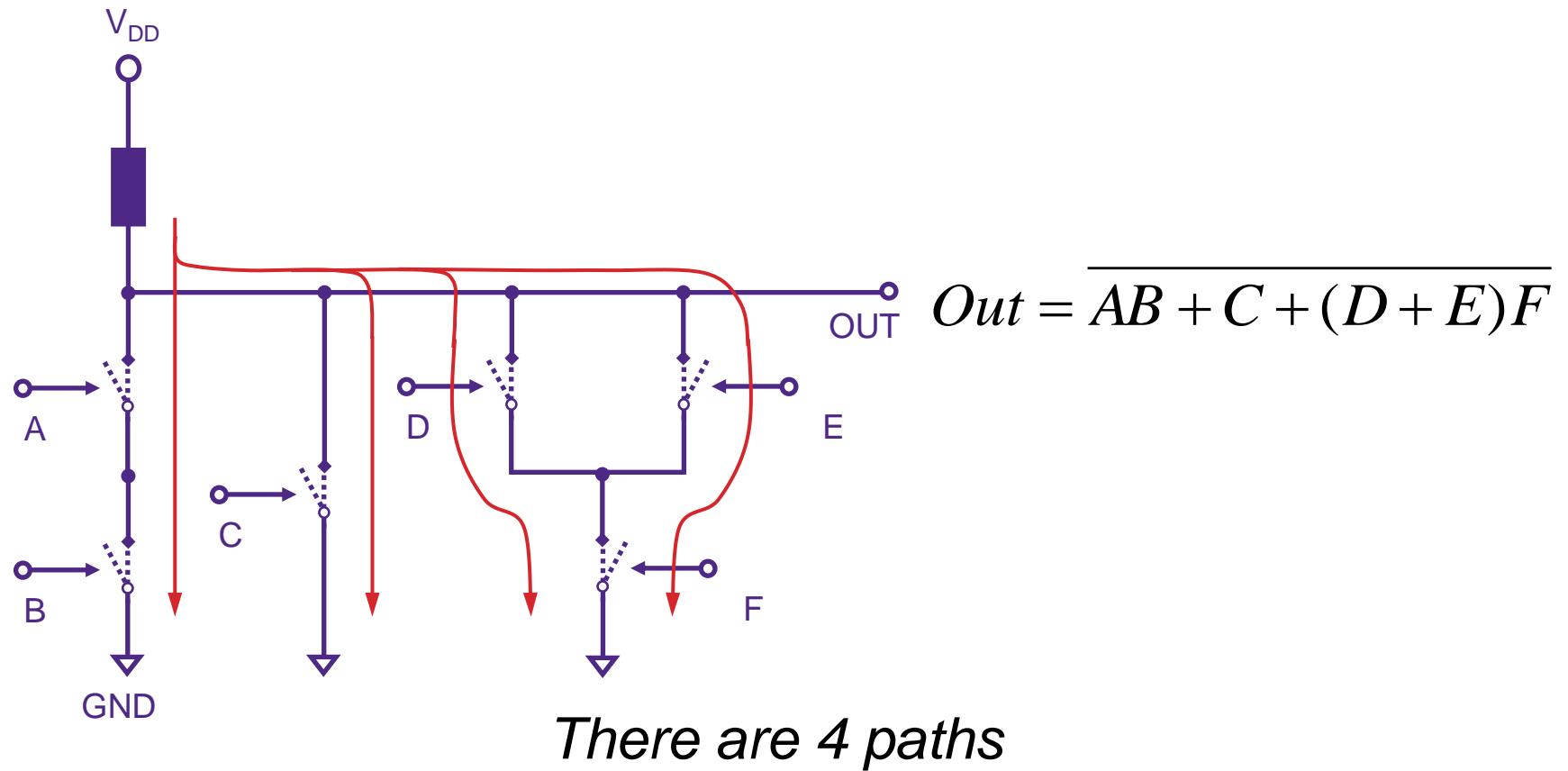
- ▶ If A or B equals to 1, then $OUT=0$
- ▶ This is the NOT (A OR B) function, i.e. NOR

PARALLEL
conduction path

Complex conduction paths == option for complex logic gates

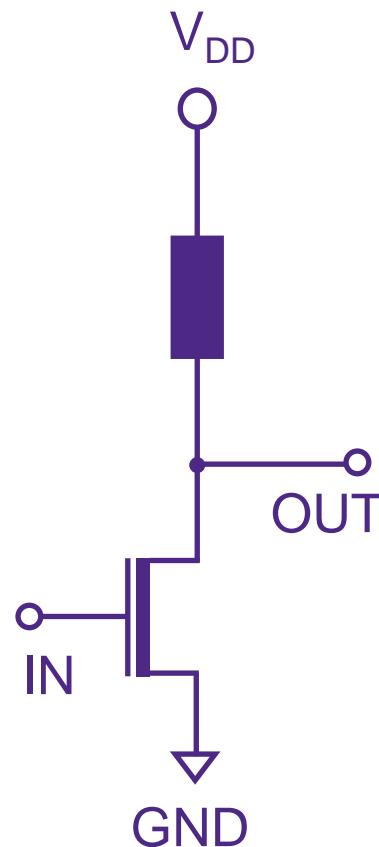
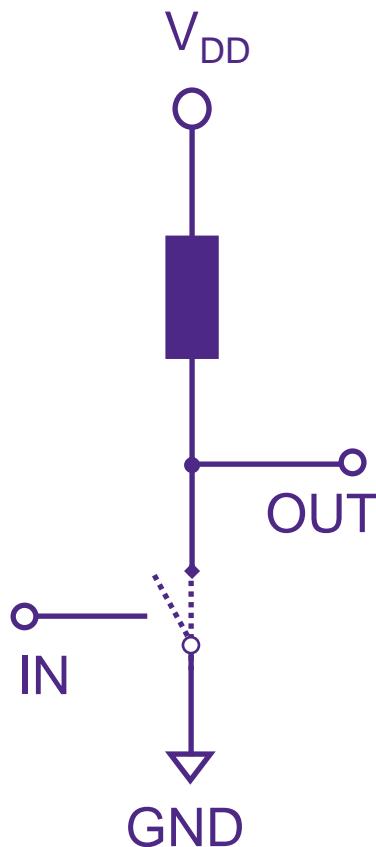
Complex logic gates

- Serial paths connected in parallel

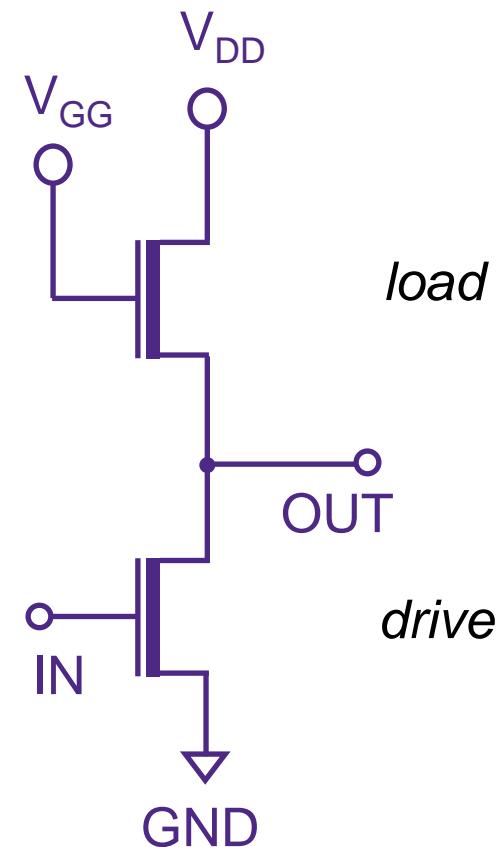


Inverter realizations

Switch = n channel
MOSFET: *normally OFF*
device

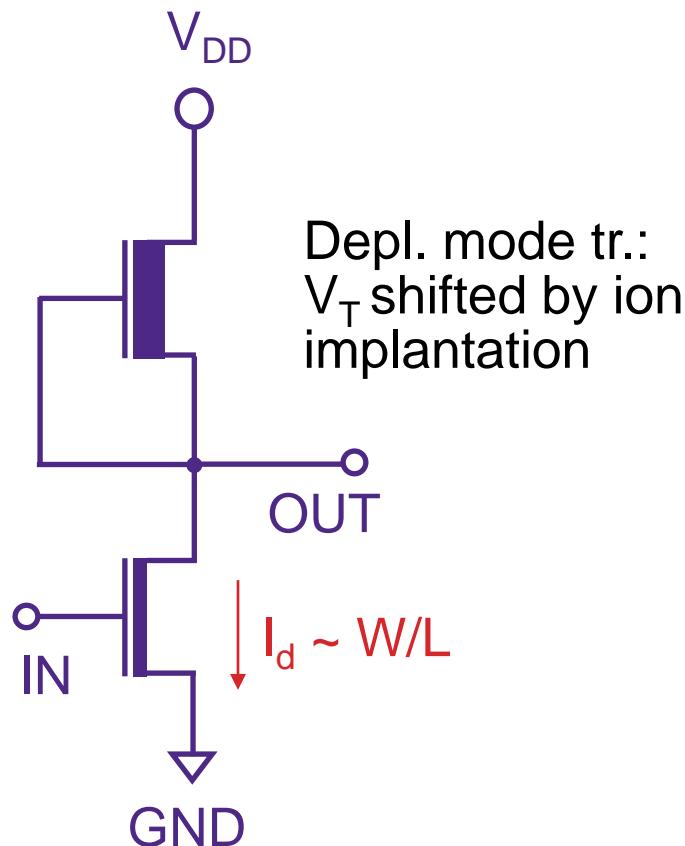


Load resistor: another
transistor, e.g. in triode
region



Needs another supply – **not OK**

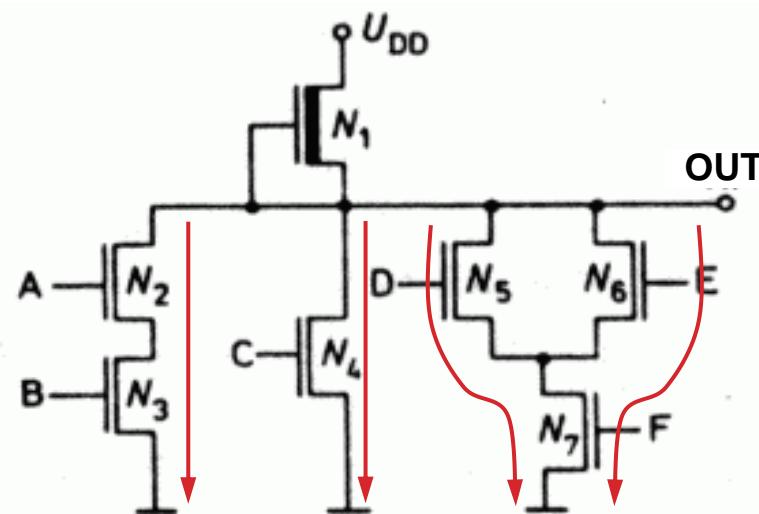
nMOS technique – very simple



- Simple process, outdated, many disadvantages
 - static consumption if OUT=0
 - if OUT = 0, it will not be a pure GND level
 - asymmetrical transfer characteristic (see later)
- In both cases the *load resistor* is replaced by a MOSFET but this transistor was not provided with an active control
 - *This is the passive load inverter*

Complex gates (in nMOS)

- Serial conduction paths in parallel, e.g.:



There are 4 paths

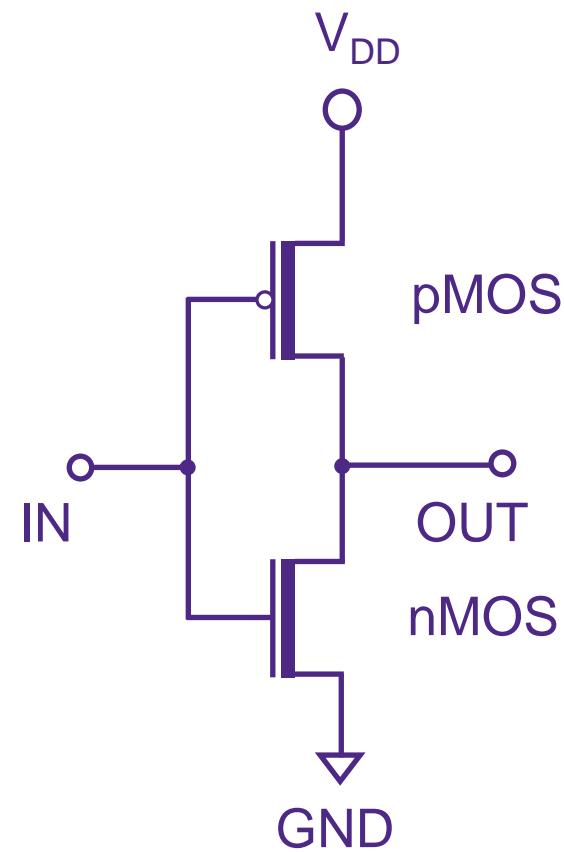
$$OUT = \overline{AB + C + (D + E)F}$$

The CMOS technique

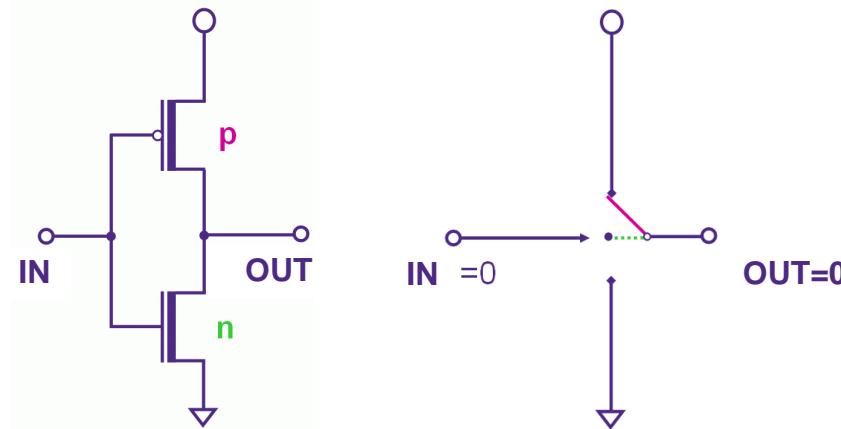
- The name comes from: Complementary **MOS**
- Idea: the load also should be provided with active control
 - if the nMOS *driver* (switching) transistor conducts, **then** the *load* transistor must be an "open" circuit
 - if the nMOS *driver* (switching) transistor is an "open circuit", **then** the *load* must be conducting
- This needs such a *normally OFF device* which needs "opposite" control signals than the nMOS transistors
 - Such device is an enhancement mode **pMOS transistor**

The CMOS inverter

- An n and a p type enhancement mode device
- Active load inverter: the two transistors have the same common control



In steady state only one device is "on", the other is "off".

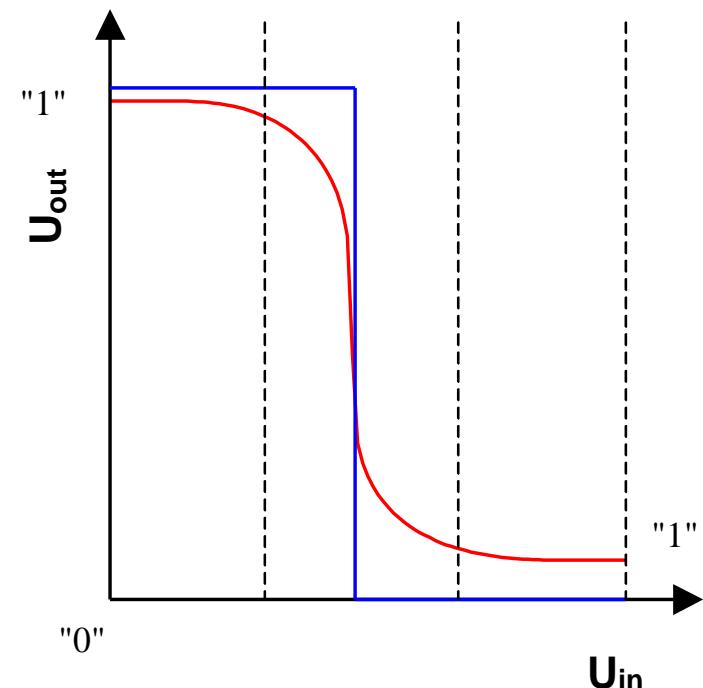


Characteristics of inverters, rudiments

- Transfer characteristic:
 - output voltage vs. input voltage

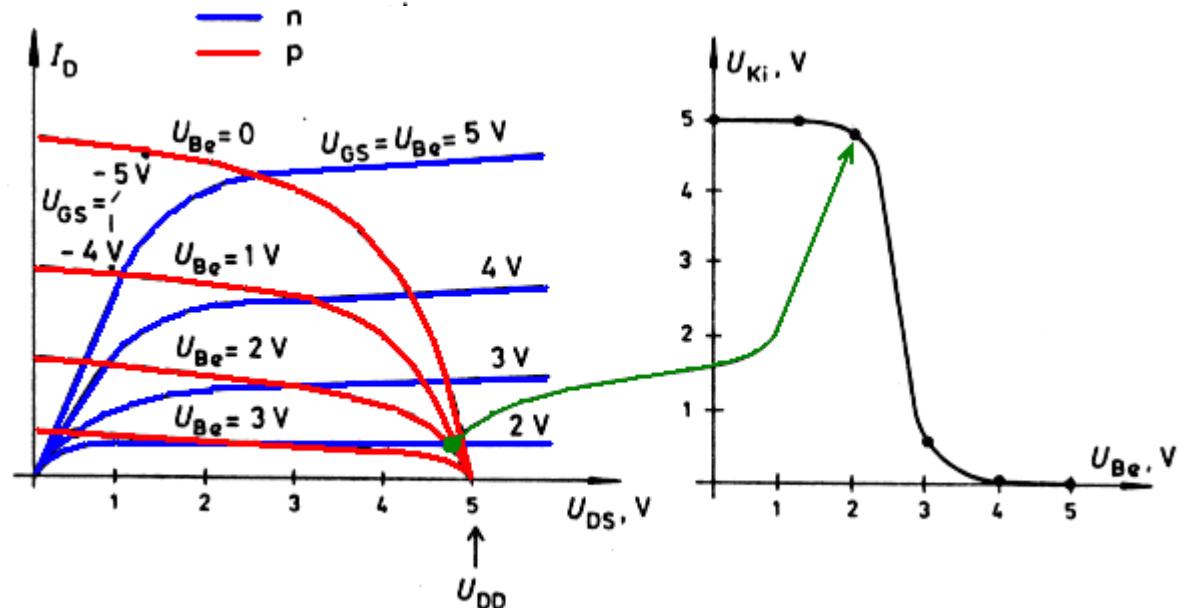
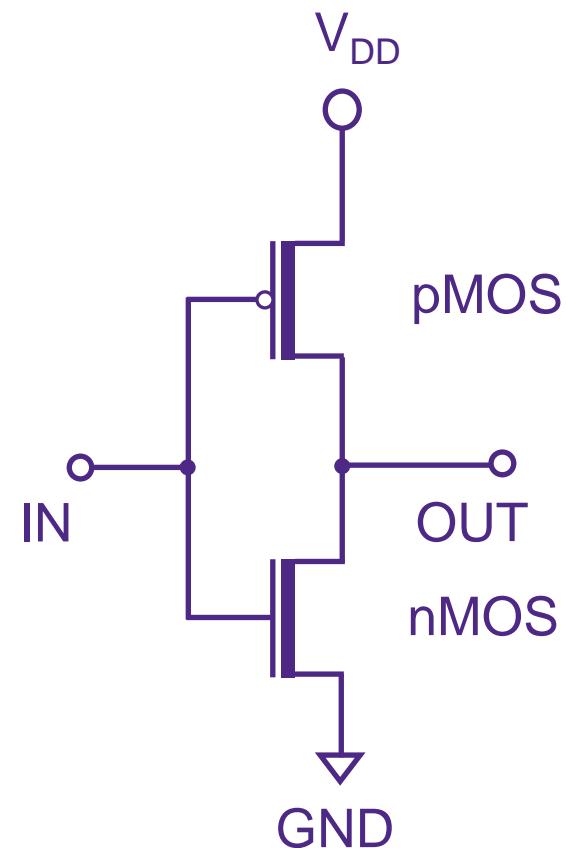
$$U_{out} = f(U_{in})$$

The output signal is the inverted version of the logic value of the input signal



transfer characteristic of an ideal and a realistic inverter

Xfer char. of a CMOS inverter



$$U_{IN} = U_{GSn}$$

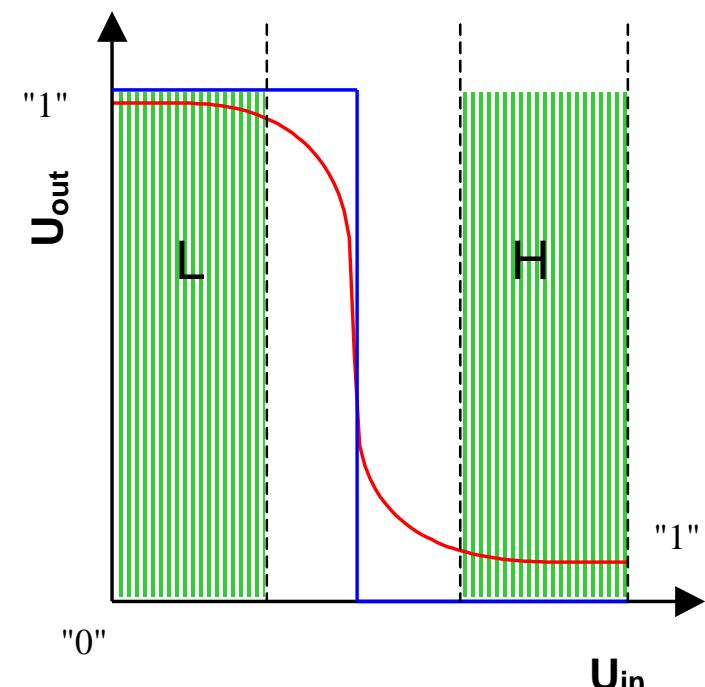
$$U_{OUT} = U_{DSn}$$

Characteristics of inverters, rudiments

- Noise immunity:

- Same U_{out} corresponds to a wide U_{in} range
- There are 3 regions in the characteristic
- On the L and H sides the characteristic is flat, i.e. any voltage change in the input has negligible effect on the output.

