# DAILY ASSESSMENT

|  |  |  |  |
| --- | --- | --- | --- |
| Date: | 11/06/2020 | Name: | GAURAV NR |
| Course: | VLSI | USN: | 4AL15EC025 |
| Topic: | |  | | --- | | **MOS transistor basics-2 and 3** | | Semester & Section: | 8TH SEM & A Section |
| Github Repository: | gaurav |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS**        Complementary MOSFET (CMOS) technology is widely used today to form circuits in numerous and varied applications. Today’s computers, CPUs and cell phones make use of CMOS due to several key advantages. CMOS offers low power dissipation, relatively high speed, high noise margins in both states, and will operate over a wide range of source and input voltages (provided the source voltage is fixed)  For the processes we will discuss, the type of transistor available is the Metal-Oxide-Semiconductor Field Effect Transistor (MOSFET). These transistors are formed **as a ‘sandwich’** consisting of a semiconductor layer, usually a slice, or wafer, from a single crystal of silicon; a layer of silicon dioxide (the oxide) and a layer of metal. Structure of a MOSFET Structure of misfet  As shown in the figure, MOS structure contains three layers −   * **The Metal Gate Electrode** * **The Insulating Oxide Layer (SiO2)** * **P – type Semiconductor (Substrate)**   MOS structure forms a capacitor, with gate and substrate are as two plates and oxide layer as the dielectric material. The thickness of dielectric material (SiO2) is usually between 10 nm and 50 nm. Carrier concentration and distribution within the substrate can be manipulated by external voltage applied to gate and substrate terminal. Now, to understand the structure of MOS, first consider the basic electric properties of P – Type semiconductor substrate.  Concentration of carrier in semiconductor material is always following the **Mass Action Law**. Mass Action Law is given by −  $$n.p=n\_{i}^{2}$$  Where,   * **n** is carrier concentration of electrons * **p** is carrier concentration of holes * **ni** is intrinsic carrier concentration of Silicon   Now assume that substrate is equally doped with acceptor (Boron) concentration NA. So, electron and hole concentration in p–type substrate is  $$n\_{po}=\frac{n\_{i}^{2}}{N\_{A}}$$  $$p\_{po}=N\_{A}$$  Here, doping concentration **NA** is (1015 to 1016 cm−3) greater than intrinsic concentration ni. Now, to understand the MOS structure, consider the energy level diagram of p–type silicon substrate.  P-type Silicon Substrate  As shown in the figure, the band gap between conduction band and valance band is 1.1eV. Here, Fermi potential ΦF is the difference between intrinsic Fermi level (Ei) and Fermi level (EFP).  Where Fermi level EF depends on the doping concentration. Fermi potential ΦF is the difference between intrinsic Fermi level (Ei) and Fermi level (EFP).  Mathematically,  $$\Phi\_{Fp}=\frac{E\_{F}-E\_{i}}{q}$$  The potential difference between conduction band and free space is called electron affinity and is denoted by qx.  So, energy required for an electron to move from Fermi level to free space is called work function (qΦS) and it is given by  $$q\Phi \_{s}=(E\_{c}-E\_{F})+qx$$  The following figure shows the energy band diagram of components that make up the MOS.  Energy Level Diagram of Components  As shown in the above figure, insulating SiO2 layer has large energy band gap of 8eV and work function is 0.95 eV. Metal gate has work function of 4.1eV. Here, the work functions are different so it will create voltage drop across the MOS system. The figure given below shows the combined energy band diagram of MOS system.  Combined Energy Band Diagram  As shown in this figure, the fermi potential level of metal gate and semiconductor (Si) are at same potential. Fermi potential at surface is called surface potential ΦS and it is smaller than Fermi potential ΦF in magnitude. Working of a MOSFET MOSFET consists of a MOS capacitor with two p-n junctions placed closed to the channel region and this region is controlled by gate voltage. To make both the p-n junction reverse biased, substrate potential is kept lower than the other three terminals potential.  If the gate voltage will be increased beyond the threshold voltage (VGS>VTO), inversion layer will be established on the surface and n – type channel will be formed between the source and drain. This n – type channel will carry the drain current according to the VDS value.  For different value of VDS, MOSFET can be operated in different regions as explained below. Linear Region At VDS = 0, thermal equilibrium exists in the inverted channel region and drain current ID = 0. Now if small drain voltage, VDS > 0 is applied, a drain current proportional to the VDS will start to flow from source to drain through the channel.  The channel gives a continuous path for the flow of current from source to drain. This mode of operation is called **linear region**. The cross sectional view of an n-channel MOSFET, operating in linear region, is shown in the figure given below.  Linear Region At the Edge of Saturation Region Now if the VDS is increased, charges in the channel and channel depth decrease at the end of drain. For VDS = VDSAT, the charges in the channel is reduces to zero, which is called **pinch – off point**. The cross sectional view of n-channel MOSFET operating at the edge of saturation region is shown in the figure given below.  Edge of Saturation Region Saturation Region For VDS>VDSAT, a depleted surface forms near to drain, and by increasing the drain voltage this depleted region extends to source.  This mode of operation is called **Saturation region**. The electrons coming from the source to the channel end, enter in the drain – depletion region and are accelerated towards the drain in high electric field.  Saturation Region MOSFET Current – Voltage Characteristics To understand the current – voltage characteristic of MOSFET, approximation for the channel is done. Without this approximation, the three dimension analysis of MOS system becomes complex. The **Gradual Channel Approximation (GCA)** for current – voltage characteristic will reduce the analysis problem. Gradual Channel Approximation (GCA) Consider the cross sectional view of n channel MOSFET operating in the linear mode. Here, source and substrate are connected to the ground. VS = VB = 0. The gate – to – source (VGS) and drain – to – source voltage (VDS) voltage are the external parameters that control the drain current ID.  The voltage, VGS is set to a voltage greater than the threshold voltage VTO, to create a channel between the source and drain. As shown in the figure, x – direction is perpendicular to the surface and y – direction is parallel to the surface.  Here, y = 0 at the source end as shown in the figure. The channel voltage, with respect to the source, is represented by **VC(Y)**. Assume that the threshold voltage VTO is constant along the channel region, between y = 0 to y = L.  Gradual Channel Approximation |

