**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **10/06/2020** | **Name:** | **Navya R** |
| **Course:** | **VLSI design** | **USN:** | **4AL16EC041** |
| **Topic:** | **MOS transistor** | **Semester & Section:** | **8 A** |
| **Github Repository:** | **Navya-R** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **Image of session** |
| |  | | --- | | **REPORT**  Basic MOS transistor |  The MOS transistor The most basic element in the design of a large scale integrated circuit is the transistor. 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.  These layers are patterned in a manner which permits transistors to be formed in the semiconductor material (the ``substrate''); a diagram showing a typical (idealized) MOSFET is shown in Figure [gif](http://www.cs.mun.ca/~paul/transistors/node1.html#figmosfet). Silicon dioxide is a very good insulator, so a very thin layer, typically only a few hundred molecules thick, is required. Actually, the transistors which we will use do not use metal for their gate regions, but instead use polycrystalline silicon (poly).  Polysilicon gate FET's have replaced virtually all of the older devices using metal gates in large scale integrated circuits. (Both metal and polysilicon FET's are sometimes referred to as IGFET's --- insulated gate field effect transistors, since the silicon dioxide under the gate is an insulator. We will still continue to use the term MOSFET to refer to polysilicon gate FET's.)    The source and drain regions are quite similar, and are labeled depending on to what they are connected. The source is the terminal, or node, which acts as the source of charge carriers; charge carriers leave the source and travel to the drain. In the case of an N channel MOSFET, the source is the more negative of the terminals; in the case of a P channel device, it is the more positive of the terminals. The area under the gate oxide is called the ``channel''.  The MOSFET can operate as a very efficient switch for current flowing between the source and drain region of the device. For the simplest type of MOSFET, the ``enhancement mode MOSFET'', which acts as a ``normally open'' switch, the operation of the device can be described qualitatively with reference    Figure [gif](http://www.cs.mun.ca/~paul/transistors/node1.html#figenhancement)(a) shows an N-channel MOSFET with the source and drain connected to power () and ground (); the substrate, or body of the device, is also connected to ground. In this case, there is a reverse biased PN junction between at least one of the N wells and the substrate, so no current can flow through the substrate. In particular, there will be no current flow in the channel region under the gate of the transistor, and therefore no current will flow between the source and drain of the device. Under these conditions, the MOSFET is turned *off*.  Figure [gif](http://www.cs.mun.ca/~paul/transistors/node1.html#figenhancement)(b) shows the same N-channel MOSFET with a positive charge applied to the gate of the device. Under these circumstances, if the gate is given a sufficiently large charge, negative charge carriers (electrons) will be attracted from the bulk of the substrate material into the channel region immediately below the oxide under the gate.  When more electrons are attracted into this region than there are positive charge carriers (holes) in the channel, then the channel effectively behaves as an N type region, and current can flow between the source and the drain. When this happens, the MOSFET is turned *on*. Note that a certain minimum charge must be applied to the gate to overcome the excess of holes already in the channel region because of the P type doping in the substrate.  This means that the switch is not turned on immediately, rather there must be some minimum amount of charge applied to the gate before the transistor is switched on. The voltage which must be applied to the gate before the transistor allows current to flow between the source and drain is called the ``threshold voltage'', designated as .  This type of transistor is called an N channel enhancement mode MOSFET. (It is called N channel because the conduction in the channel is due to N type charge carriers; it is said to be an ``enhancement mode'' device because the channel conduction is enhanced by a charge applied  to the gate.) Figure [gif](http://www.cs.mun.ca/~paul/transistors/node1.html#figcharacteristics) shows a set of typical characteristic curves[gif](http://www.cs.mun.ca/~paul/transistors/footnode.html#262) for the current  between the drain and source of a MOSFET as a function of the voltage  for a range of gate voltages, . |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date:** | **10/6/2020** | **Name:** | **Navya R** | |
| **Course:** | **Java** | **USN:** | **4al16ec041** | |
| **Topic:** | **Arrays of Strings**  **Classes and Objects** | **Semester & Section:** | **8 A** | |
| **AFTERNOON SESSION DETAILS** | | | |
| **REPORT**  A Class is like an object constructor, or a "blueprint" for creating objects. Create a Class To create a class, use the keyword class: **MyClass.java** Create a class named "MyClass" with a variable x:  public class MyClass {  int x = 5;  } Create an Object In Java, an object is created from a class. We have already created the class named MyClass, so now we can use this to create objects.  To create an object of MyClass, specify the class name, followed by the object name, and use the keyword new: **Example** Create an object called "myObj" and print the value of x:  public class MyClass {  int x = 5;  public static void main(String[] args) {  MyClass **myObj** = new MyClass();  System.out.println(myObj.x);  }  } Multiple Objects You can create multiple objects of one class: **Example** Create two objects of MyClass:  public class MyClass {  int x = 5;  public static void main(String[] args) {  MyClass **myObj1** = new MyClass(); // Object 1  MyClass **myObj2** = new MyClass(); // Object 2  System.out.println(myObj1.x);  System.out.println(myObj2.x);  }  }  A Java constructor is special method that is called when an object is instantiated. In other words, when you use the new keyword. The purpose of a Java constructor is to initializes the newly created object before it is used. This Java constructors tutorial will explore Java constructors in more detail.  Here is a simple example that creates an object, which results in the class constructor being called:  MyClass myClassObj = new MyClass();  This example results in a new MyClass object being created, and the no-arg constructor of MyClass to be called. You will learn what the no-arg constructor is later.  A Java class constructor initializes instances (objects) of that class. Typically, the constructor initializes the fields of the object that need initialization. Java constructors can also take parameters, so fields can be initialized in the object at creation time. Defining a Constructor in Java Here is a simple Java constructor declaration example. The example shows a very simple Java class with a single constructor.  public class MyClass {  public MyClass() {  }  }  The constructor is this part:  public MyClass() {  }  The first part of a Java constructor declaration is an access modifier. The access modifier have the same meanings as for methods and fields. They determine what classes can access (call) the constructor. | | | |