L and H regions

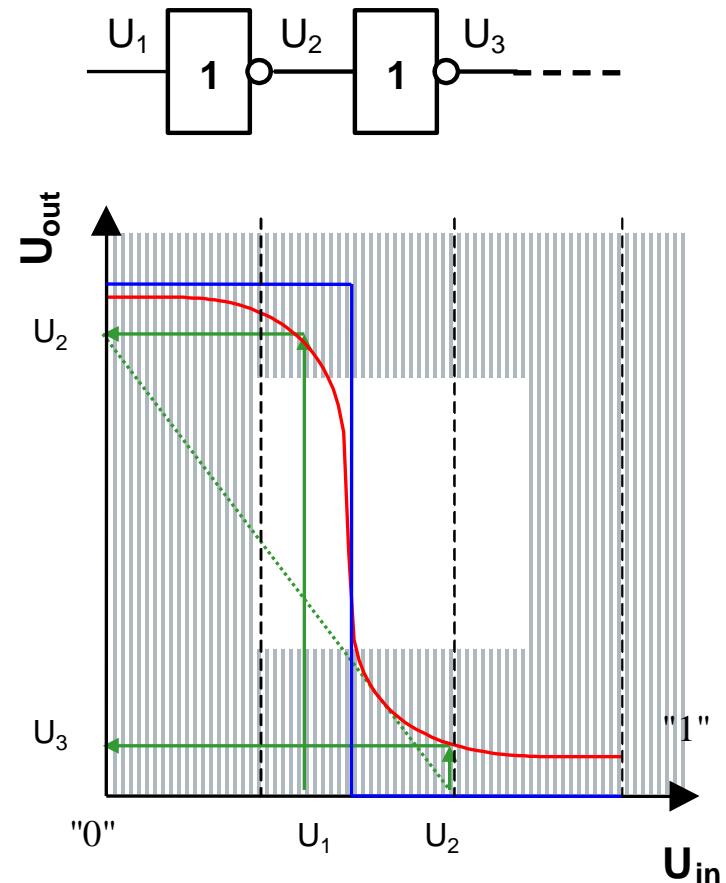


transfer characteristic of an ideal and a realistic inverter

Characteristics of inverters, rudiments

- Signal regeneration
 - depends on the slope of the middle region

U_1 is a "bad" logic 0 signal.
 Output U_2 of the first inverter is already close to an acceptable logic 1 level.
 output voltage U_3 at the second inverter is already a "good" logic 0 level.

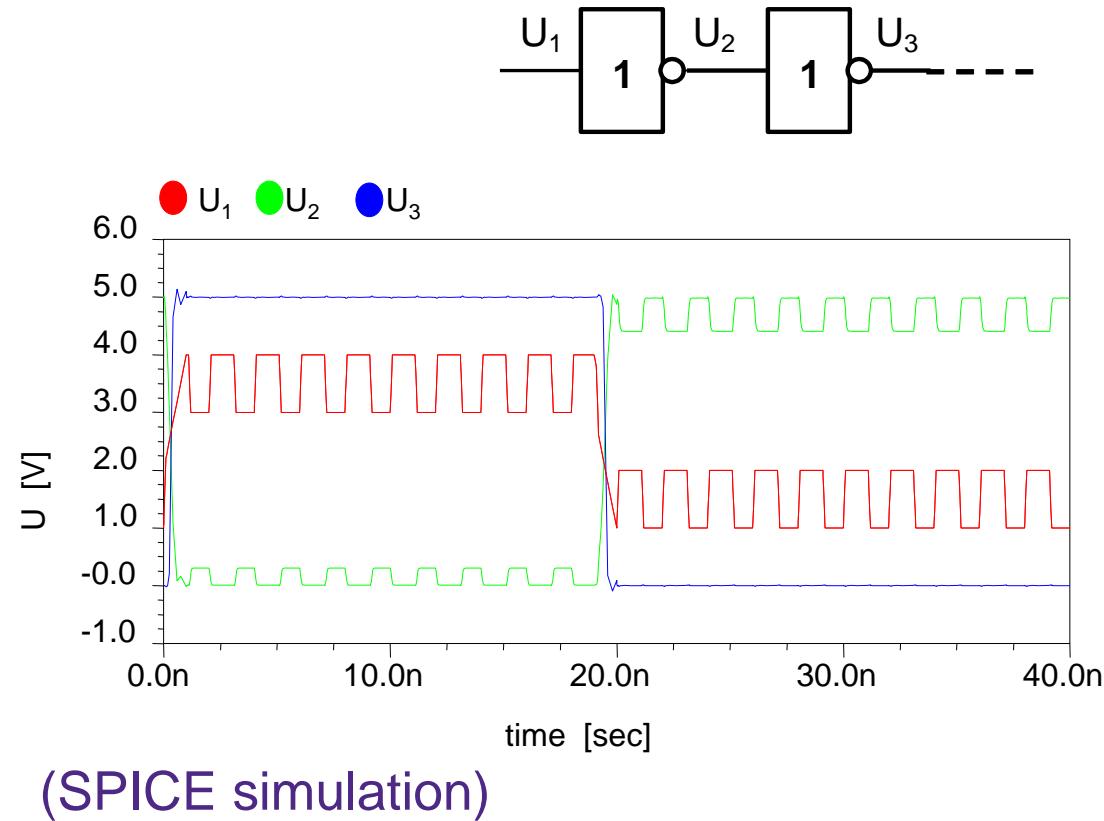


transfer characteristic of an ideal and a realistic inverter

Characteristics of inverters, rudiments

- Signal regeneration

$U_L=0V, U_H=5V$



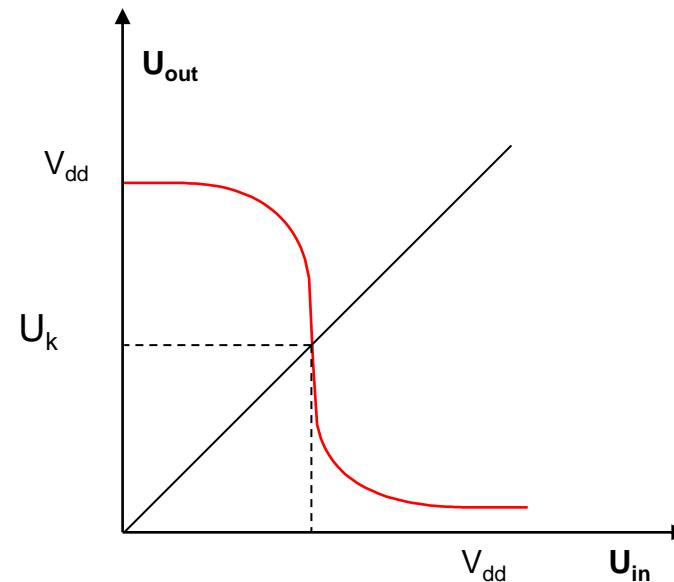
In case of U_3 both the voltage level and the signal form are visibly regenerated!

Characteristics of inverters, rudiments

- Inverter logic threshold voltage

The level, under which the signals will be converted into logical 0 and above which the signals will be converted by the inverter chain into logical 1

Intersection of the $U_{in}=U_{out}$ line and the x-fer characteristic

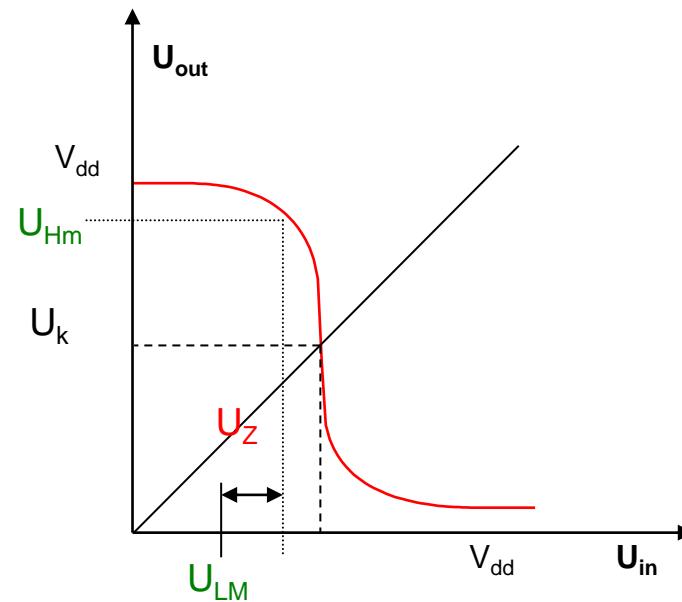


Characteristics of inverters, rudiments

- Logic level ranges

The voltage range of the logic 0 and 1 values **within which the circuit works safely in the respective logic level**

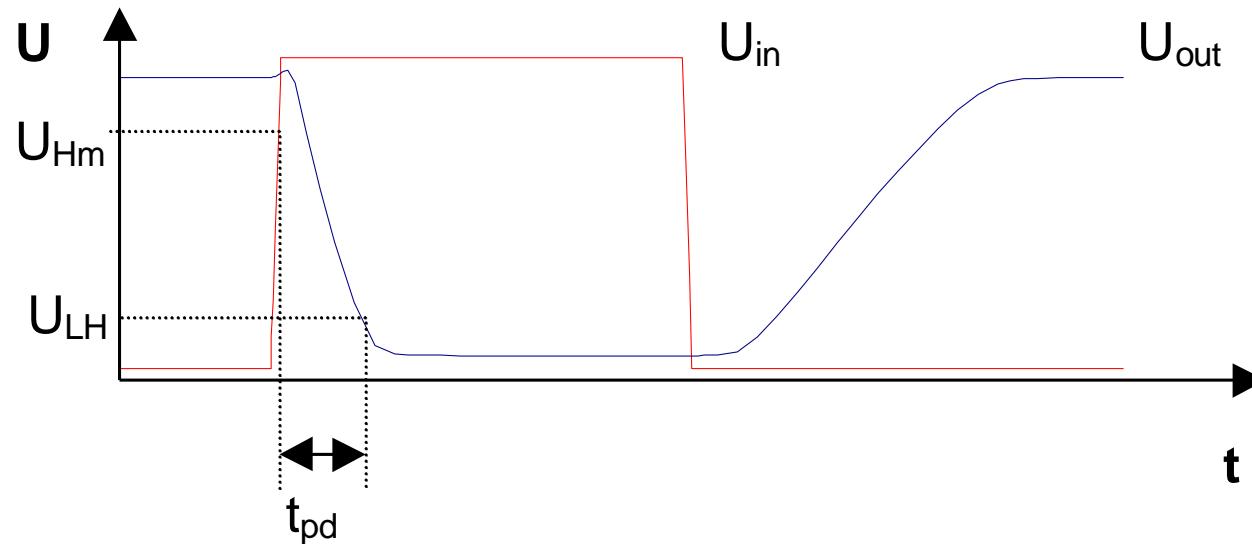
Example: 74HC00,
 $V_{dd}=3V$,
 $U_{LM}=0.9V$
 $U_{Hm}=2.1V$



Important voltage values
 U_{LM} , max. of logic 0
 U_{Hm} , min. op logic 1

Characteristics of inverters, rudiments

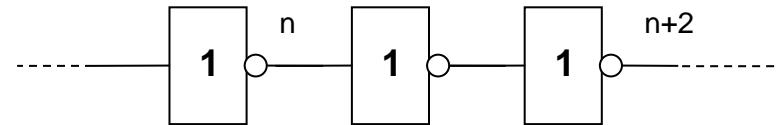
- Propagation delay



t_{pd} is difficult to define, and may be different for switching on and off (e.g. nMOS inverters)

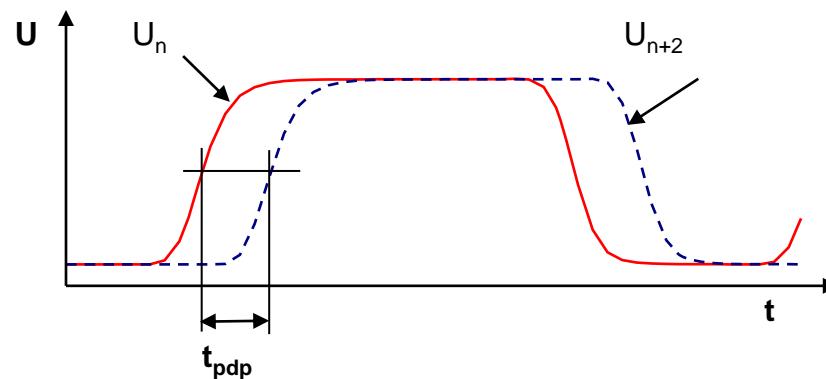
Characteristics of inverters, rudiments

- Inverter pair delay



A long chain of uniform inverters is assumed. After a certain number of inverters the signal form will be determined by the inverter properties only.

After propagating through 2 inverters the signal will be the same, the delay will be $t_{pdः}$ – *the inverter pair propagation delay*

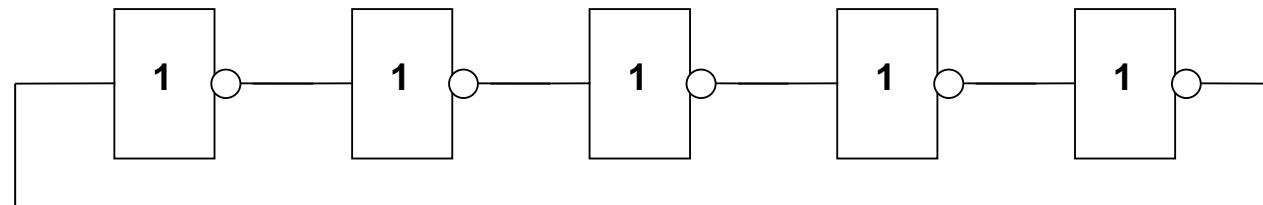


Characteristics of inverters, rudiments

- Measuring the inverter pair delay

THE RING OSCILLATOR

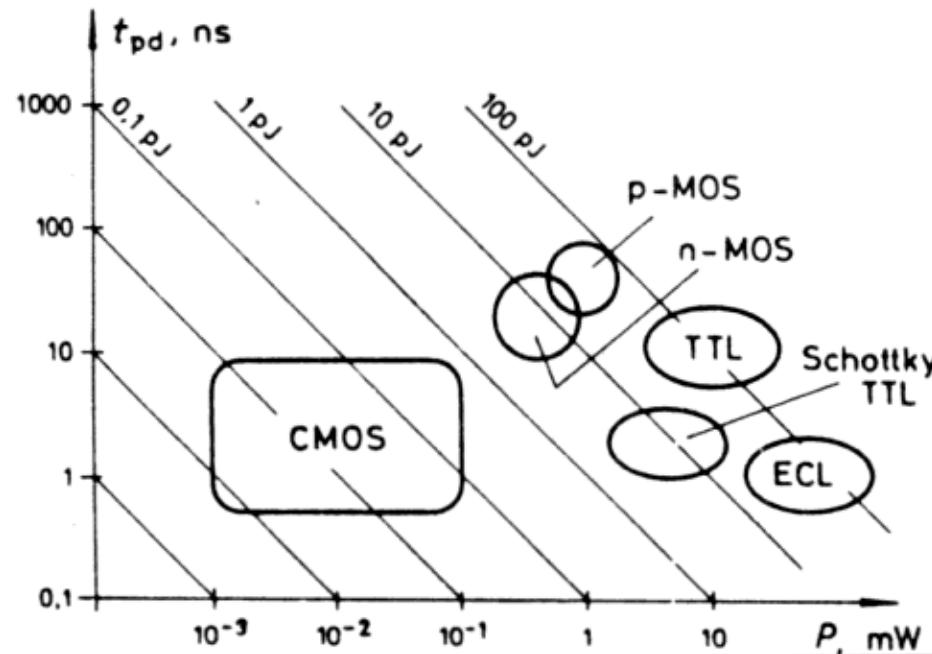
Odd number of inverters connected in a chain, no stable state \Rightarrow oscillation



$$T = n \cdot t_{pd}$$

Characteristics of inverters, rudiments

- Power-delay product ($P\tau$)
 - **low power and small delay** refer to good quality,
 - their product is a **figure of merit** for the quality of a circuit family.
 - the physical meaning: the minimal energy, needed to work on 1 bit of information.