**DAILY ASSESSMENT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **11/06/2020** | **Name:** | **GAURAV N R** |
| **Course:** | **Begineer PHP and SQL** | **USN:** | **4AL15EC025** |
| **Topic:** | **PHP functions**  **Using external files and images** | **Semester & Section:** | **8TH SEM & A Section** |
| **Github Repository:** | **gaurav** |  |  |

|  |
| --- |
| **AFTERNOON SESSION DETAILS** |
|  |

|  |
| --- |
| **Report**-  PHP functions are similar to other programming languages. A function is a piece of code which takes one more input in the form of parameter and does some processing and returns a value.  You already have seen many functions like fopen() and fread() etc. They are built-in functions but PHP gives you option to create your own functions as well.  There are two parts which should be clear to you −   * Creating a PHP Function * Calling a PHP Function   In fact you hardly need to create your own PHP function because there are already more than 1000 of built-in library functions created for different area and you just need to call them according to your requirement.  Please refer to [PHP Function Reference](https://www.tutorialspoint.com/php/php_function_reference.htm) for a complete set of useful functions. Creating PHP Function Its very easy to create your own PHP function. Suppose you want to create a PHP function which will simply write a simple message on your browser when you will call it. Following example creates a function called writeMessage() and then calls it just after creating it.  Note that while creating a function its name should start with keyword function and all the PHP code should be put inside { and } braces as shown in the following example below −  <html>    <head>  <title>Writing PHP Function</title>  </head>    <body>    <?php  /\* Defining a PHP Function \*/  function writeMessage() {  echo "You are really a nice person, Have a nice time!";  }    /\* Calling a PHP Function \*/  writeMessage();  ?>    </body>  </html>  This will display following result −  You are really a nice person, Have a nice time! PHP Functions with Parameters PHP gives you option to pass your parameters inside a function. You can pass as many as parameters your like. These parameters work like variables inside your function. Following example takes two integer parameters and add them together and then print them.  <html>    <head>  <title>Writing PHP Function with Parameters</title>  </head>    <body>    <?php  function addFunction($num1, $num2) {  $sum = $num1 + $num2;  echo "Sum of the two numbers is : $sum";  }    addFunction(10, 20);  ?>    </body>  </html>  This will display following result −  Sum of the two numbers is : 30 Passing Arguments by Reference It is possible to pass arguments to functions by reference. This means that a reference to the variable is manipulated by the function rather than a copy of the variable's value.  Any changes made to an argument in these cases will change the value of the original variable. You can pass an argument by reference by adding an ampersand to the variable name in either the function call or the function definition.  Following example depicts both the cases.  <html>    <head>  <title>Passing Argument by Reference</title>  </head>    <body>    <?php  function addFive($num) {  $num += 5;  }    function addSix(&$num) {  $num += 6;  }    $orignum = 10;  addFive( $orignum );    echo "Original Value is $orignum<br />";    addSix( $orignum );  echo "Original Value is $orignum<br />";  ?>    </body>  </html>  This will display following result −  Original Value is 10  Original Value is 16 PHP Functions returning value A function can return a value using the return statement in conjunction with a value or object. return stops the execution of the function and sends the value back to the calling code.  You can return more than one value from a function using return array(1,2,3,4).  Following example takes two integer parameters and add them together and then returns their sum to the calling program. Note that return keyword is used to return a value from a function.  <html>    <head>  <title>Writing PHP Function which returns value</title>  </head>    <body>    <?php  function addFunction($num1, $num2) {  $sum = $num1 + $num2;  return $sum;  }  $return\_value = addFunction(10, 20);    echo "Returned value from the function : $return\_value";  ?>    </body>  </html>  This will display following result −  Returned value from the function : 30 Setting Default Values for Function Parameters You can set a parameter to have a default value if the function's caller doesn't pass it.  Following function prints NULL in case use does not pass any value to this function.  <html>    <head>  <title>Writing PHP Function which returns value</title>  </head>    <body>    <?php  function printMe($param = NULL) {  print $param;  }    printMe("This is test");  printMe();  ?>    </body>  </html>  This will produce following result −  This is test Dynamic Function Calls It is possible to assign function names as strings to variables and then treat these variables exactly as you would the function name itself. Following example depicts this behaviour.  <html>    <head>  <title>Dynamic Function Calls</title>  </head>    <body>    <?php  function sayHello() {  echo "Hello<br />";  }    $function\_holder = "sayHello";  $function\_holder();  ?>    </body>  </html>  This will display following result −  Hello |