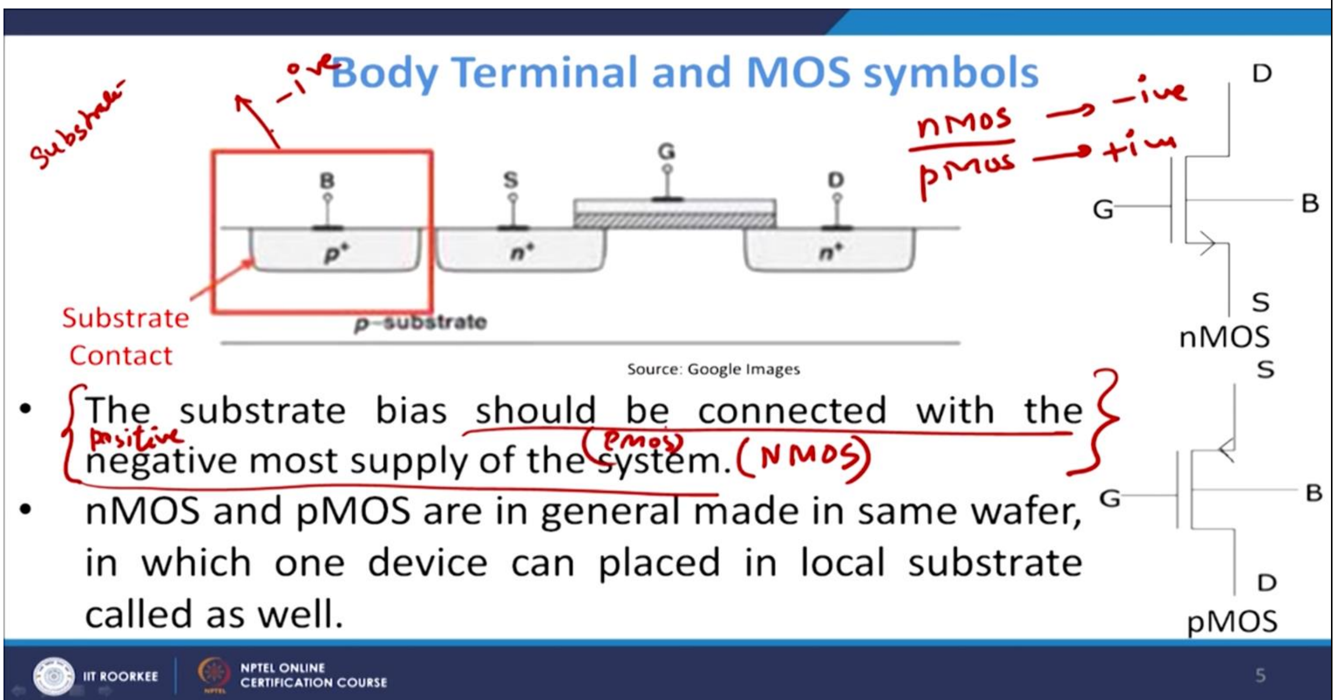
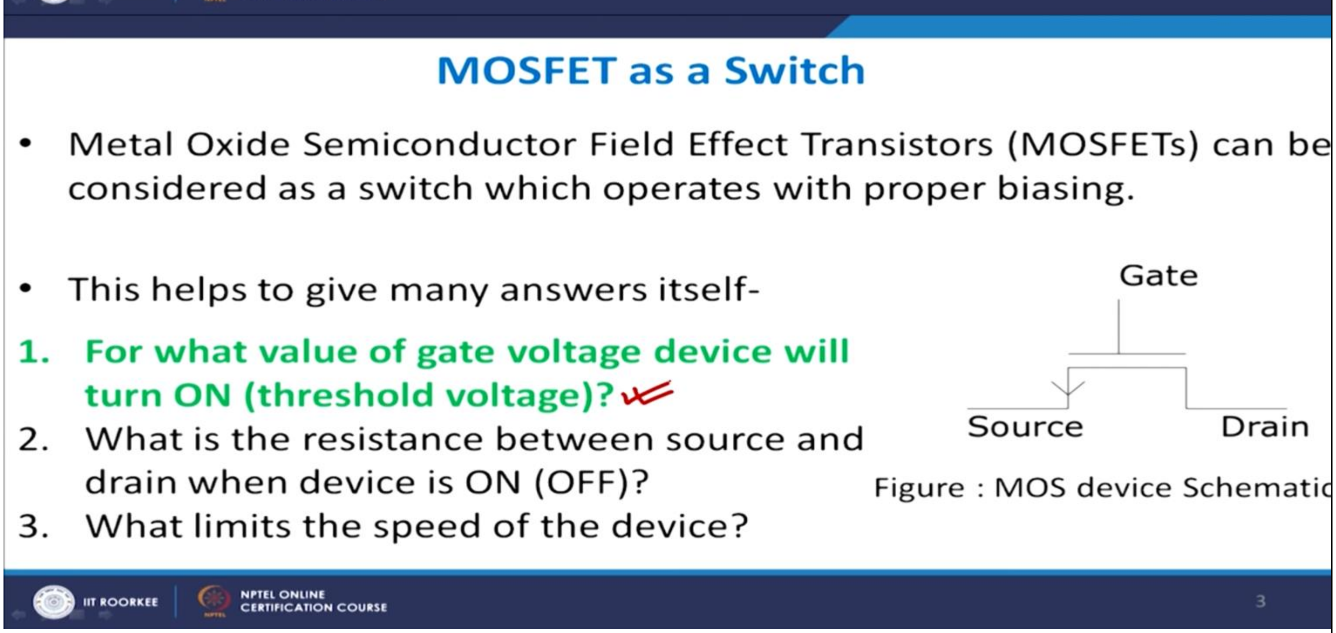
DAILY ASSESSMENT FORMAT