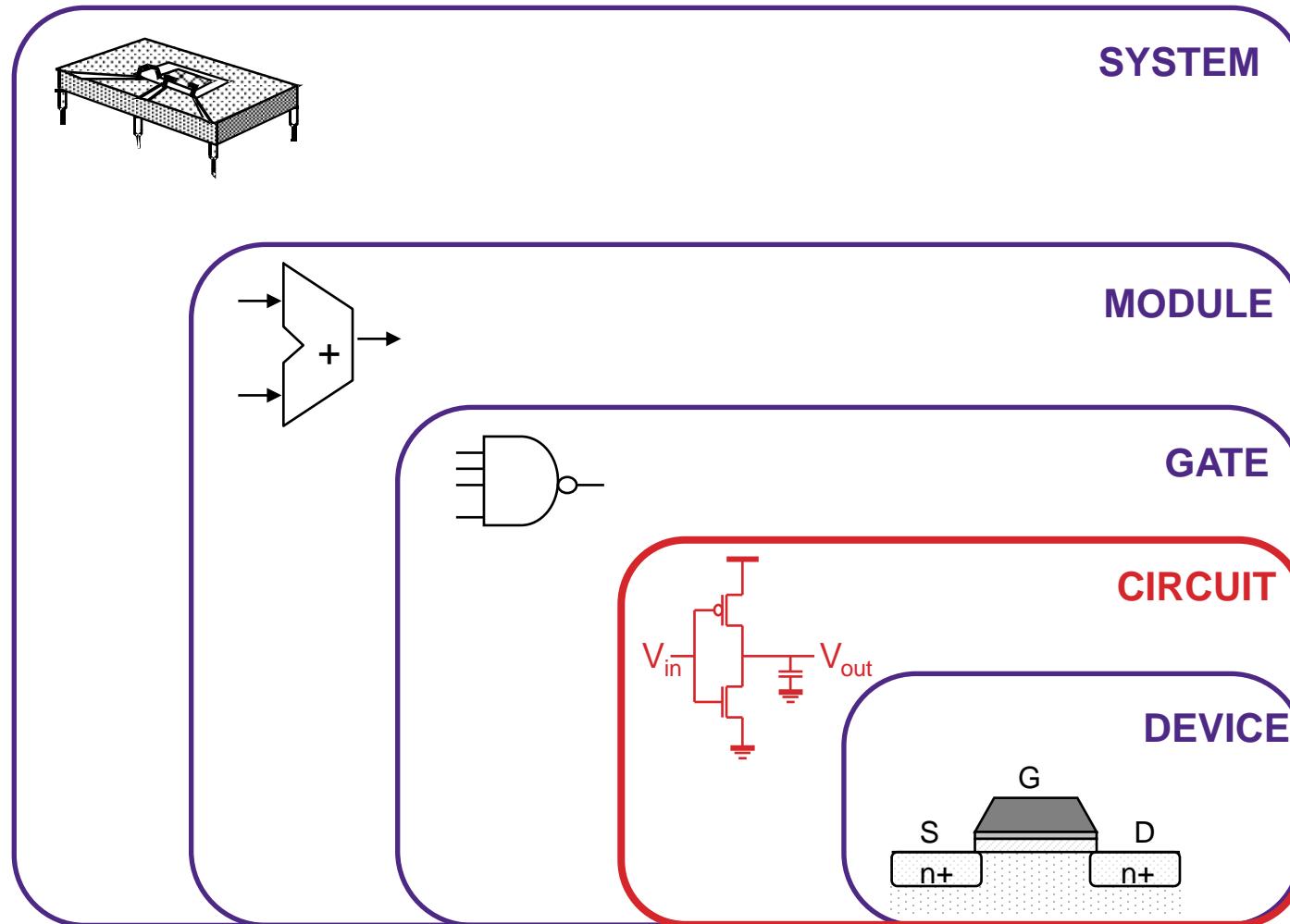


Budapest University of Technology and Economics
Department of Electron Devices

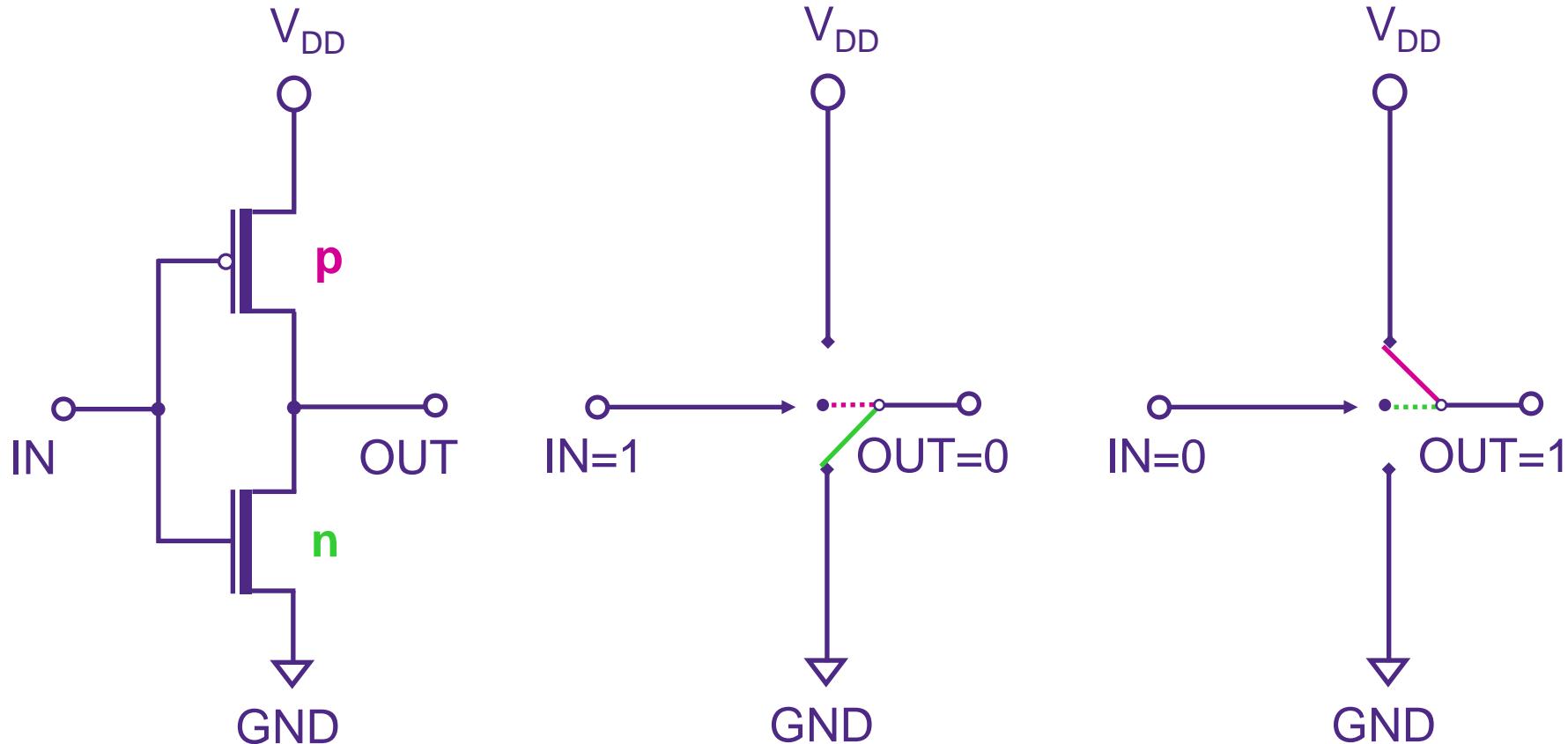
Microelectronics, BSc course

**nMOS/CMOS Logic Gates II:
Schematic, layout, x-section, std
cells**

The abstraction level of our study:

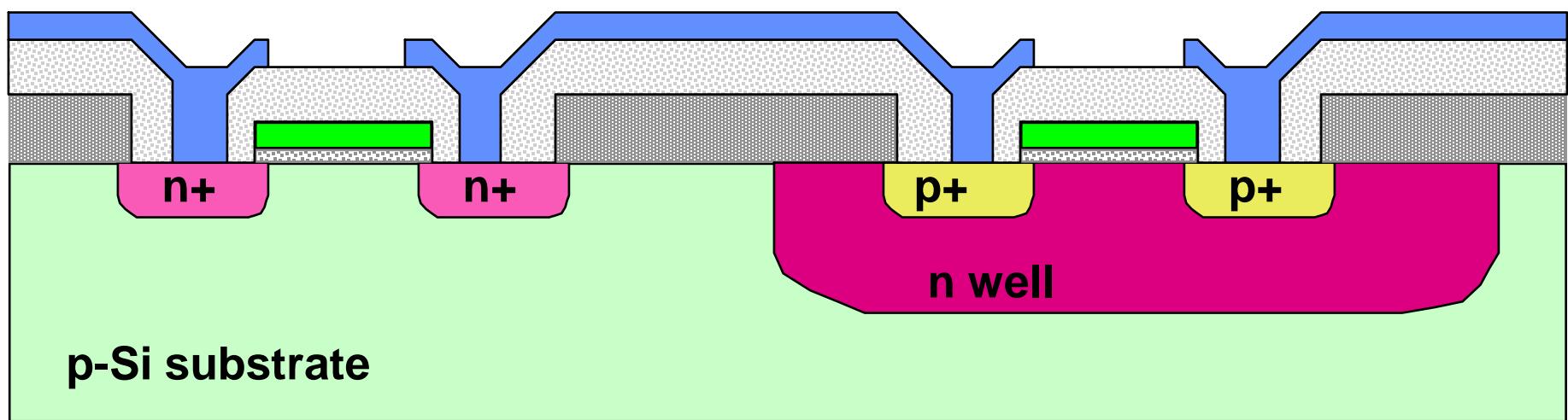


The CMOS inverter – recall



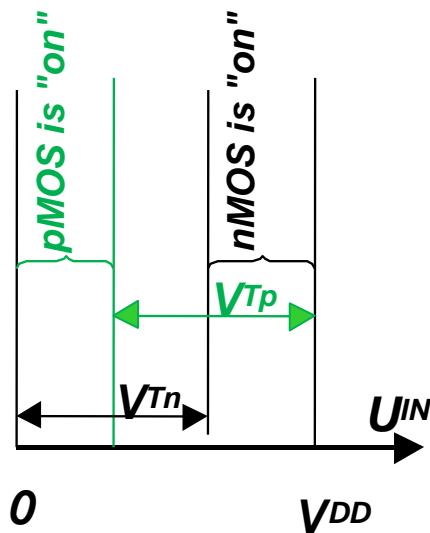
In steady-state only one transistor is "on", the other one is always "off"

X-sectional view of a CMOS inverter



Characteristic of the CMOS inverter

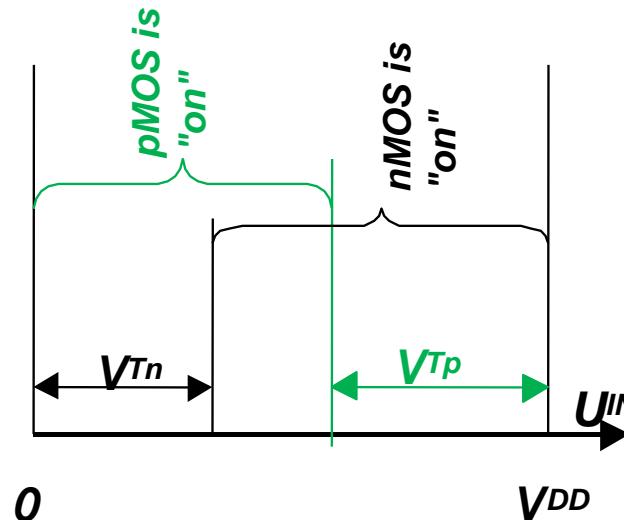
*2 basic cases, depending on the **supply voltage** and threshold voltages of the transistors*



1. small supply voltage:

$$V_{DD} < V_{Tn} + |V_{Tp}|$$

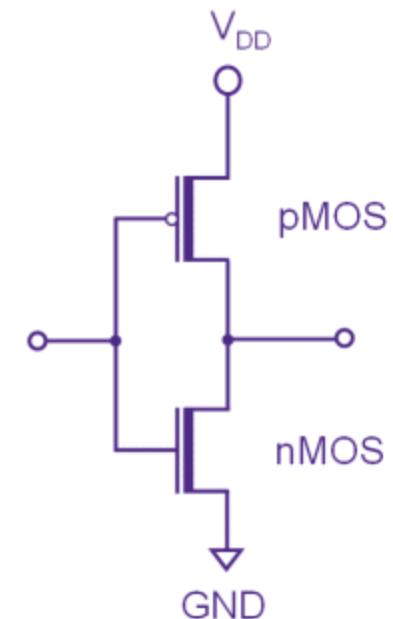
only one transistor is "on" at a time



2. larger supply voltage:

$$V_{DD} > V_{Tn} + |V_{Tp}|$$

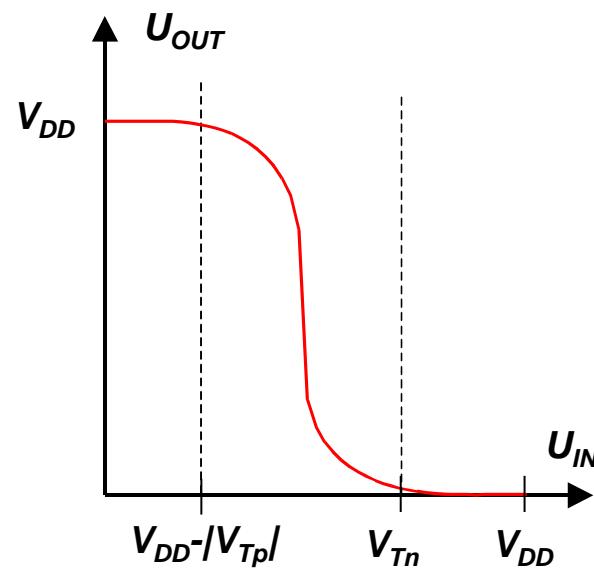
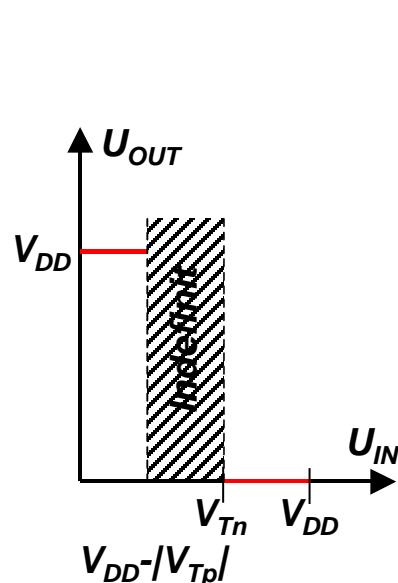
when switching over, both transistors are "on" at the same time



Characteristic of the CMOS inverter

- 1. small supply voltage: $V_{DD} < V_{Tn} + |V_{Tp}|$

the characteristics: $U_{OUT} = \begin{cases} V_{DD} & \text{if } U_{IN} < V_{DD} - |V_{Tp}| \\ \text{indefinit} & \text{if } V_{Tn} < U_{IN} < V_{DD} - |V_{Tp}| \\ 0 & \text{if } U_{IN} > V_{Tn} \end{cases}$



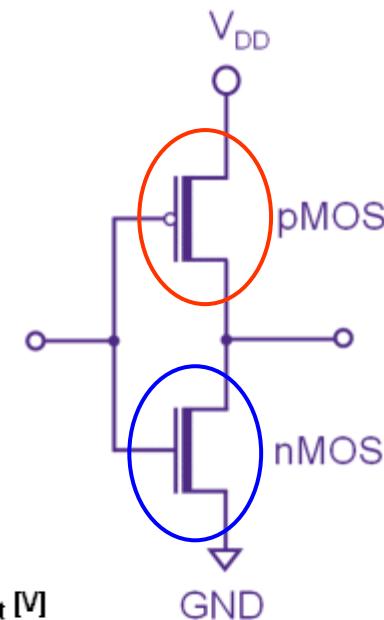
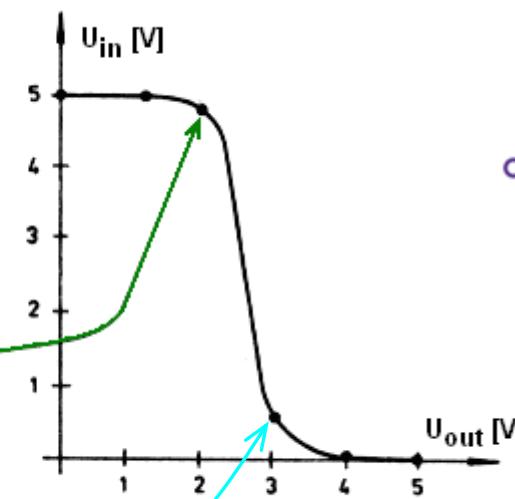
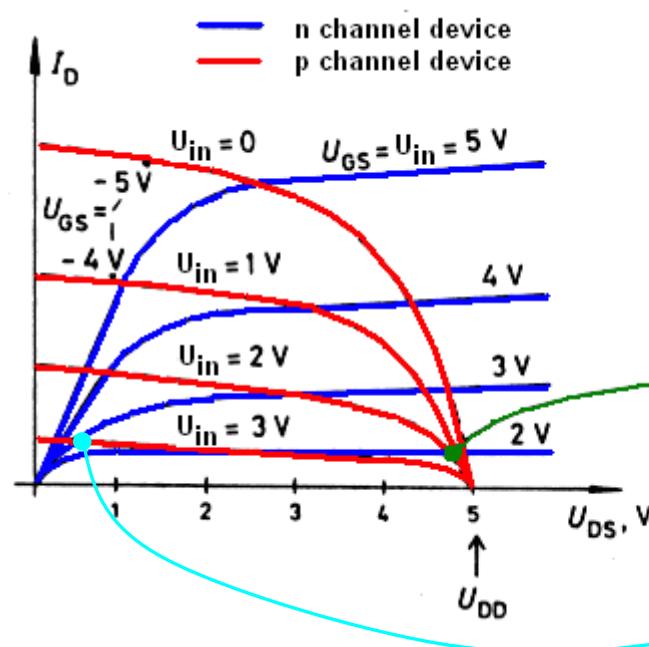
The middle part of the transfer characteristic is very steep, this is the specific advantage of CMOS inverters.

Characteristic of the CMOS inverter

- 2. large supply voltage: $V_{DD} > V_{Tn} + |V_{Tp}|$

Switching over? - "mutual conduction"

- Constructing the characteristic:



The CMOS inverter

Design for symmetrical operation:

If $U_{IN}=U_{inv}$ logic threshold voltage, both transistors have equal current:

$$K_n(U_{inv} - V_{Tn})^2 = K_p(U_{DD} - U_{inv} - |V_{Tp}|)^2$$

$$U_{inv} = \frac{U_{DD} - |V_{Tp}| + V_{Tn}\sqrt{K_n / K_p}}{1 + \sqrt{K_n / K_p}}$$

$$K_x = \left(\frac{W}{L} \right)_x \frac{\mu_x C_{ox}}{2}$$

$$U_{GSn} = U_K$$

$$U_{GSp} = V_{DD} - U_K$$

The inverter logic threshold voltage depends on the ratio of the current constants of the transistors.

To have U_{inv} at $V_{DD}/2$ and $|V_{Tn}|=|V_{Tp}|$, then $K_n=K_p$ has to be set.

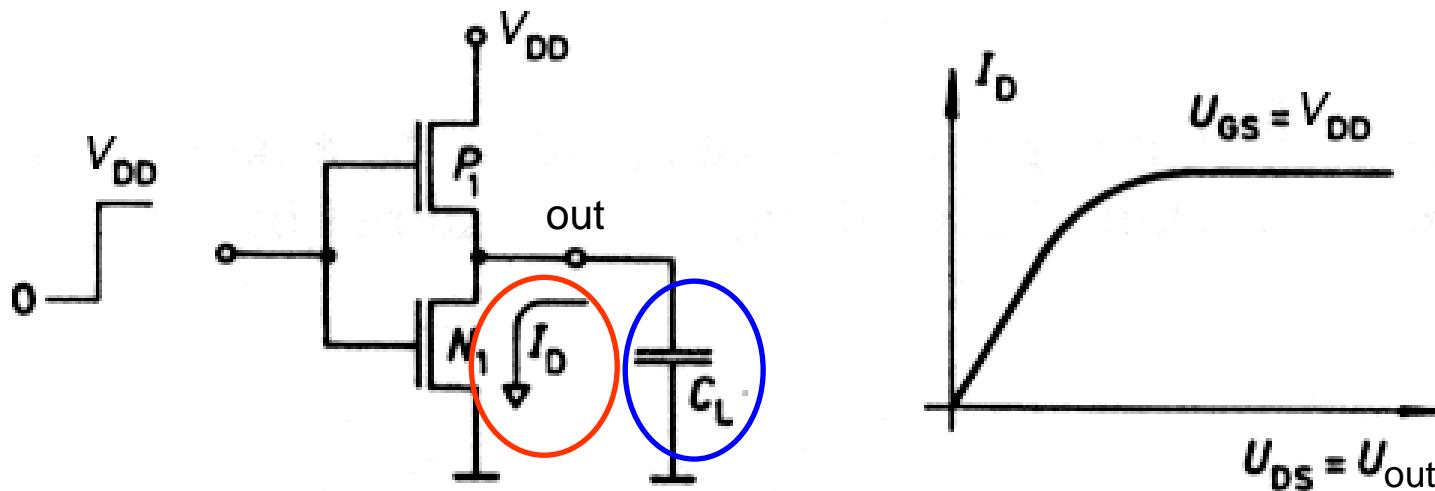
$$\left(\frac{W}{L} \right)_P = 2..2.5 \times \left(\frac{W}{L} \right)_n$$

since hole mobility is 2 ... 2.5 times less

The logic threshold voltage can be set by the W/L ratios

The CMOS inverter / dynamic char.

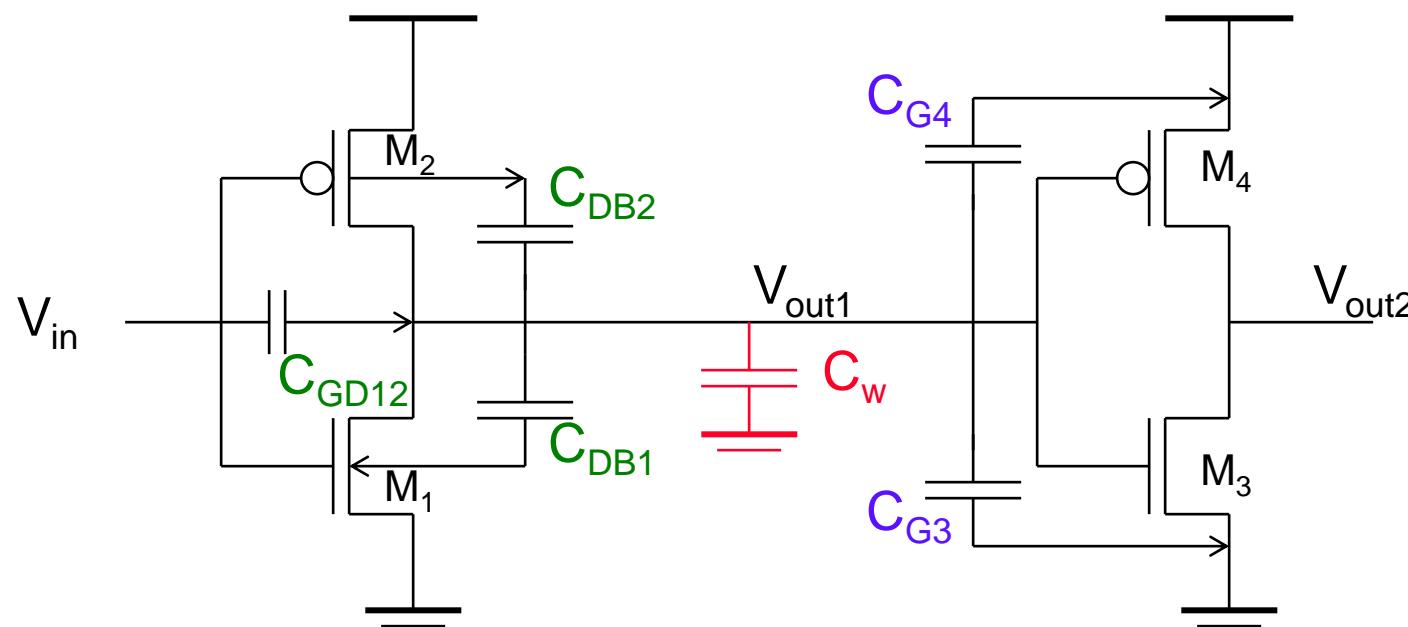
- Calculation of the switching times
 - What do they depend on?
 - the current driving capability of the output
 - the capacitive load on the output



- If the characteristics of the two transistors are exactly complementary ($K_n = K_p$ and $V_{Tn} = |V_{Tp}|$), rising and falling times will be equal

The capacitances

- Intrinsic capacitances of the driving stage
- Input capacitance of the loading stage (next gate) – extrinsic or fanout capacitances
- wiring (interconnect) capacitance



intrinsic MOS transistor capacitances

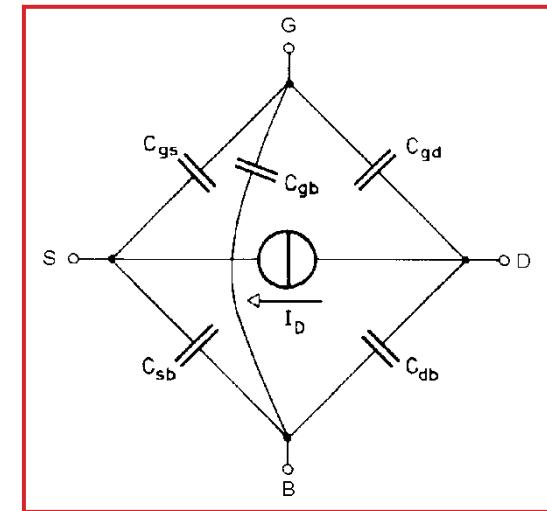
extrinsic MOS transistor (fanout) capacitances

wiring (interconnect) capacitance

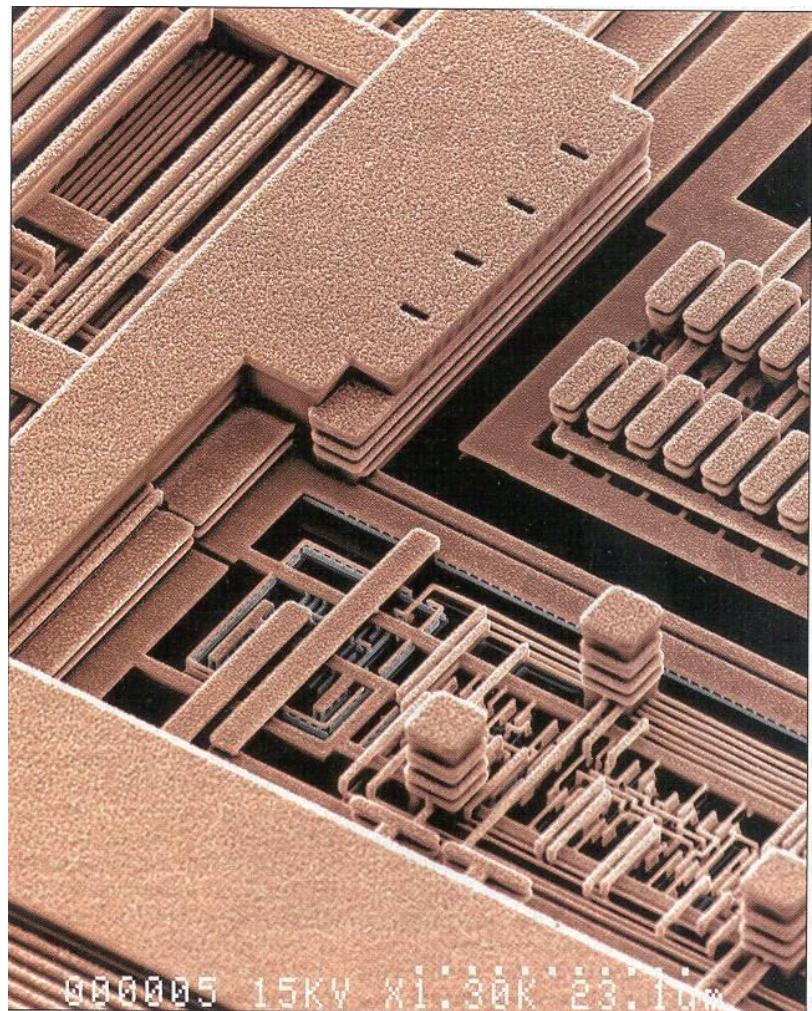
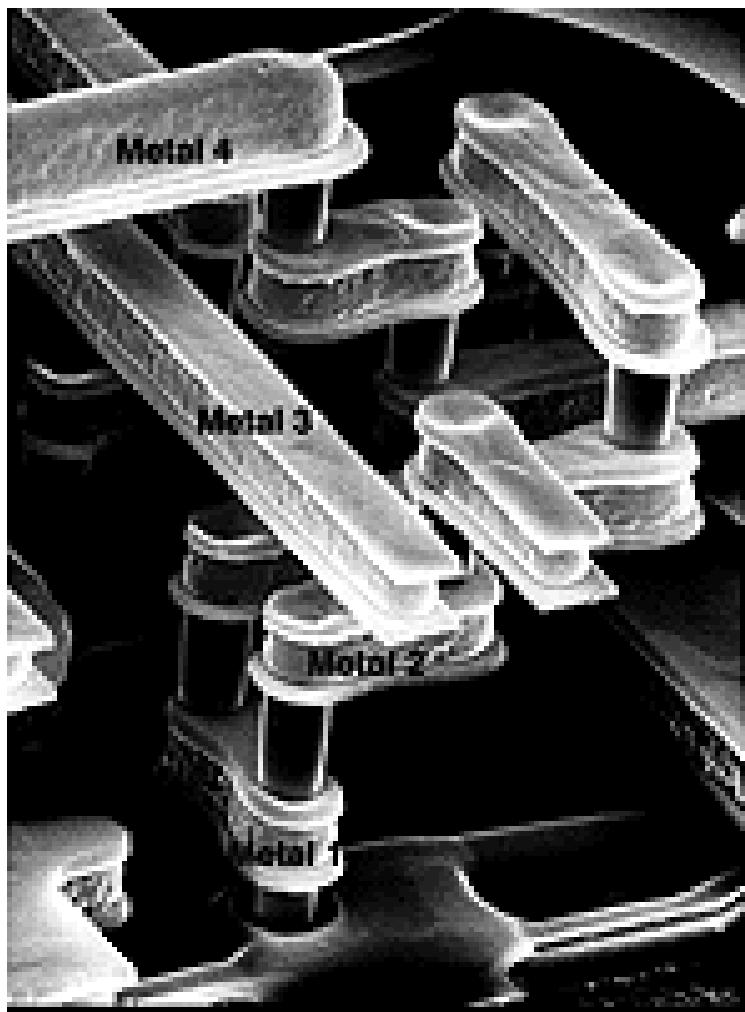
The capacitances

- The intrinsic capacitances:
 - S-G G-D overlap capacitances
 - the MOS capacitance of the channel
 - capacitances of pn junctions

- The wiring capacitance
 - depends on the interconnect geometry (width, length)
 - with the advance of manufacturing processes this capacitance tends to increase



Modern metallization



Intel 0.25 μm process

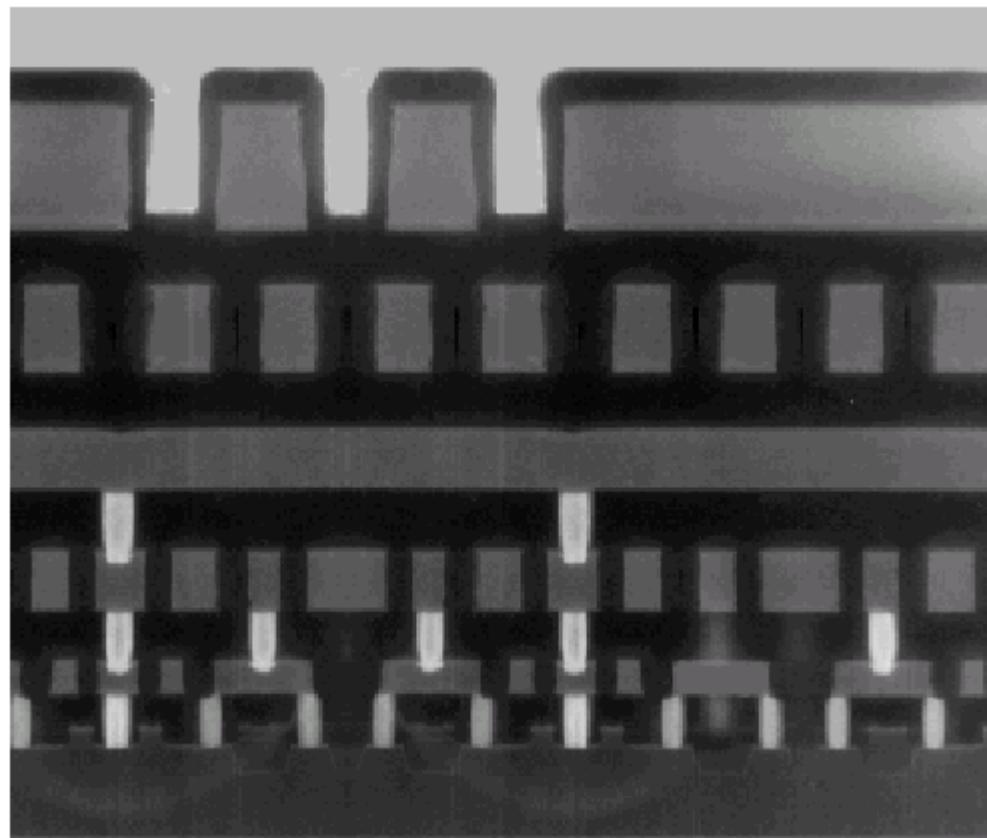
5 metal layers

Ti/Al - Cu/Ti/TiN

Polysilicon dielectric

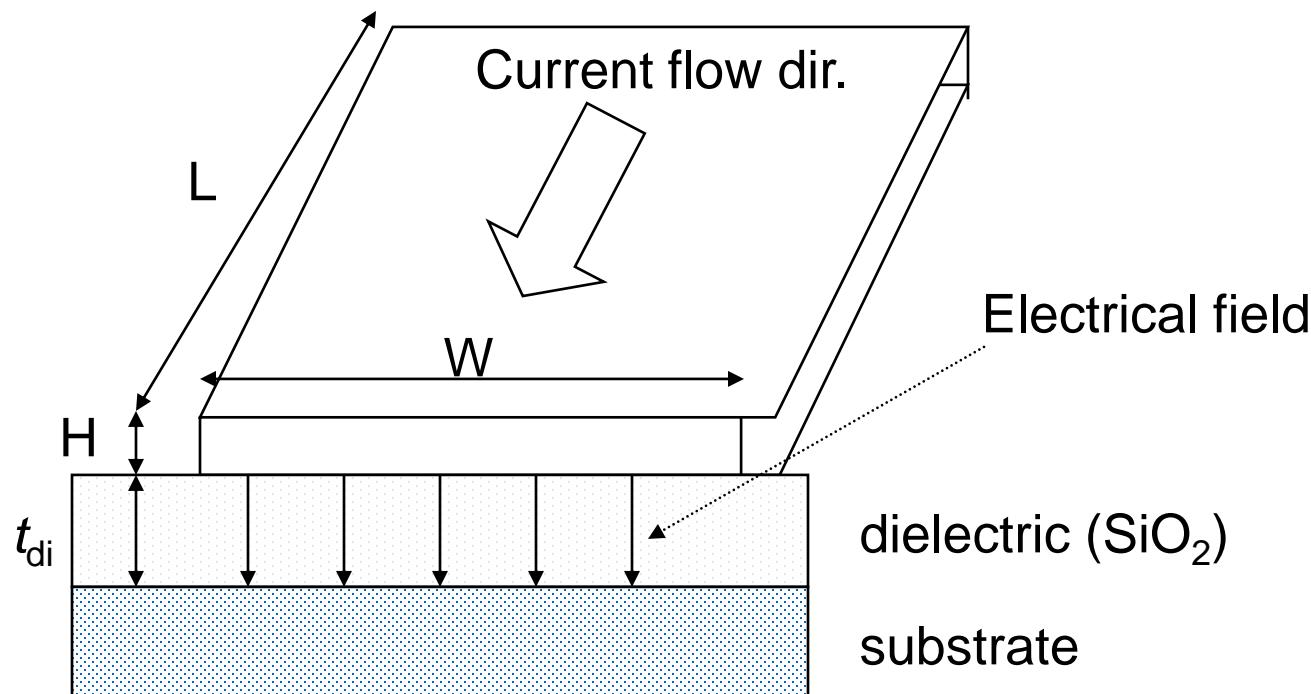
LAYER	PITCH	THICK	A.R.
Isolation	0.67	0.40	-
Polysilicon	0.64	0.25	-
Metal 1	0.64	0.48	1.5
Metal 2	0.93	0.90	1.9
Metal 3	0.93	0.90	1.9
Metal 4	1.60	1.33	1.7
Metal 5	2.56	1.90	1.5
	μm	μm	

Layer pitch, thickness and aspect ratio



Interconnect capacitances

Interconnect - substrate: *parallel plate capacitance*

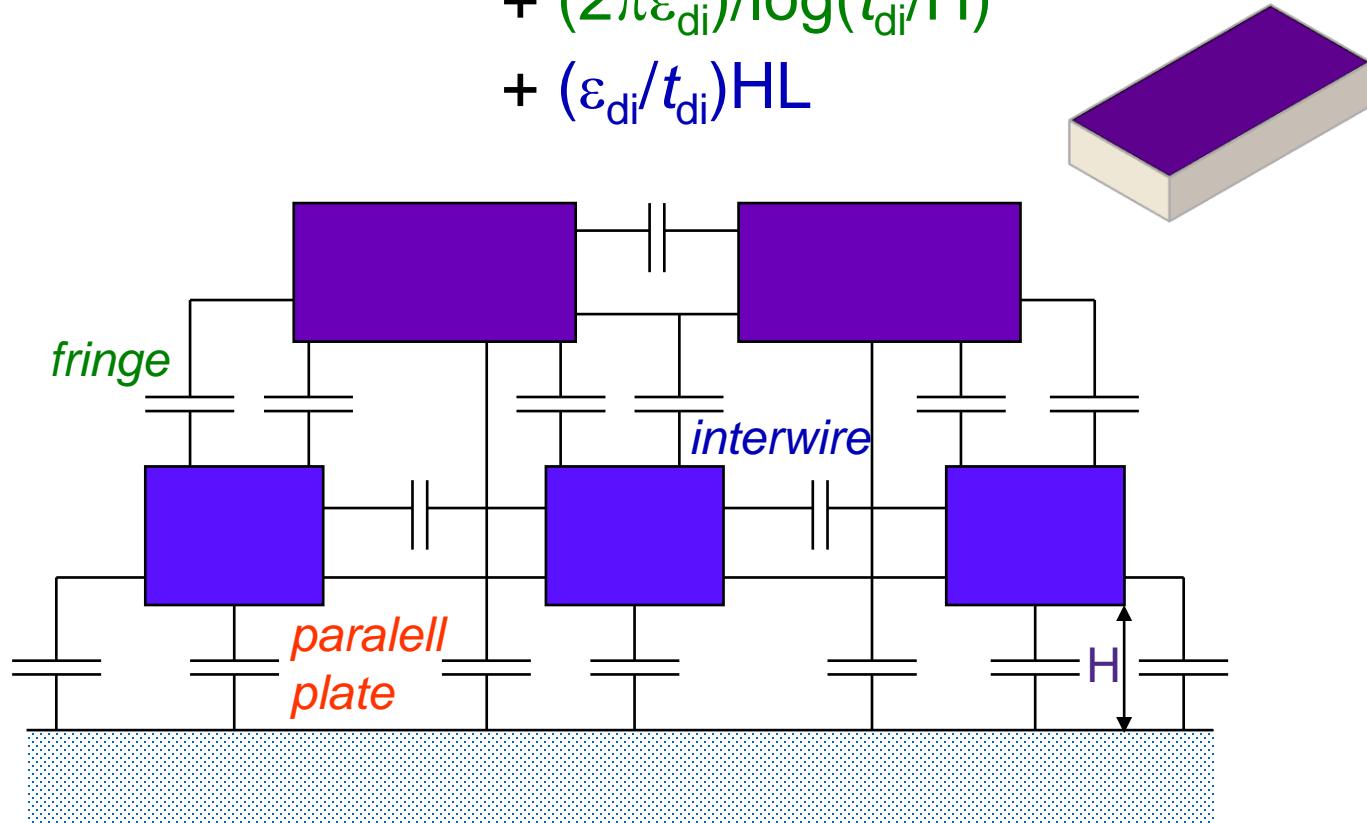


Dielectric
constant
($\text{SiO}_2 \Rightarrow 3.9$)

$$C_{pp} = (\epsilon_{di}/t_{di}) WL$$

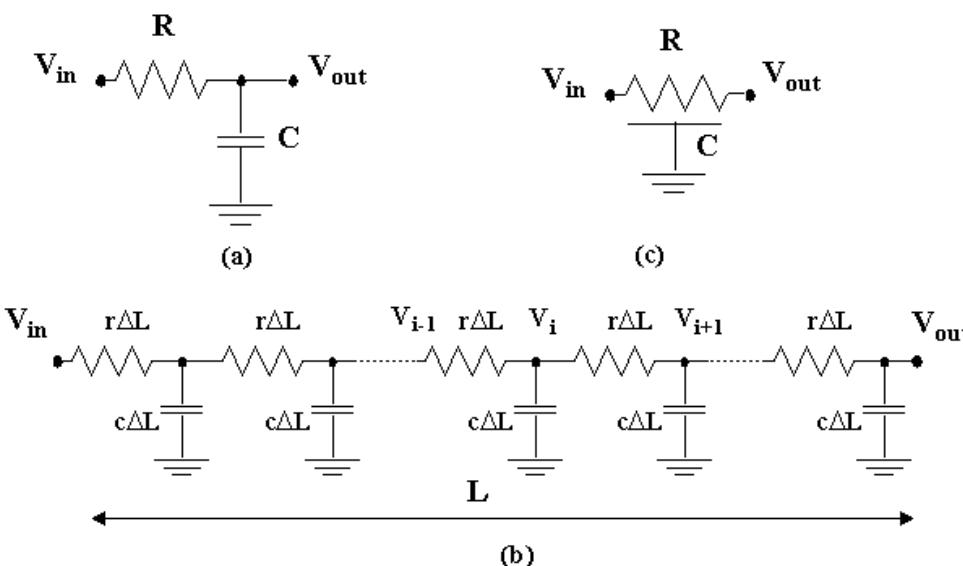
Interconnect capacitances

$$\begin{aligned}
 C_{\text{wire}} &= C_{\text{pp}} + C_{\text{fringe}} + C_{\text{interwire}} \\
 &= (\epsilon_{\text{di}}/t_{\text{di}})WL \\
 &\quad + (2\pi\epsilon_{\text{di}})/\log(t_{\text{di}}/H) \\
 &\quad + (\epsilon_{\text{di}}/t_{\text{di}})HL
 \end{aligned}$$



Other issues of interconnects

- Series resistance
- Distributed parameter RC line (see transmission lines)



Sort of a representation of the diffusion equation

$$rc \frac{\partial^2 V}{\partial t} = \frac{\partial^2 V}{\partial x^2}$$

$$\tau(V_{out}) = \frac{rc L^2}{2}$$

The CMOS inverter / dynamic char.