Date:	11th June 2020	**Name:**	Soundarya NA
Course:	VLSI	**USN:**	4AL16EC077
Topic:	VLSI	**Semester & Section:**	8th - B

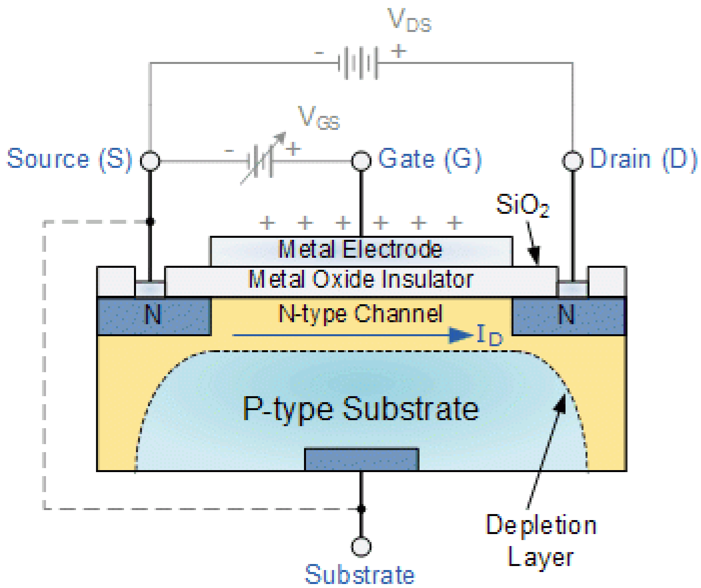
FORENOON SESSION DETAILS
Image of session





Report:

MOSFET's operate the same as JFET's but have a gate terminal that is electrically isolated from the conductive channel.



As well as the Junction Field Effect Transistor (JFET), there is another type of Field Effect Transistor available whose Gate input is electrically insulated from the main current carrying channel and is therefore called an Insulated Gate Field Effect Transistor.