- Calculation of switching times
 - identical times, integration for the extreme values of the voltage of the load capacitance:

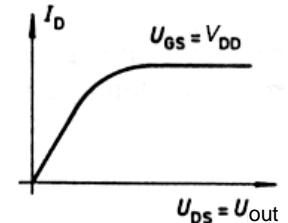
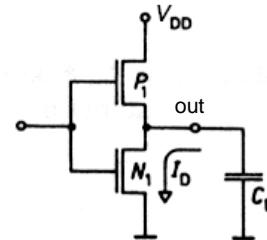
$$t_l = \int_{V_{DD}}^{V_{LM}} \frac{C_L}{I_D} dU$$

- If

$$I_D \approx K(V_{DD} - V_T)^2$$

then

$$t_l = \frac{C_L(V_{DD} - V_{LM})}{K(V_{DD} - V_T)^2}$$



V_{LM} – minimal voltage of the load capacitance

Can be reduced by increasing the supply voltage or the W/L ratio

Power consumption of CMOS inv.

- There is no static consumption since there is no static current
- There is dynamic consumption during switching which consists of 2 parts:
 - Mutual conduction:
 - During the rise of the input voltage both transistors are "on"

$$V_{Th} < U_{IN} < V_{DD} - V_{Tp}$$

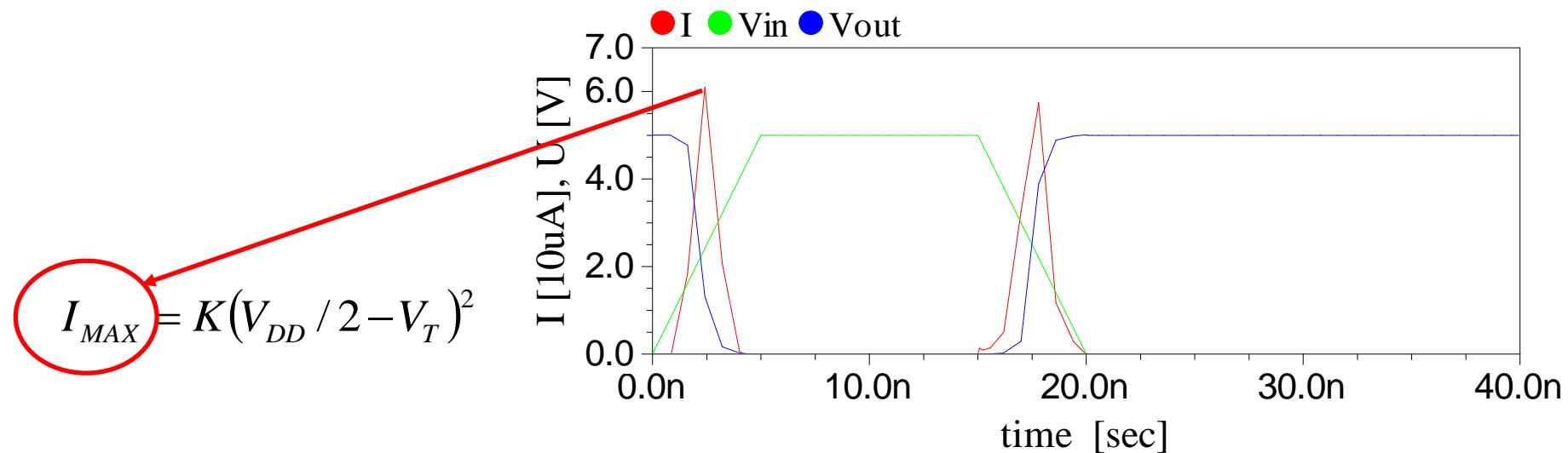
- Charge pumping:
 - At switching over the output to 1 the $C_{L \text{ loading}}$ capacitor is charged to the supply voltage through the p transistor, then it is discharged towards the ground through the n transistor.

Charge is pumped from VDD to GND.

Power consumption of CMOS inv.:

- Mutual conduction ("short power"):**

- During a certain period of the rise of the input signal both transistors are "on" if $V_{Tn} < U_{IN} < V_{DD} - V_{Tp}$ this is called mutual conduction



- charge flowing through: $\Delta Q = b t_{UD} I_{MAX}$, where t_{UD} is the time while current is flowing, b is a constant depending on the signal shape. $b \approx 0.1-0.2$*

$$P = f \Delta Q V_{DD} = f V_{DD} b t_{UD} K (V_{DD} / 2 - V_T)^2$$

$$P \sim f V_{DD}^3$$

Power consumption of CMOS inv.:

- **Charge pumping:**

- At switching the C_L load capacitance is charged to VDD through the p-channel device when the output changes to 1, later, when switching the output to 0, it is discharged towards GND through the n-channel device.

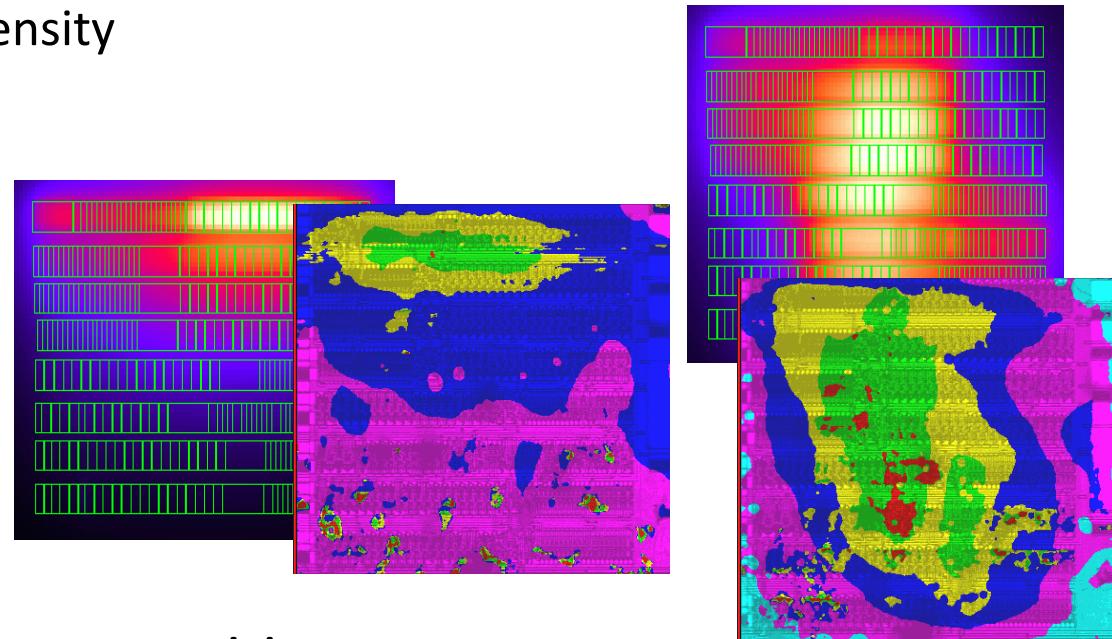
$$\Delta Q_L = C_L V_{DD}$$

$$P_{cp} = f C_L V_{DD}^2$$

- The power consumption due to charge pumping is proportional to the frequency and the square of the supply voltage.
- ▶ **Total consumption:** sum of the two components (if there is mutual conduction), directly proportional to the frequency and the 2nd and 3rd power of the supply voltage.

Components of the consumption of CMOS circuits

- Dynamic components – at every switching event
 - mutual conduction, charge pumping
 - proportional to the event density
 - clock frequency
 - circuit activity



- ▶ Further components due to parasitics:
 - subthreshold currents
 - leakage currents of pn junctions – nowadays already significant
 - leakage (tunneling) through the a gate dielectric

Construction

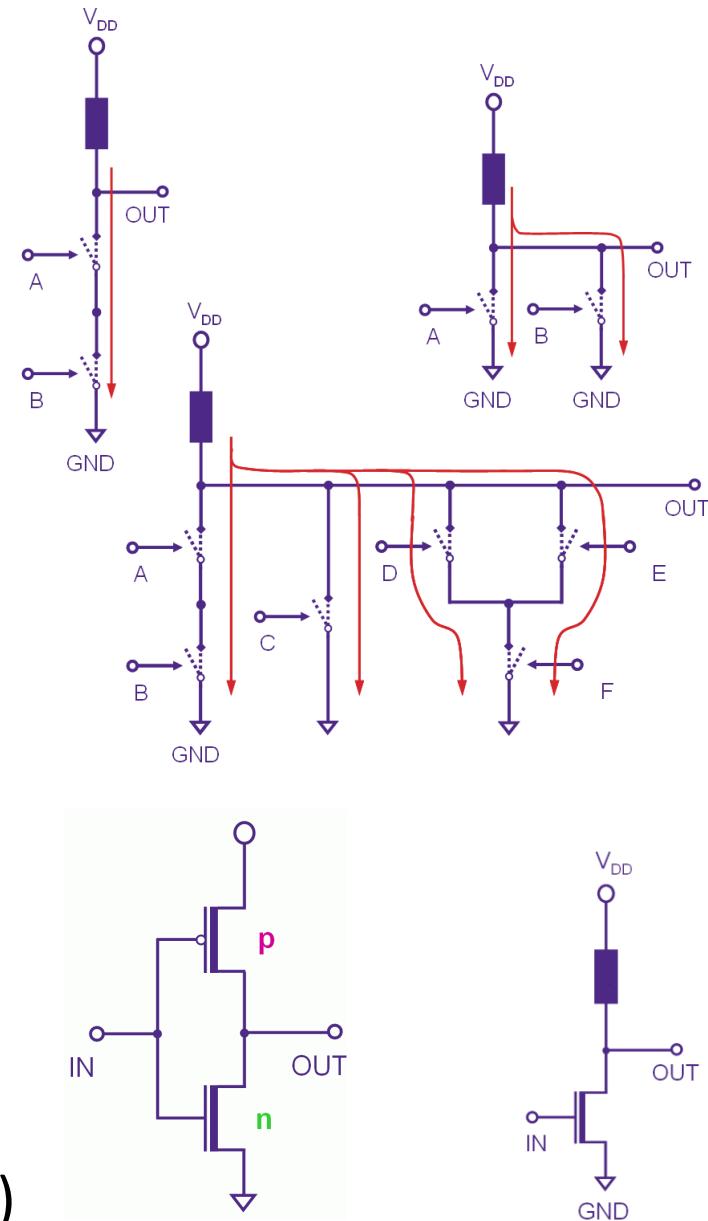
- ▶ Constructing CMOS gates on schematic (circuit) level
- ▶ Process and layout
 - ▶ Recall the nMOS process and layout
 - ▶ Stick diagram layout
 - ▶ Full layout
 - ▶ Create standard cells

CMOS gates

- **PDN:** Create an nMOS switching circuit (pull down network):
 - series path: NAND function
 - parallel path: NOR function
 - combination of these: complex gate
 - Switches: **nMOS** transistors

- **PUN:** *Load of the former nMOS gates with no active control is replaced by a full network*: the dual circuit of the nMOS pull down network:
 - Dual topology (series → parallel, parallel → series, aka the loop-cut duality)
 - Dual component: **pMOS** transistors

- The transistor gates in the PDN and PUN receive the **same control signals** (inputs)



CMOS gates

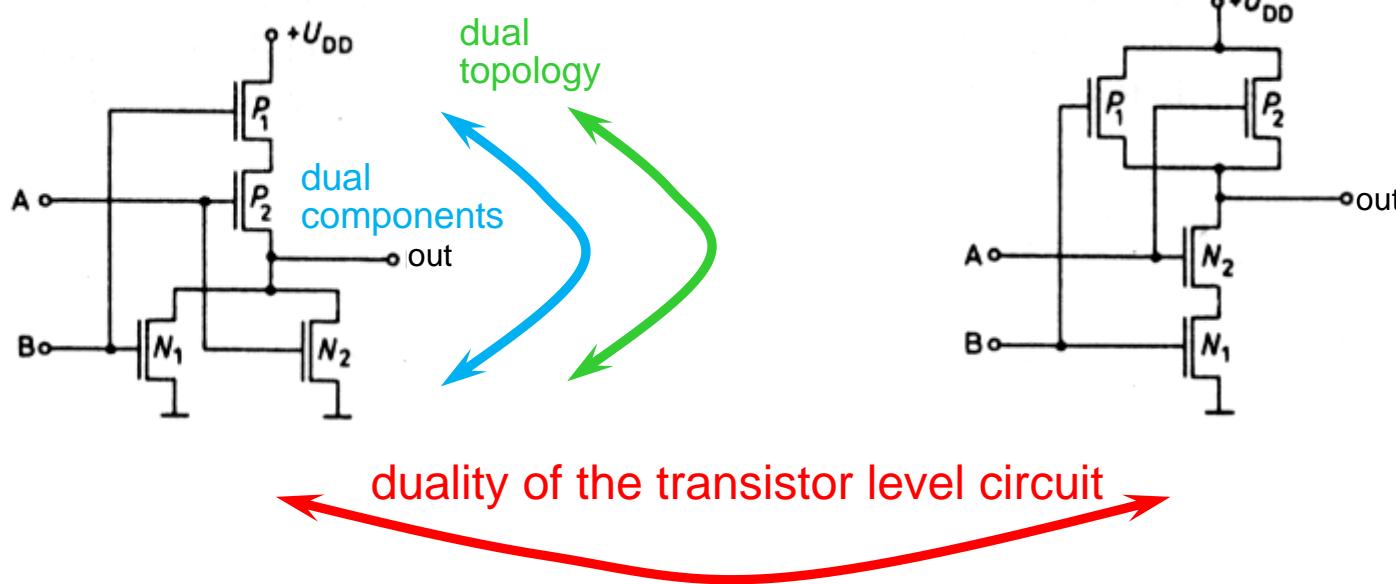
- In a CMOS inverter both transistors are actively controlled
- In case of gates there will be a PUN (pull up network: pMOS circuit) and a PDN (pull down network: nMOS circuit). The number of transistors both in PUN and PDN is equal to the number of inputs of the gate
 - For input combinations where the output is 0, the PDN realizes a short towards GND and the PUN is an open circuit;
 - if the output function is equal to 1, the PDN will be an open circuit and the PUN realizes a short towards VDD.

Circuits with dual topology should be realized from n and p channel transistors

- Gates of transistors receiving the same signal are connected

CMOS gates

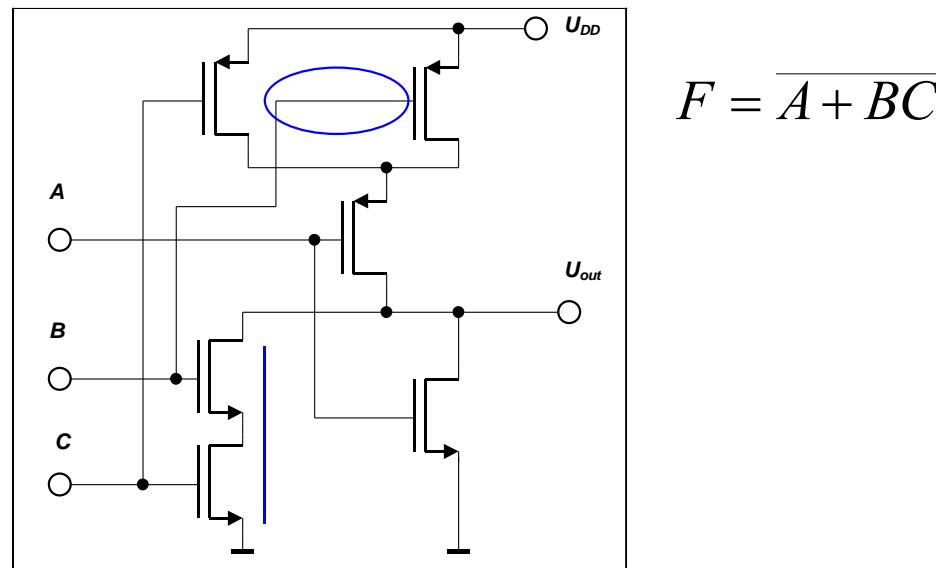
- NOR gate
- NAND gate



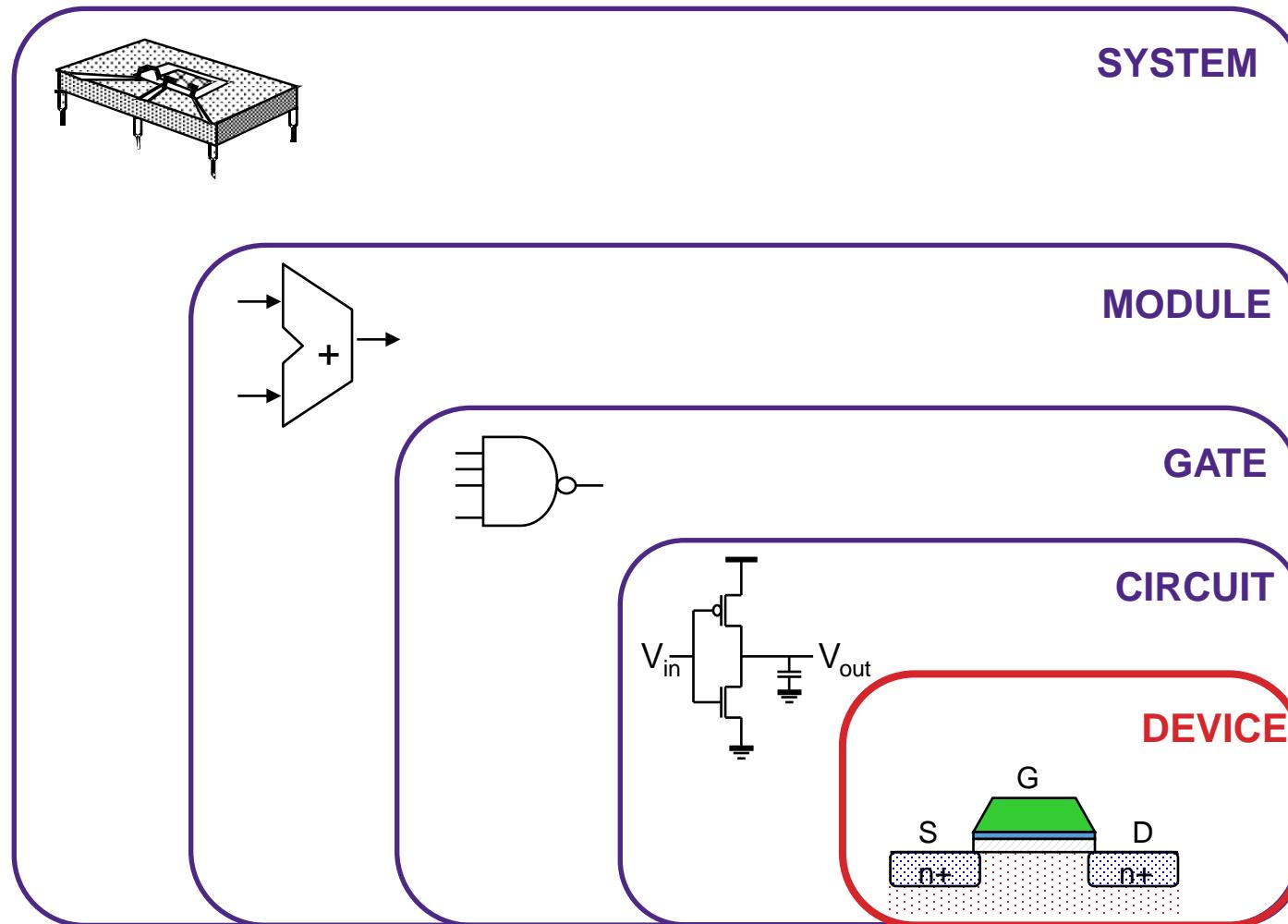
For an n input CMOS gate $2n$ transistors are needed
(passive load gates need only $n+1$ transistors)

Construction complex CMOS gates

- dual topology (loop \Rightarrow cut, cut \Rightarrow loop)
- dual components: nMOS replaced by pMOS
- transistor gates corresponding to the same signal must be connected
- proper sizing of the W/L ratios (electron/hole mobility mismatch)



The abstraction level of our study:

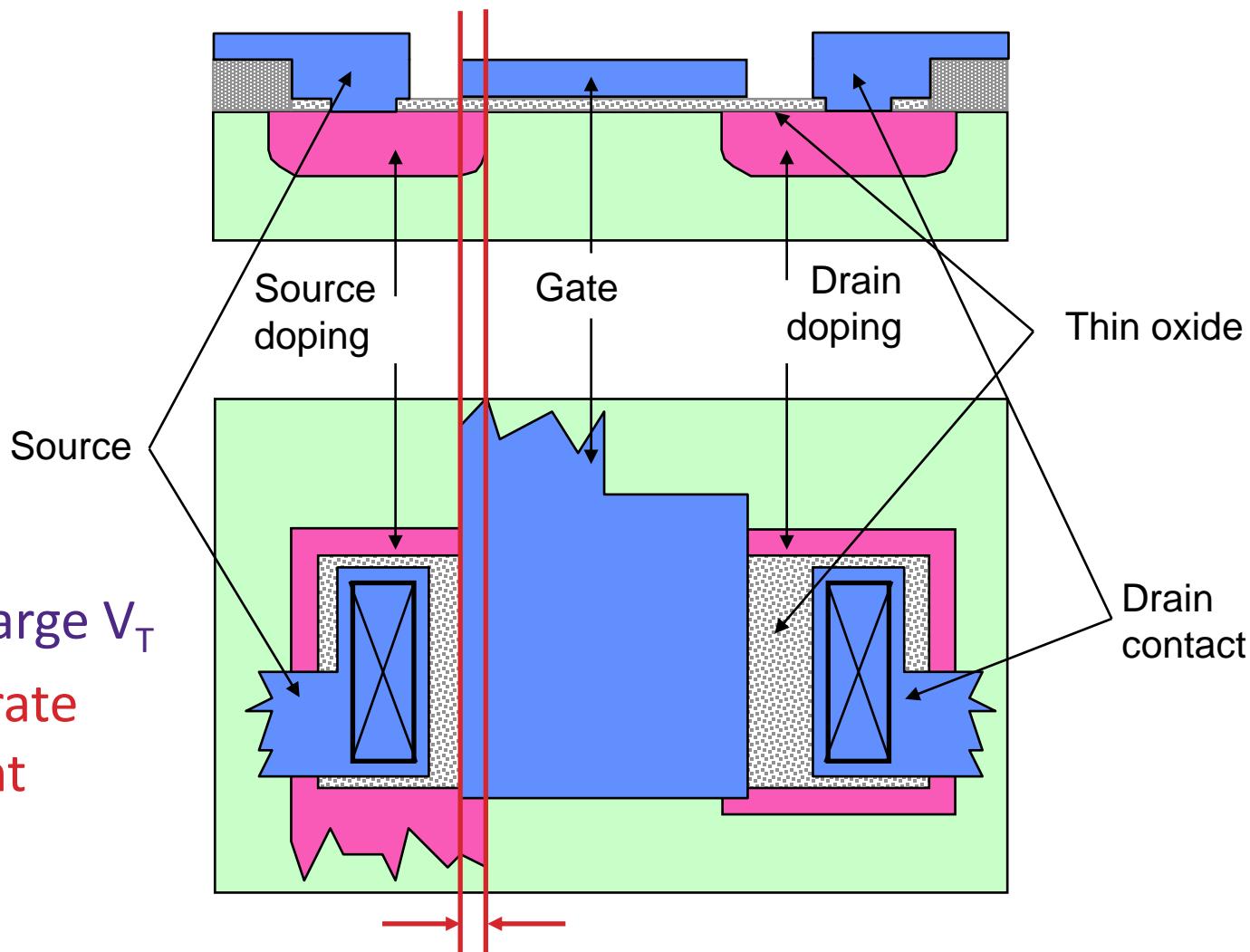


Metal gate MOS transistor

Layout view:

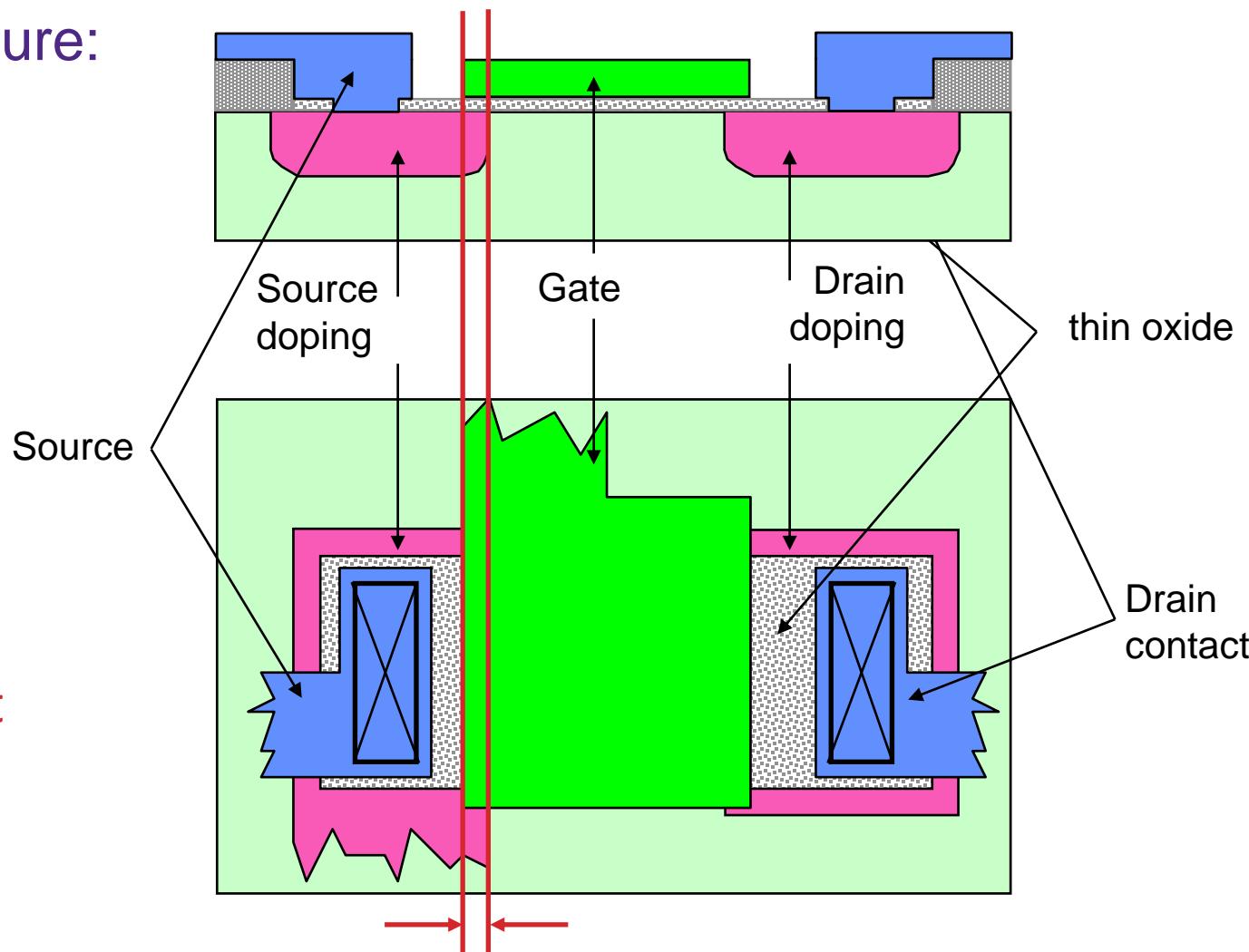
Problems:

- metal gate – large V_T
- requires accurate mask alignment



Poly-Si gate MOS transistor

In-depth structure:



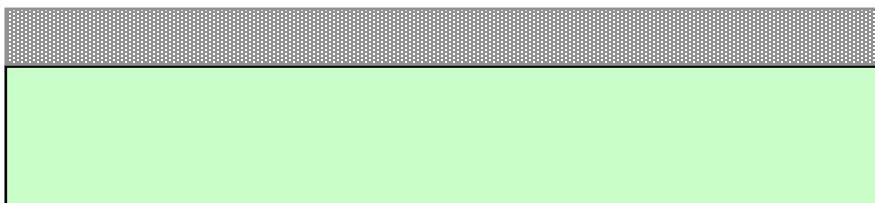
Layout view:

Advantages

- smaller V_T
- self alignment

A poli-Si gate nMOS process

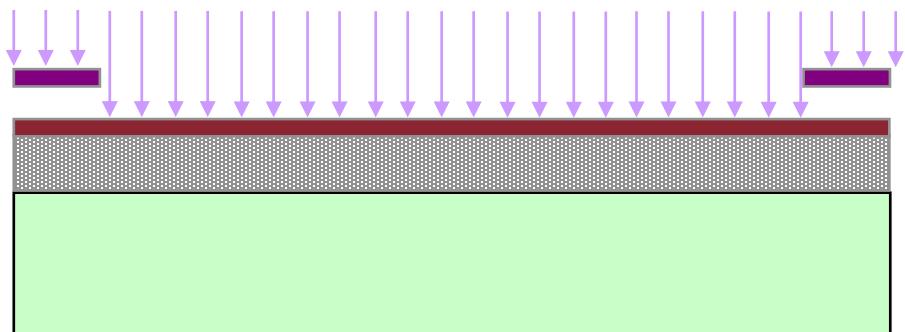
- ▶ Start with: p type substrate (Si wafer)
 - cleaing,
 - grow thick SiO_2 – this is called *field oxide*



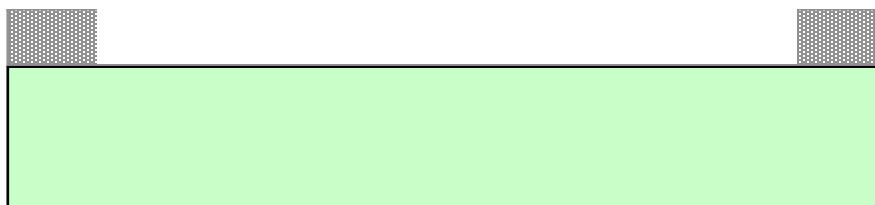
A poli-Si gate nMOS process

► Create the active zone with photolithography

- coat with resist,
- expose to UV light through a mask,
- development, removal of exposed resists
- etching of SiO_2 removal of the resist



M1: active zone

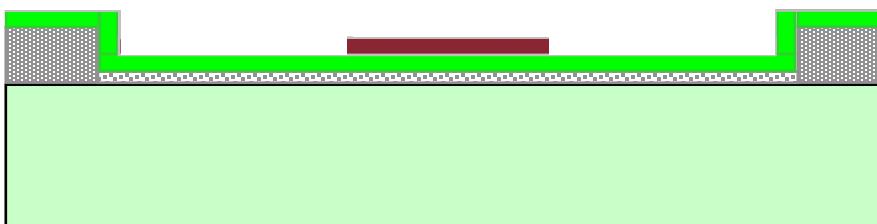


A poli-Si gate nMOS process

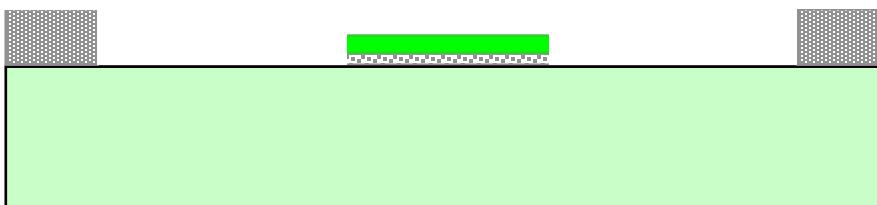
► Create the gate structure:

- growth of thin oxide
- deposit poly-Si
- pattern poly-Si with photolithography
- etch poly-Si, etch thin oxide

(resist, exposure, develop)

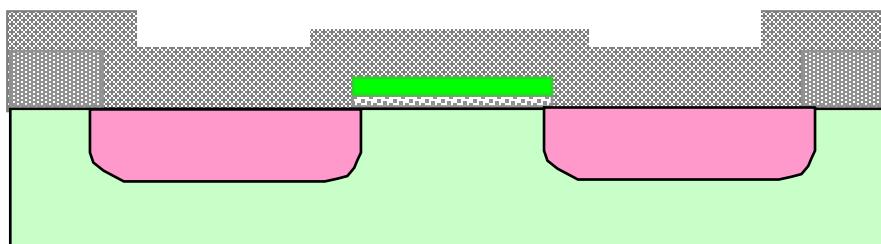
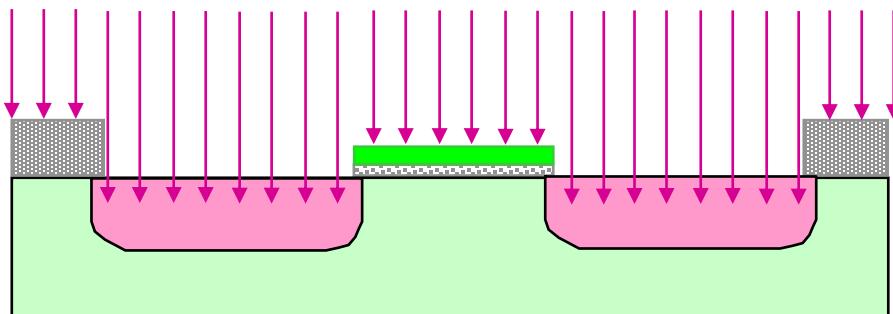


M2: poly-Si pattern



A poli-Si gate nMOS process

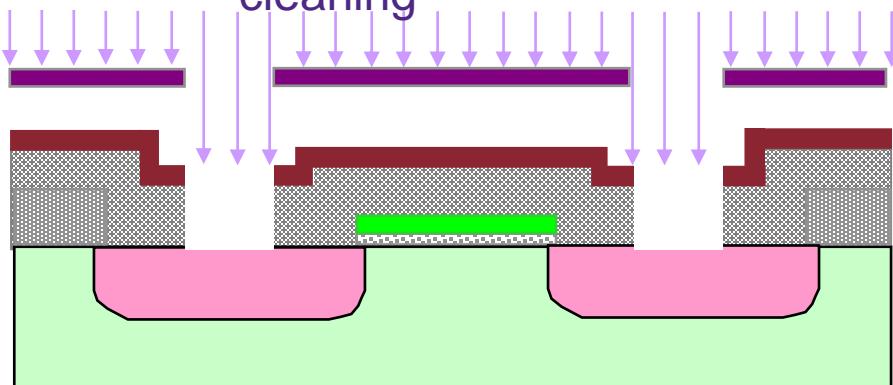
- ▶ S/D doping (implantation)
 - the exide (thin, thick) masks the dopants
 - this way the self-alignment of the gate is assured
- ▶ Passivation: deposit PSG



A poli-Si gate nMOS process

► Open contact windows through PSG

- photolithography (resist, expose pattern, develop)
- etching (copy the pattern)
- cleaning

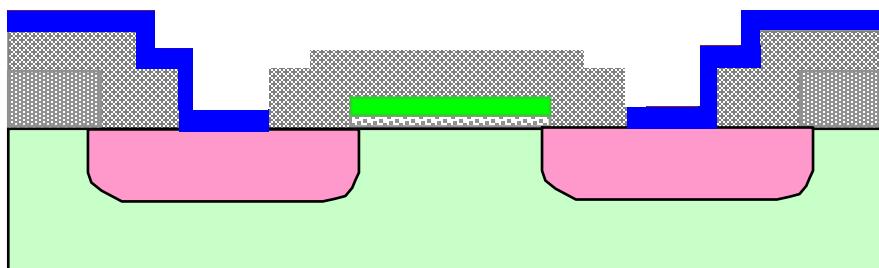


M3: contact window pattern

A poli-Si gate nMOS process

► Metallization

- Deposit Al
- photolithography, etching, cleaning

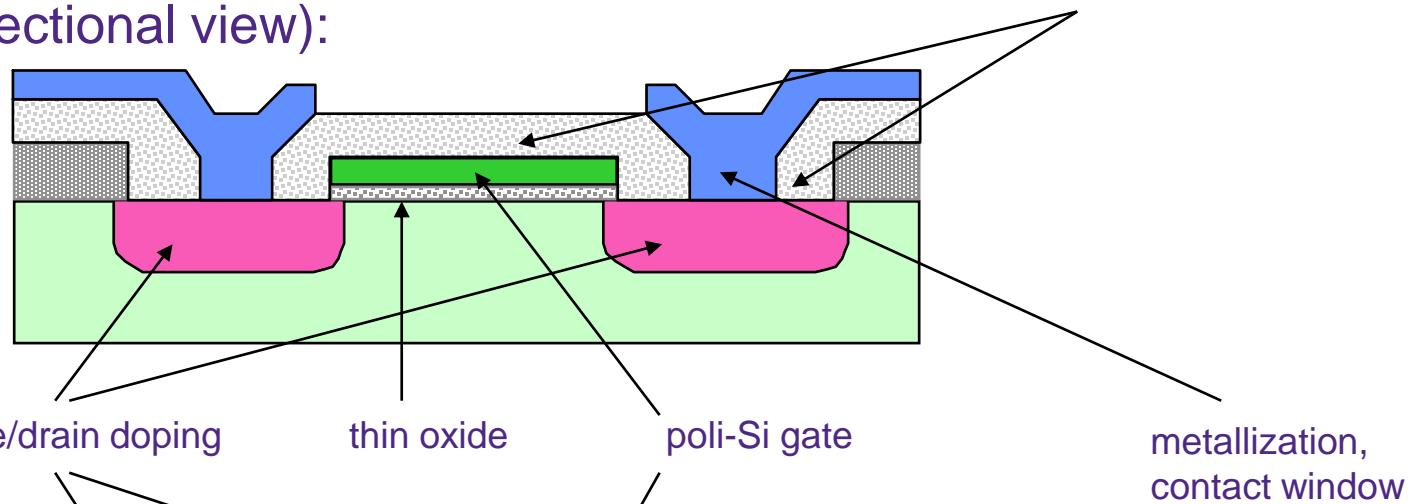


M4: metallization pattern

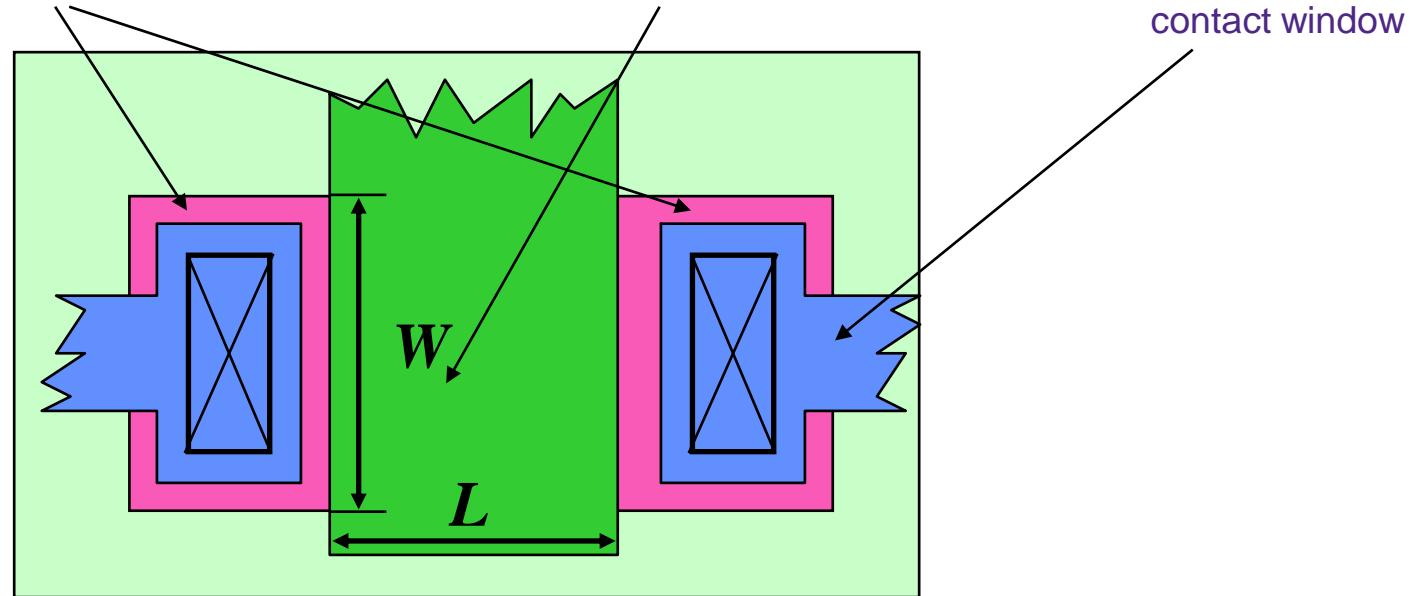
- The recipe of the process is given, the in-depth structure is determined by the sequence of the masks
- One needs to specify the shapes on the masks
 - ***The set of shapes on subsequent masks is called layout***

A poli-Si gate nMOS process

Structure (x-sectional view):



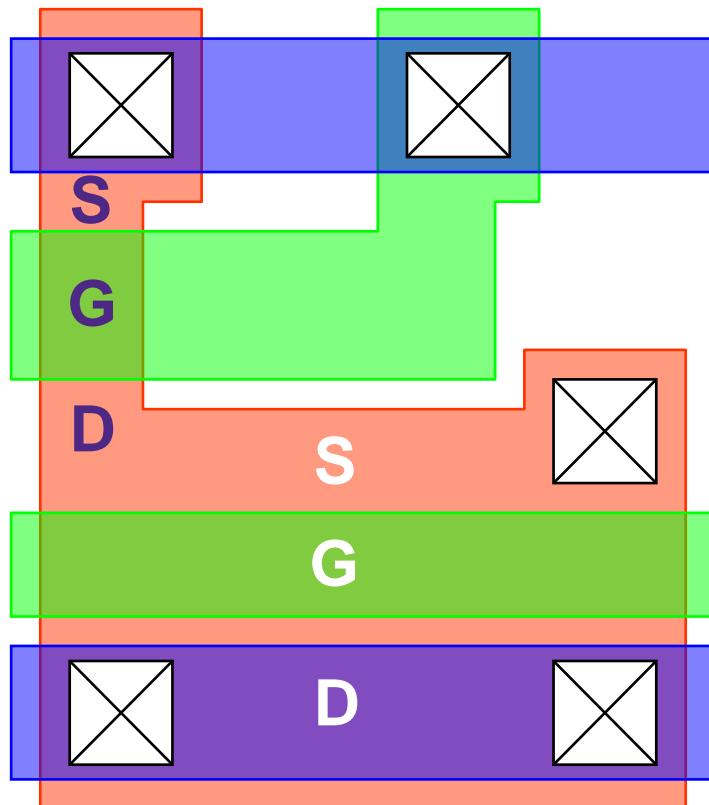
Layout (top view):



Steps of the self-aligned poli-Si gate process

- 1) Open window for the active region M
 - photolithography, field oxide etching
- 2) Growth of thin oxide
- 3) Window for hidden contacts M
 - Contacts the poli-Si gate (yet to be deposited) with the active region (after doping).
- 3) Deposit poli-Si
- 4) Patterning of poli-Si M
- 5) Open window through the thin oxide (etching only)

Layout of a depletion mode inverter

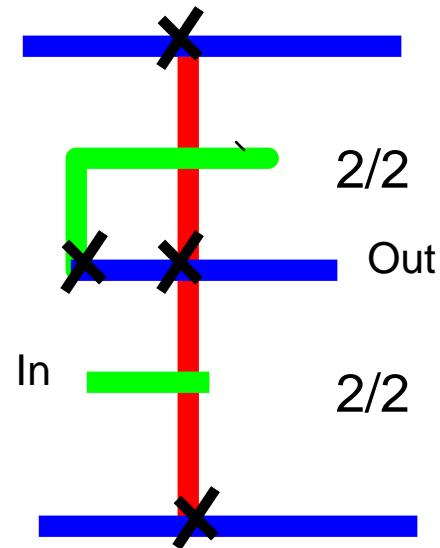
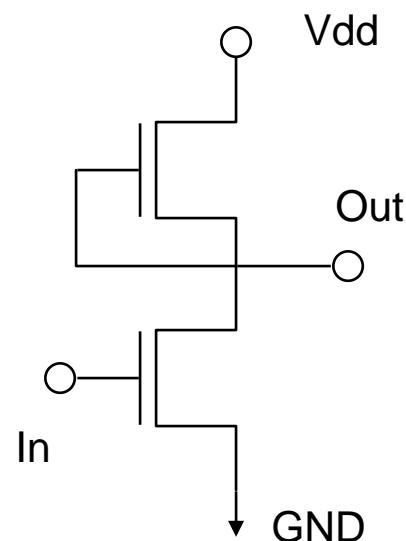


- Layout == set of 2D shapes on subsequent masks
- Masks are color coded:
 - active zone: red
 - poly-Si: green
 - contact windows: black
 - metal: blue
- Mask == layout layer

Where is a transistor? Channel between two doped regions:
CHANNEL = ACTIVE AND POLY

Simplified layout: *stick diagram*

- active
- poly
- metal
- ✗ contact

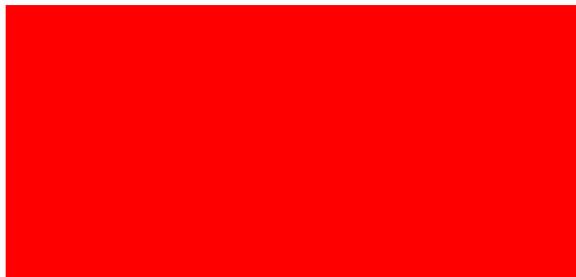


W/L ratios are given

Si-compilers

- Logic schematic / netlist / high level description
- Transistor level schematic with W/L information
- Stick diagram layout
- Actual layout
 - Automatic conversion between these representations
 - HARDWARE SYNTHESIS
 - 1. From behavioural description structural description
 - 2. Implementation of the structural description with a given realization mode / manufacturing process: *technology mapping*
 - *We have seen basics of the realization of an application specific integrated circuit (ASIC)*
 - *Designs can also be mapped to an FPGA*

Layout primitives: simple shapes



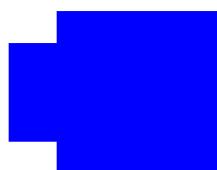
Active zone (window opening through the oxide)



Gate (mask of poly-Si pattern)

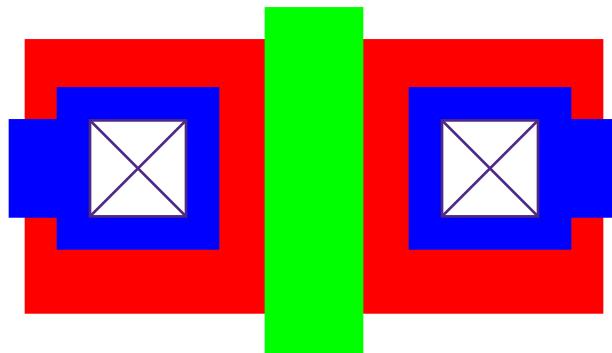


Contacts (window opening mask through oxide/PSG)

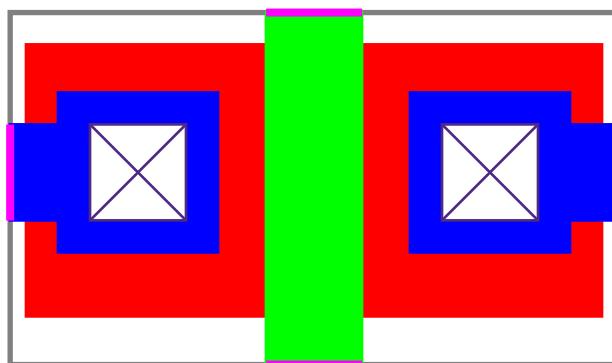


S/D lines (mask of metallization pattern)

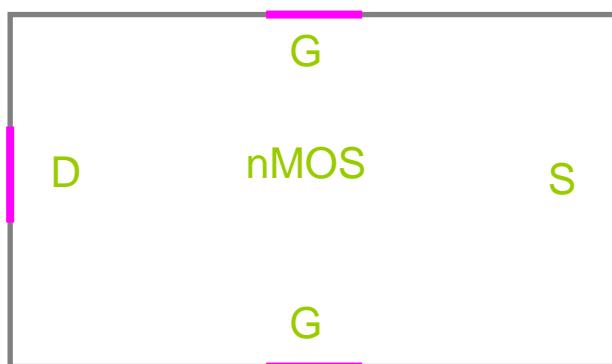
Layout macros – from primitives



layout of an nMOS transistor: *layout primitives on actual layers corresponding to real masks*

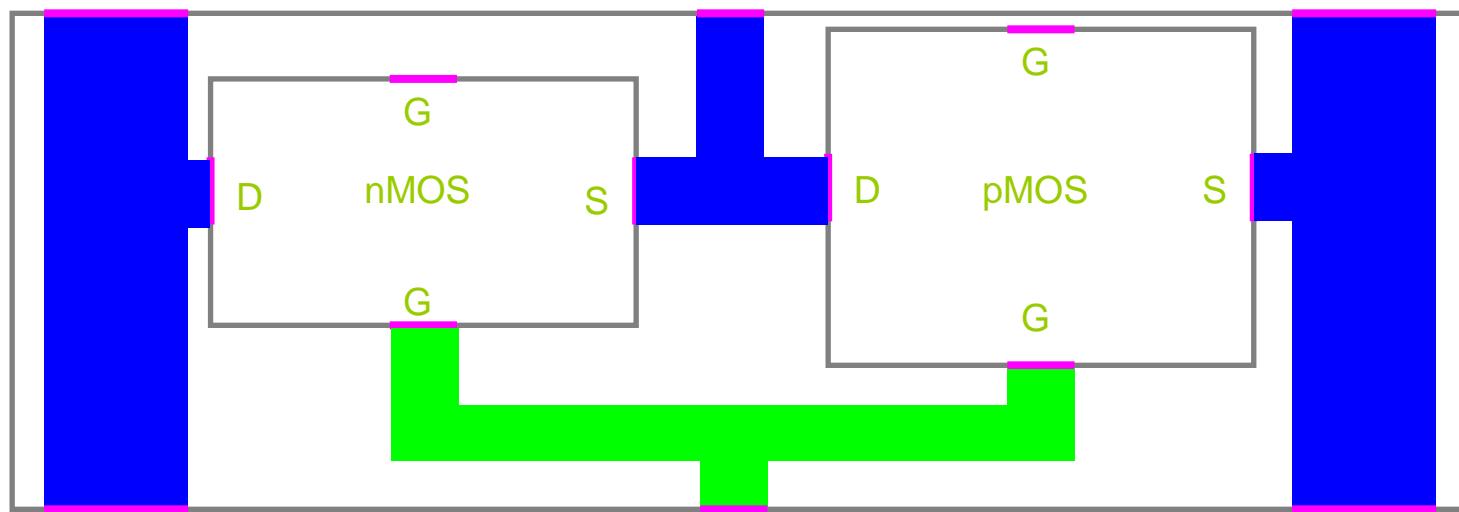


nMOS transistor layout + outline + pins



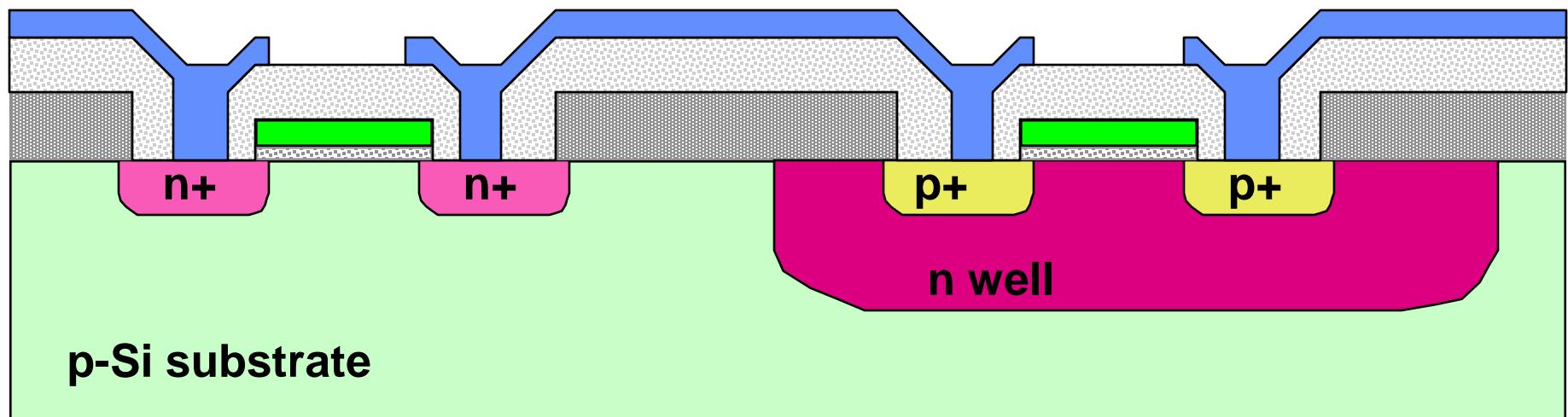
nMOS transistor macro:
outline, pins, scripts: *pseudo layers*

Layout macros – from macros and primitives



Gate level layout

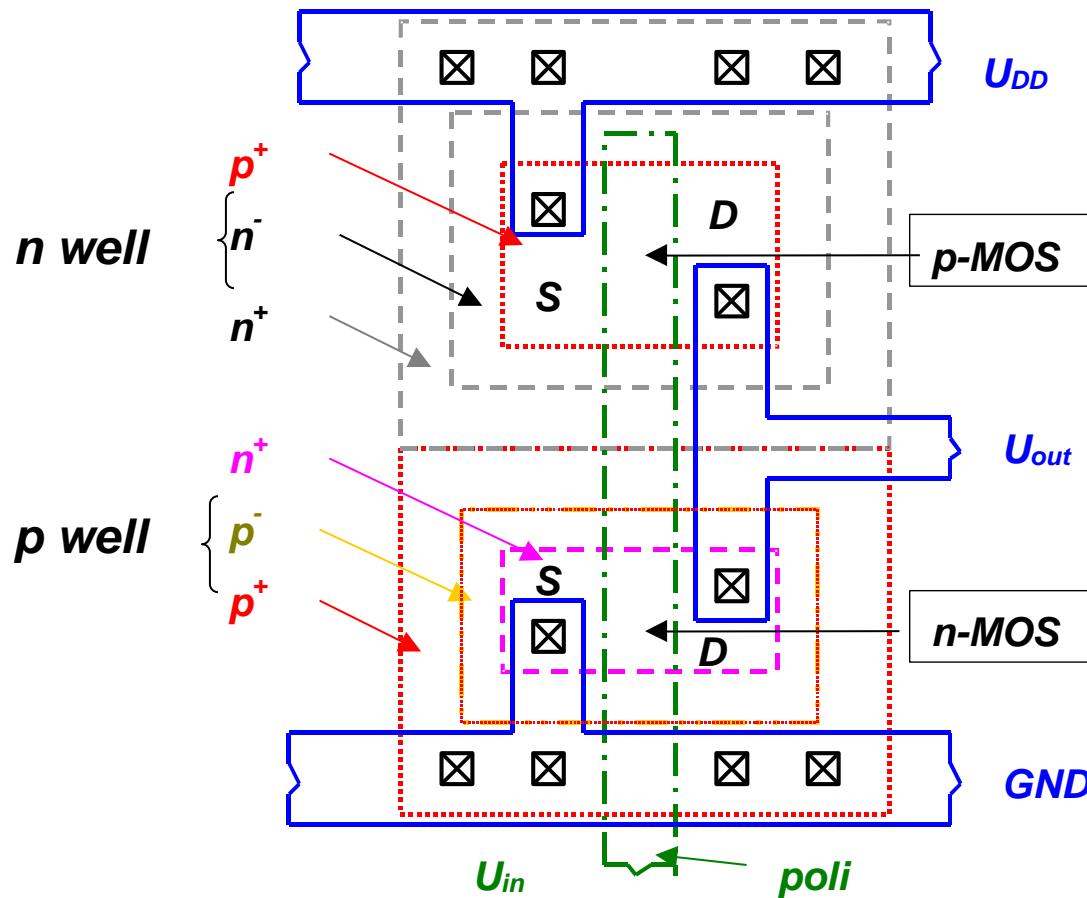
CMOS structure (inverter)



CMOS structures

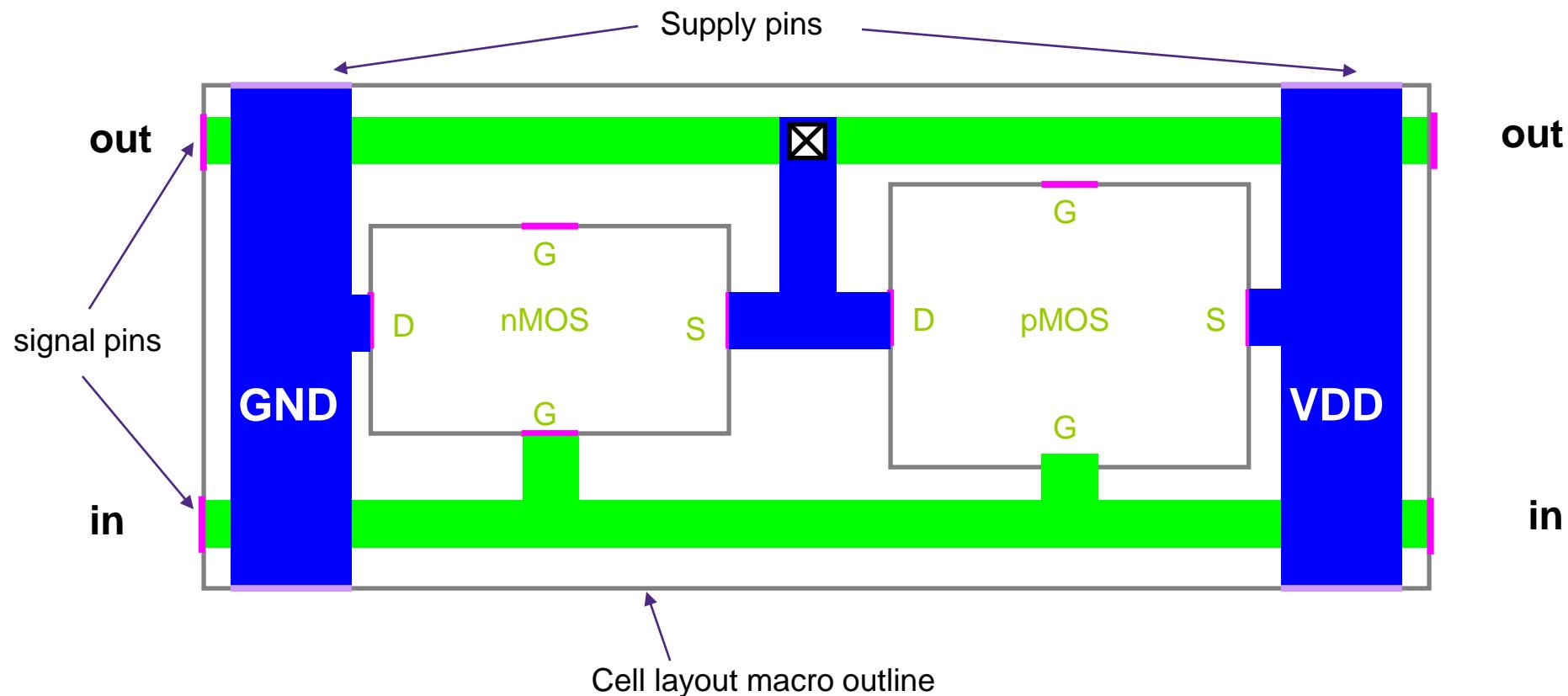
- Further masks:
 - n-well (or p-well, depending on the substrate)
 - p doping (or n doping, depending on the substrate)
- Multiple metal layer CMOS:
 - each metallization needs own mask,
 - contact windows, vias
- There could be multiple poly-Si layers (analog CMOS)
- Typically: 15..20 masks
- Certain rules need to be kept for manufacturability: ***design rules***
 - ***come from the process, given by Si-foundry***