The most common type of insulated gate FET which is used in many different types of electronic circuits is called the Metal Oxide Semiconductor Field Effect Transistor or MOSFET for short.

The IGFET or MOSFET is a voltage controlled field effect transistor that differs from a JFET in that it has a "Metal Oxide" Gate electrode which is electrically insulated from the main semiconductor n-channel or p-channel by a very thin layer of insulating material usually silicon dioxide, commonly known as glass.

This ultra-thin insulated metal gate electrode can be thought of as one plate of a capacitor. The isolation of the controlling Gate makes the input resistance of the MOSFET extremely high way up in the Mega-ohms (MΩ) region thereby making it almost infinite.

As the Gate terminal is electrically isolated from the main current carrying channel between the drain and source, "NO current flows into the gate" and just like the JFET, the MOSFET also acts like a voltage controlled resistor where the current flowing through the main channel between the Drain and Source is proportional to the input voltage. Also like the JFET, the MOSFETs very high input resistance can easily accumulate large amounts of static charge resulting in the MOSFET becoming easily damaged unless carefully handled or protected.

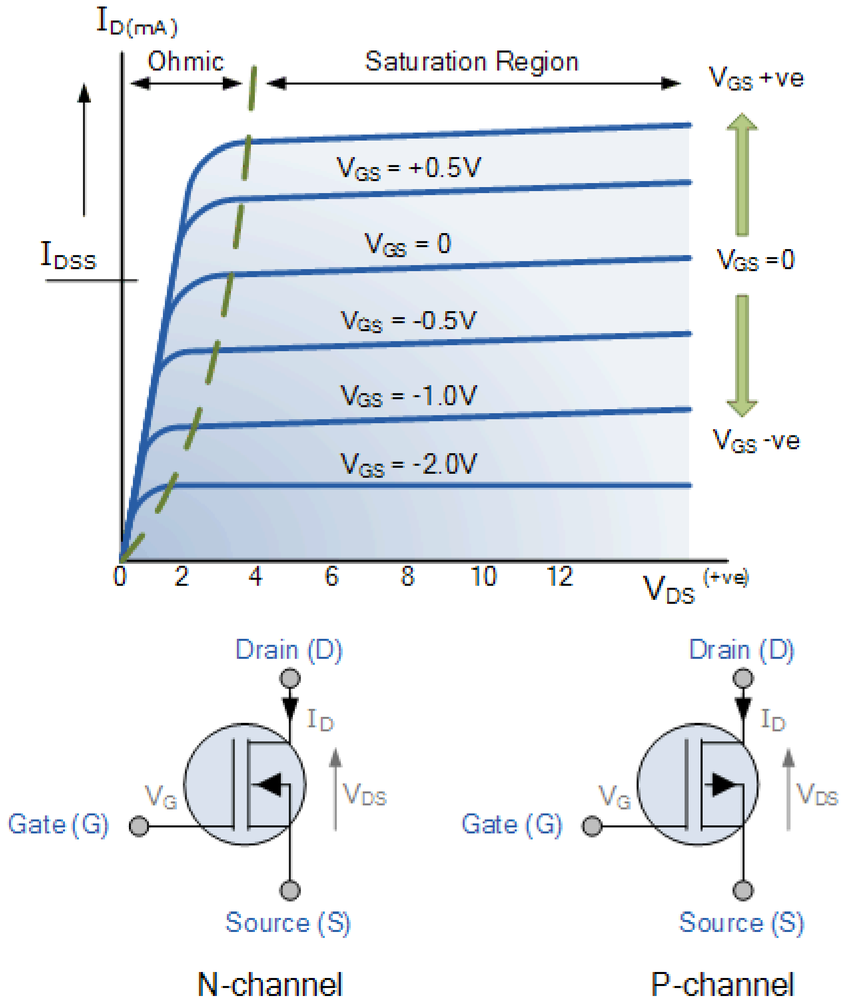
Depletion mode MOSFET:

The Depletion-mode MOSFET, which is less common than the enhancement mode types is normally switched “ON” (conducting) without the application of a gate bias voltage. That is the channel conducts when VGS = 0 making it a “normally-closed” device. The circuit symbol shown above for a depletion MOS transistor uses a solid channel line to signify a normally closed conductive channel.