Layout of a CMOS inverter



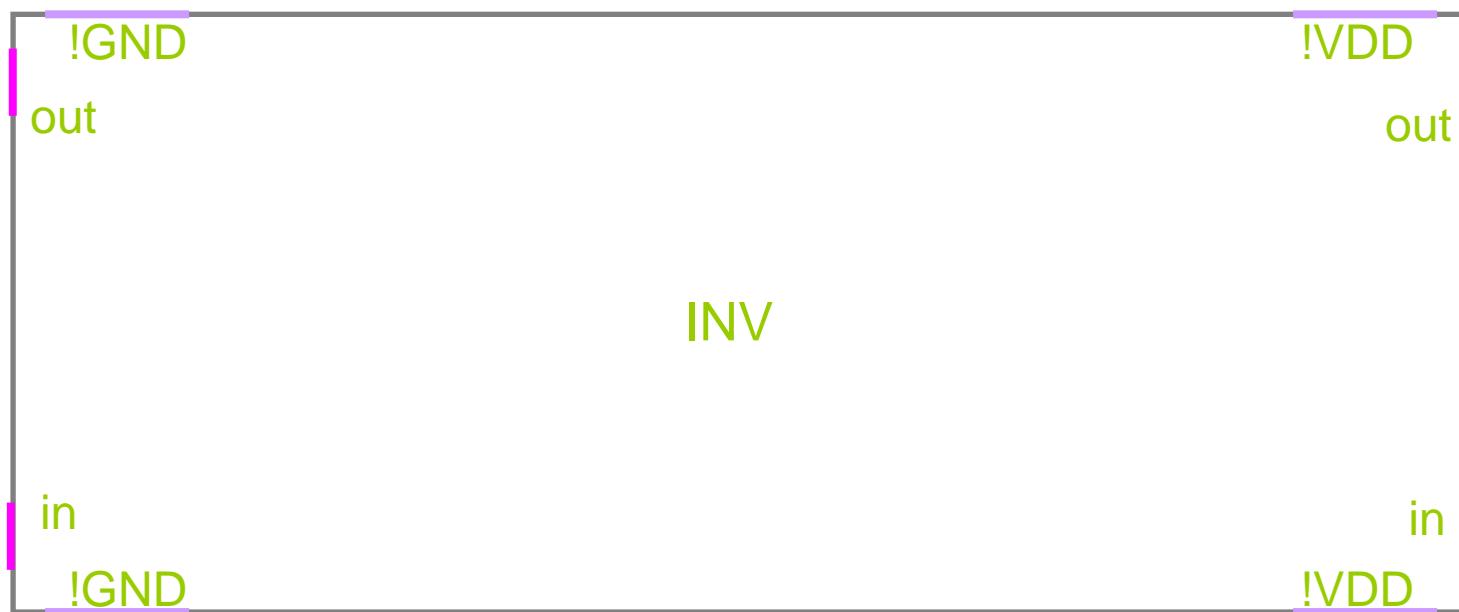
Standard cells of gates

- The CMOS inverter layout shown before has also been created according to conventions of standard cell design

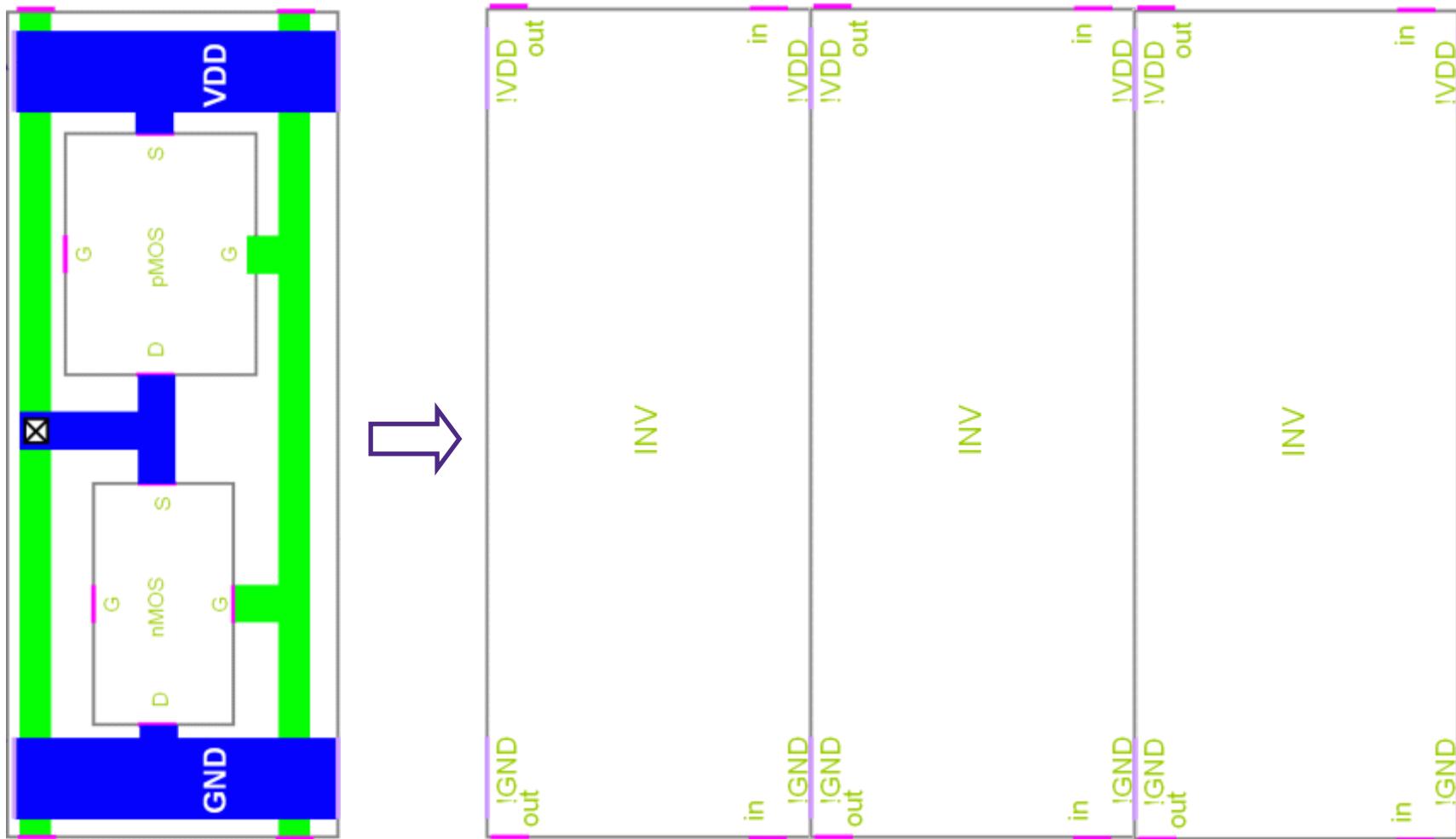


Standard cells

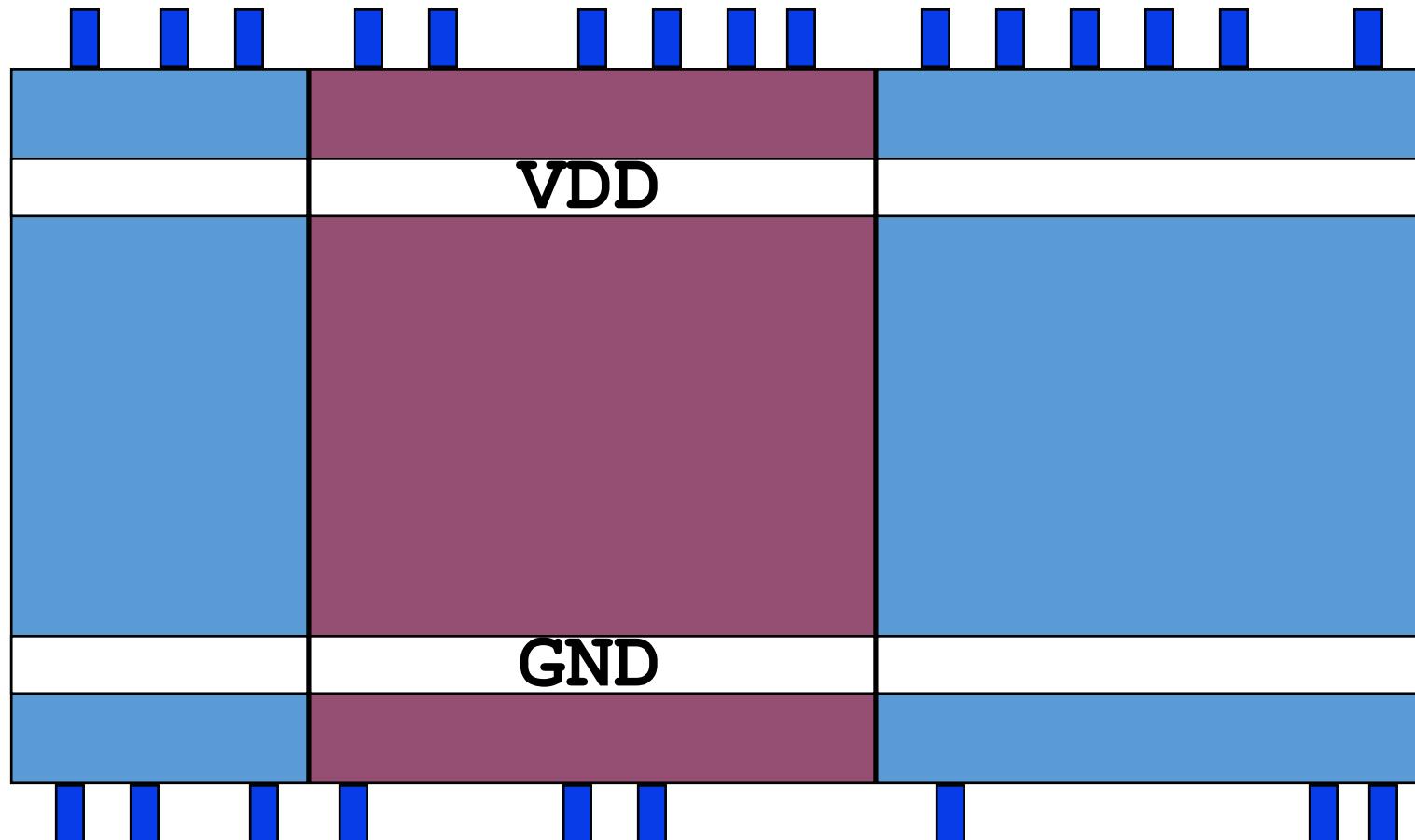
- In layout view one can refer to the inverter through its layout macro (cell outline and pins)



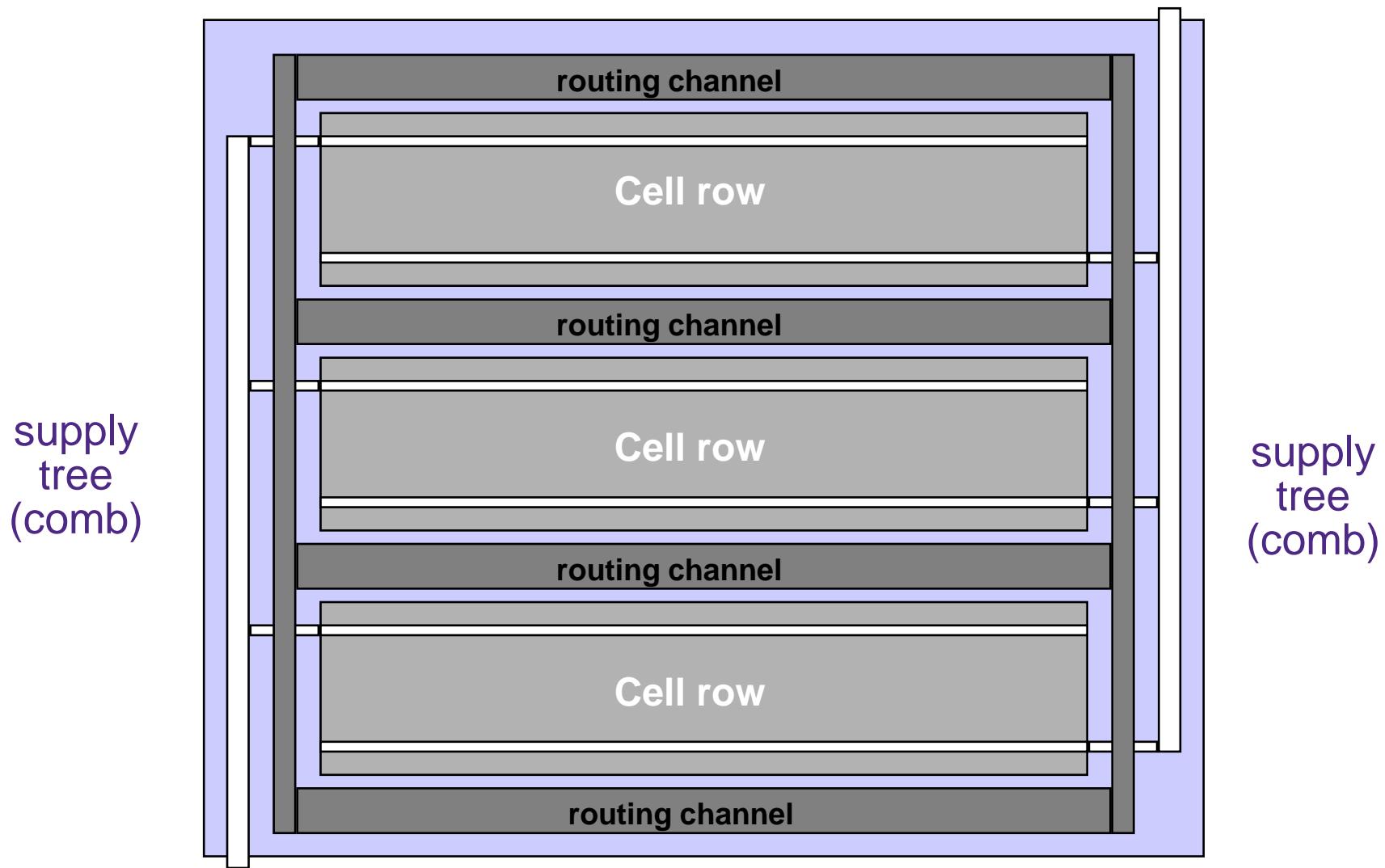
Standard cells



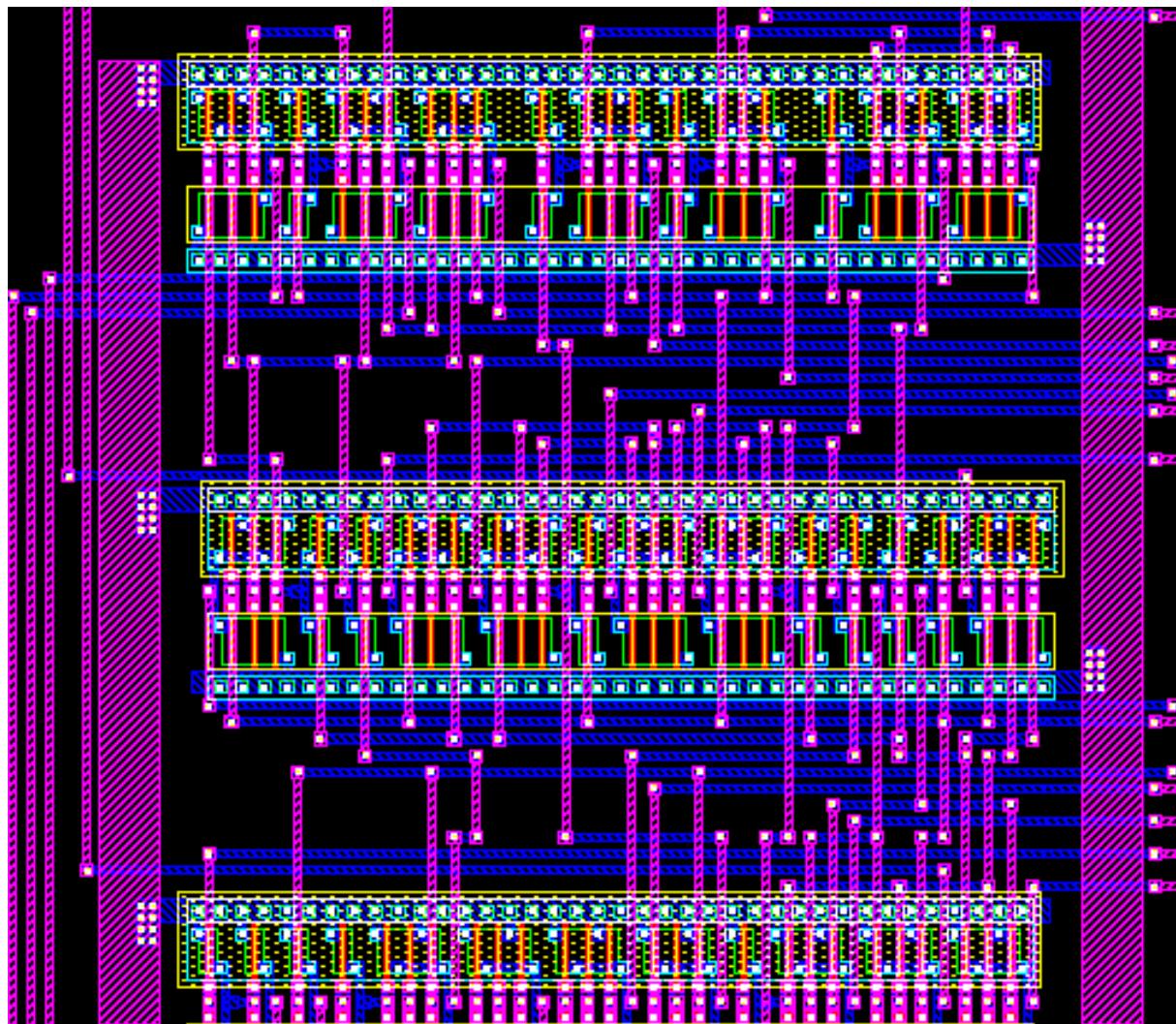
Standard cells in a row:



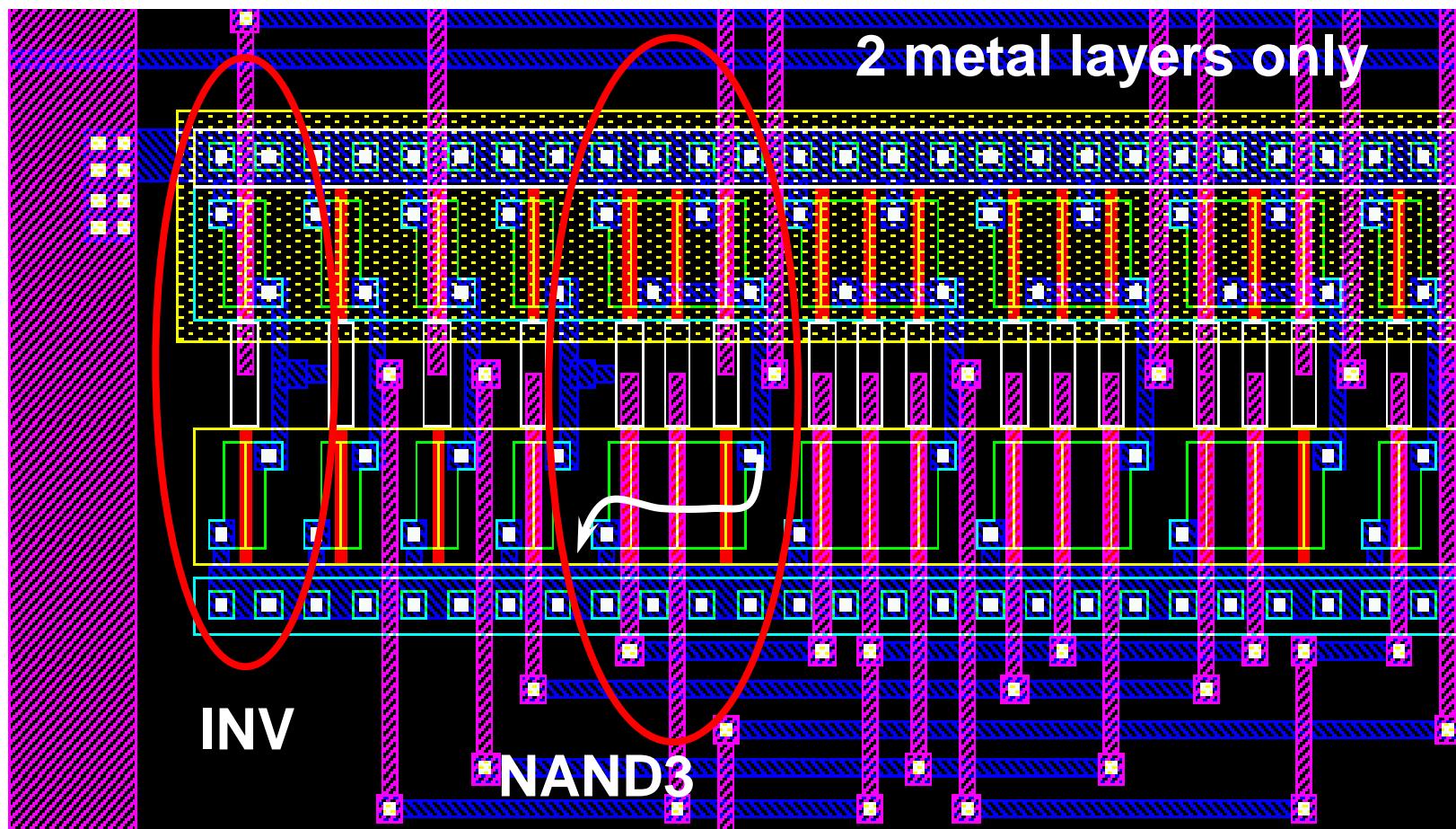
Standard cell IC:



Detail of a standard cell IC:



Details of a CMOS circuit



Layout extraction: checking, real delays

The same in CAD tools of VLSI design