For the n-channel depletion MOS transistor, a negative gate-source voltage, -VGS will deplete (hence its name) the conductive channel of its free electrons switching the transistor “OFF”. Likewise for a p-channel depletion MOS transistor a positive gate-source voltage, +VGS will deplete the channel of its free holes turning it “OFF”.

In other words, for an n-channel depletion mode MOSFET: +VGS means more electrons and more current. While a -VGS means less electrons and less current. The opposite is also true for the p-channel types. Then the depletion mode MOSFET is equivalent to a “normally-closed” switch.

Depletion-mode N-Channel MOSFET and circuit Symbols



The depletion-mode MOSFET is constructed in a similar way to their JFET transistor counterparts were the drain-source channel is inherently conductive with the electrons and holes already present within the n-type or p-type channel. This doping of the channel produces a conducting path of low resistance between the Drain and Source with zero Gate bias.

Enhancement mode MOSFET:

The more common Enhancement-mode MOSFET or eMOSFET, is the reverse of the depletion-mode type. Here the conducting channel is lightly doped or even undoped making it non-conductive. This results in the device being normally “OFF” (non-conducting) when the gate bias voltage, VGS is equal to zero. The circuit symbol shown above for an enhancement MOS transistor uses a broken channel line to signify a normally open non-conducting channel.

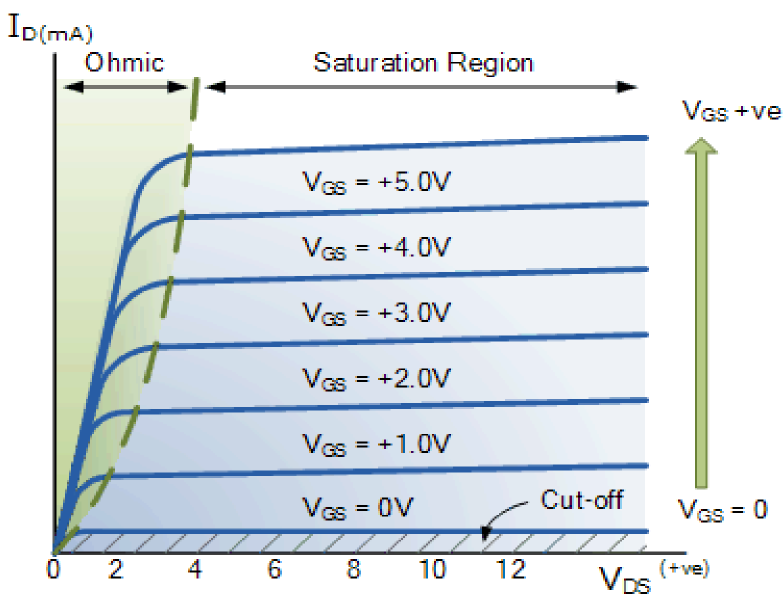
For the n-channel enhancement MOS transistor a drain current will only flow when a gate voltage (VGS) is applied to the gate terminal greater than the threshold voltage (VTH) level in which conductance takes place making it a transconductance device.

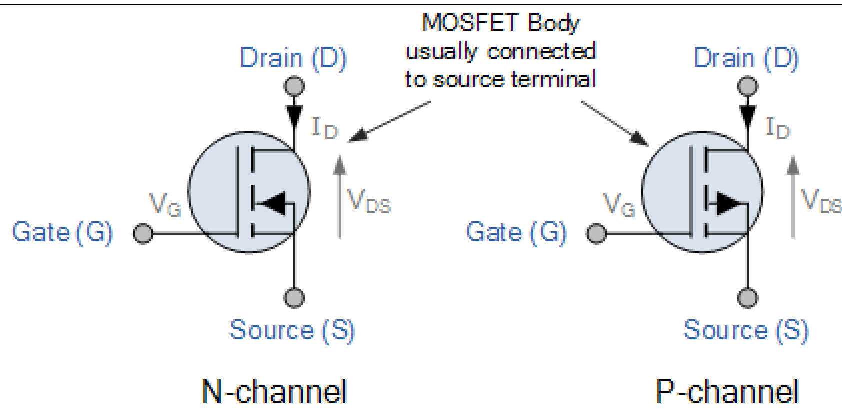
The application of a positive (+ve) gate voltage to a n-type eMOSFET attracts more electrons towards the oxide layer around the gate thereby increasing or enhancing (hence its name) the thickness of the channel allowing more current to flow. This is why this kind of transistor is called an enhancement mode device as the application of a gate voltage enhances the channel.

Increasing this positive gate voltage will cause the channel resistance to decrease further causing an increase in the drain current, ID through the channel. In other words, for an n-channel enhancement mode MOSFET: +VGS turns the transistor "ON", while a zero or -VGS turns the transistor "OFF". Thus the enhancement-mode MOSFET is equivalent to a "normally-open" switch.

The reverse is true for the p-channel enhancement MOS transistor. When VGS = 0 the device is "OFF" and the channel is open. The application of a negative (-ve) gate voltage to the p-type eMOSFET enhances the channels conductivity turning it "ON". Then for an p-channel enhancement mode MOSFET: +VGS turns the transistor "OFF", while -VGS turns the transistor "ON".

Enhancement-mode N-Channel MOSFET and Circuit Symbols:





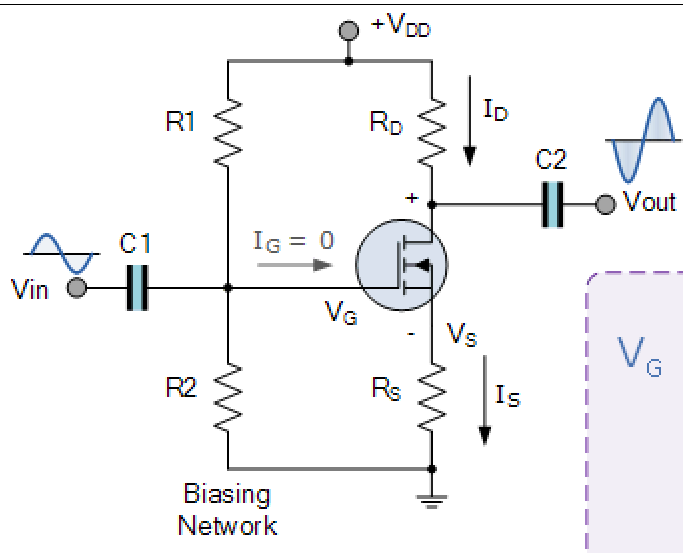
Enhancement-mode MOSFETs make excellent electronics switches due to their low “ON” resistance and extremely high “OFF” resistance as well as their infinitely high input resistance due to their isolated gate. Enhancement-mode MOSFETs are used in integrated circuits to produce CMOS type Logic Gates and power switching circuits in the form of as PMOS (P-channel) and NMOS (N-channel) gates. CMOS actually stands for Complementary MOS meaning that the logic device has both PMOS and NMOS within its design.

The MOSFET Amplifier:

Just like the previous Junction Field Effect transistor, MOSFETs can be used to make single stage class “A” amplifier circuits with the enhancement mode n-channel MOSFET common source amplifier being the most popular circuit. Depletion mode MOSFET amplifiers are very similar to the JFET amplifiers, except that the MOSFET has a much higher input impedance.

This high input impedance is controlled by the gate biasing resistive network formed by R1 and R2. Also, the output signal for the enhancement mode common source MOSFET amplifier is inverted because when VG is low the transistor is switched “OFF” and VD (Vout) is high. When VG is high the transistor is switched “ON” and VD (Vout) is low as shown.

Enhancement-mode N-Channel MOSFET Amplifier:



$$V_G = \left(\frac{R2}{R1+R2}\right)V_{DD}$$

$$I_D = \frac{V_S}{R_S}$$

The DC biasing of this common source (CS) MOSFET amplifier circuit is virtually identical to the JFET amplifier. The MOSFET circuit is biased in class A mode by the voltage divider network formed by resistors R1 and R2. The AC input resistance is given as RIN = RG = 1MΩ.

Metal Oxide Semiconductor Field Effect Transistors are three terminal active devices made from different semiconductor materials that can act as either an insulator or a conductor by the application of a small signal voltage.

The MOSFETs ability to change between these two states enables it to have two basic functions: “switching” (digital electronics) or “amplification” (analogue electronics). Then MOSFETs have the ability to operate within three different regions:

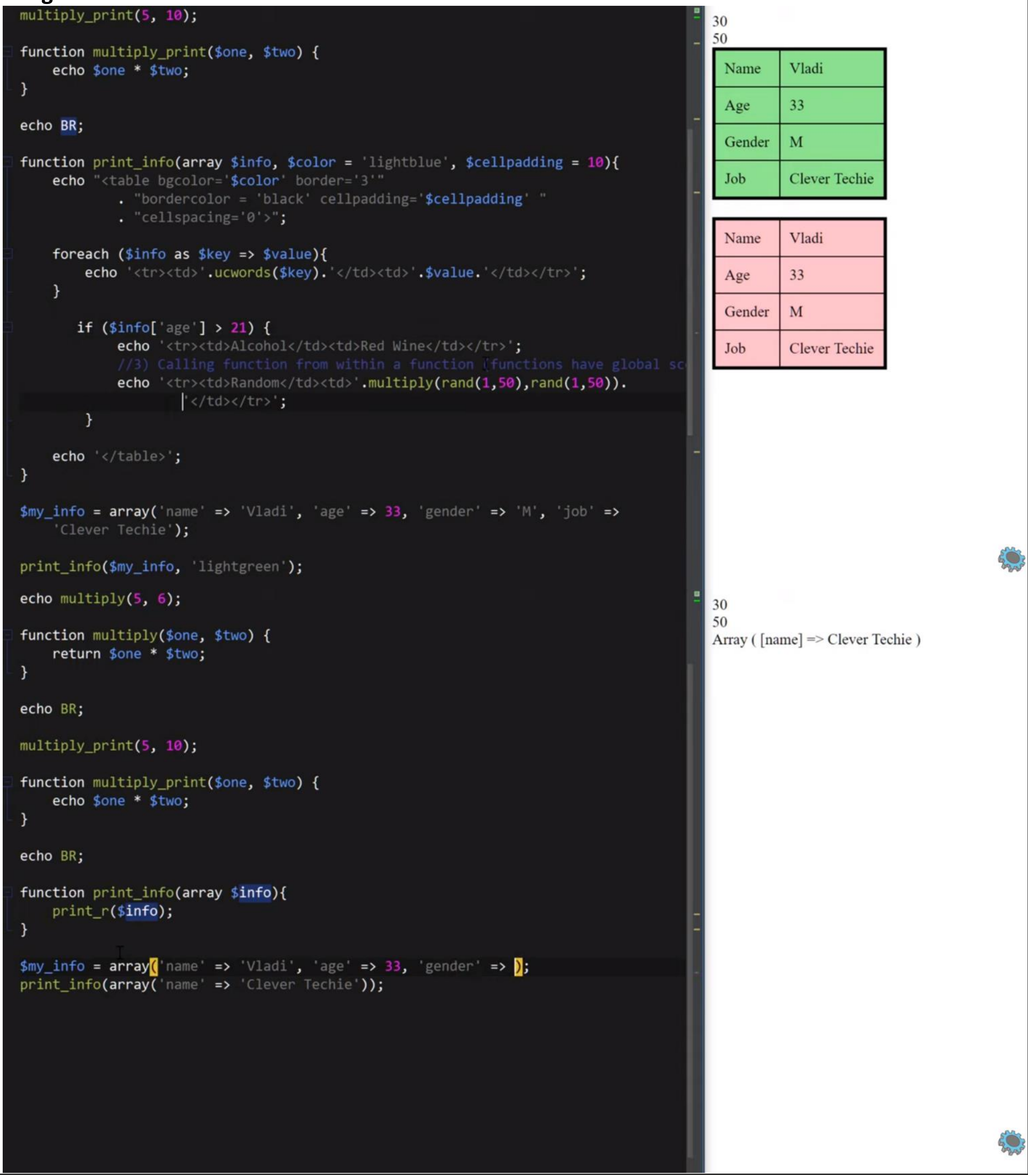
1. Cut-off Region – with VGS < Vthreshold the gate-source voltage is much lower than the transistors threshold voltage so the MOSFET transistor is switched “fully-OFF” thus, ID = 0, with the transistor acting like an open switch regardless of the value of VDS.

2. Linear (Ohmic) Region – with VGS > Vthreshold and VDS < VGS the transistor is in its constant resistance region behaving as a voltage-controlled resistance whose resistive value is determined by the gate voltage, VGS level.

3. Saturation Region – with VGS > Vthreshold and VDS > VGS the transistor is in its constant current region and is therefore “fully-ON”. The Drain current ID = Maximum with the transistor acting as a closed switch.

Date:	11th June 2020	Name:	Soundarya NA
Course:	UDEMY	USN:	4AL16EC077
Topic:	MySQL	Semester & Section:	8th - B

Image:



Report:

PHP Functions:

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

Creating a 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.

Code:

```
<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>
```

Output:

You are really a nice person, Have a nice time!

PHP Functions with parameter:

```
<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>
```

Output:

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.

Code:

```
<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>
```

Output:

Original Value is 10

Original Value is 16

PHP Functions returning values:

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.

Code:

```
<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>
```

Output:

Returned value from the function: 30

Setting default values for function parameter:

Code:

```
<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>
```

Output:

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 behavior.

Code:

```
<html>

  <head>
    <title>Dynamic Function Calls</title>
  </head>

  <body>

    <?php
      function sayHello() {
        echo "Hello<br />";
      }

      $function_holder = "sayHello";
      $function_holder();
    ?>

  </body>
</html>
```

Output:

Hello

Using external files in images:

PHP is capable of utilizing SSI to include an external file in the file being executed. Two commands that do this are INCLUDE () and REQUIRE (). The difference between them is that when placed within a false conditional statement, the INCLUDE is not pulled but the REQUIRE is pulled and ignored. This means that in a conditional statement, it is faster to use INCLUDE. These commands are phrased as follows:

```
INCLUDE 'http://www.yoursite.com/path/to/file.php' ;
//or
REQUIRE 'http://www.yoursite.com/path/to/file.php' ;
```

Some of the most common uses for these commands include holding variables that are used across multiple files or holding headers and footers. If an entire site's layout is housed in external files called with SSI, any changes to site design need only be made to these files and the entire site changes accordingly.

Pulling the file:

```
//variables.php
 $name = 'Loretta';
 $age = '27';
 ?>
//report.php
 include 'variables.php';
 // or you can use the full path; include 'http://www.yoursite.com/folder/folder2/variables.php';

 print $name . " is my name and I am " . $age . " years old.";
 ?>
";
 //The line below will work because $name is GLOBAL
```

```
print "I like my name, " . $name;
print "
";
//The next line will NOT work because $age is NOT defined as global


print "I like being " . $age . " years old.";
?>
```