# DAILY ASSESSMENT

|  |  |  |  |
| --- | --- | --- | --- |
| Date: | 12/06/2020 | Name: | Chesmi B R |
| Course: | VLSI | USN: | 4AL16EC100 |
| Topic: | |  | | --- | | **CMOS inverter basics** | | Semester & Section: | 8TH SEM & A Section |
| Github Repository: | chesmibr |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **FORENOON SESSION DETAILS**       Report: CMOS inverters (Complementary NOSFET Inverters) are some of the most widely used and adaptable MOSFET inverters used in chip design. They operate with very little power loss and at relatively high speed. Furthermore, the CMOS inverter has good logic buffer characteristics, in that, its noise margins in both low and high states are large.  This short description of CMOS inverters gives a basic understanding of the how a CMOS inverter works. It will cover input/output characteristics, MOSFET states at different input voltages, and power losses due to electrical current.  A CMOS inverter contains a PMOS and a NMOS transistor connected at the drain and gate terminals, a supply voltage VDD at the PMOS source terminal, and a ground connected at the NMOS source terminal, were VIN is connected to the gate terminals and VOUT is connected to the drain terminals.(See diagram). It is important to notice that the CMOS does not contain any resistors, which makes it more power efficient that a regular resistor-MOSFET inverter.As the voltage at the input of the CMOS device varies between 0 and 5 volts, the state of the NMOS and PMOS varies accordingly. If we model each transistor as a simple switch activated by VIN, the inverter’s operations can be seen very easily:    **Transistor "switch model"**  The switch model of the MOSFET transistor is defined as follows:   |  |  |  | | --- | --- | --- | | **MOSFET** | **Condition on MOSFET** | **Stateof MOSFET** | | NMOS | Vgs<Vtn | OFF | | NMOS | Vgs>Vtn | ON | | PMOS | Vsg<Vtp | OFF | | PMOS | Vsg>Vtp | ON |   When VIN is low, the NMOS is "off", while the PMOS stays "on": instantly charging VOUT to logic high. When Vin is high, the NMOS is "on and the PMOS is "on: draining the voltage at VOUT to logic low.  This model of the CMOS inverter helps to describe the inverter conceptually, but does not accurately describe the voltage transfer characteristics to any extent. A more full description employs more calculations and more device states.  **Multiple state transistor model**  The multiple state transistor model is a very accurate way to model the CMOS inverter. It reduces the states of the MOSFET into three modes of operation: Cut-Off, Linear, and Saturated: each of which have a different dependence on Vgs and Vds. The formulas which govern the state and the current in that given state is given by the following tabel:  In order to simplify calculations, I have made use of an internet circuit simulation device called "MoHAT." This tool allows the user to simulate circuits containing a few transistors in a simple and visually appealing way. The circuits shown below show the state of each transistor (black for cut-off, red for linear, and green for saturation) accompanied by the voltage transfer characteristic curve (VOUT vs. VIN). The vertical line plotted on the VTC corresponds to the value of VIN on the circuit diagram. The following series of diagrams depict the CMOS inverter in varying input voltages ranging from low to high in ascending order.   |  |  |  | | --- | --- | --- | |  | **Table of figures** |  | | figure | mode operation | Logic output | | 1 | VIN < VIL | High | | 2 | VIN < VIL | High | | 3 | VIL < VIN <VIH | Undetermind | | 4 | VIN > VIH | Low | | 5 | VIN > VIH | Low |           **Power dissapation analysis of CMOS inverter**  As I mentioned before, the CMOS inverter shows very low power dissipation when in proper operation. In fact, the power dissipation is virtually zero when operating close to VOH and VOL. The following graph shows the drain to source current (effectively the overall current of the inverter) of the NMOS as a function of input voltage. Note that the current in the far left and right regions (low and high VIN respectively) have low current, and the peak current in the middle is only .232mA (a 1.16mW power dissipation). |

**DAILY ASSESSMENT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **12/06/2020** | **Name:** | **Chesmi B R** |
| **Course:** | **Begineer PHP and SQL** | **USN:** | **4AL16EC100** |
| **Topic:** | **Email with PHP**  **Real life PHP introduction**  **About the author** | **Semester & Section:** | **8TH SEM & A Section** |
| **Github Repository:** | **chesmibr** |  |  |

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

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Report**-  PHP must be configured correctly in the **php.ini** file with the details of how your system sends email. Open php.ini file available in **/etc/** directory and find the section headed **[mail function]**.  Windows users should ensure that two directives are supplied. The first is called SMTP that defines your email server address. The second is called sendmail\_from which defines your own email address.  The configuration for Windows should look something like this −  [mail function]  ; For Win32 only.  SMTP = smtp.secureserver.net  ; For win32 only  sendmail\_from = webmaster@tutorialspoint.com  Linux users simply need to let PHP know the location of their **sendmail** application. The path and any desired switches should be specified to the sendmail\_path directive.  The configuration for Linux should look something like this −  [mail function]  ; For Win32 only.  SMTP =  ; For win32 only  sendmail\_from =  ; For Unix only  sendmail\_path = /usr/sbin/sendmail -t -i  Now you are ready to go − Sending plain text email PHP makes use of **mail()** function to send an email. This function requires three mandatory arguments that specify the recipient's email address, the subject of the the message and the actual message additionally there are other two optional parameters.  mail( to, subject, message, headers, parameters );  Here is the description for each parameters.   |  |  | | --- | --- | | **Sr.No** | **Parameter & Description** | | 1 | **To**  Required. Specifies the receiver / receivers of the email | | 2 | **subject**  Required. Specifies the subject of the email. This parameter cannot contain any newline characters | | 3 | **message**  Required. Defines the message to be sent. Each line should be separated with a LF (\n). Lines should not exceed 70 characters | | 4 | **headers**  Optional. Specifies additional headers, like From, Cc, and Bcc. The additional headers should be separated with a CRLF (\r\n) | | 5 | **parameters**  Optional. Specifies an additional parameter to the send mail program |   As soon as the mail function is called PHP will attempt to send the email then it will return true if successful or false if it is failed.  Multiple recipients can be specified as the first argument to the mail() function in a comma separated list. Sending HTML email When you send a text message using PHP then all the content will be treated as simple text. Even if you will include HTML tags in a text message, it will be displayed as simple text and HTML tags will not be formatted according to HTML syntax. But PHP provides option to send an HTML message as actual HTML message.  While sending an email message you can specify a Mime version, content type and character set to send an HTML email. Example Following example will send an HTML email message to xyz@somedomain.com copying it to afgh@somedomain.com. You can code this program in such a way that it should receive all content from the user and then it should send an email.  <html>    <head>  <title>Sending HTML email using PHP</title>  </head>    <body>    <?php  $to = "xyz@somedomain.com";  $subject = "This is subject";    $message = "<b>This is HTML message.</b>";  $message .= "<h1>This is headline.</h1>";    $header = "From:abc@somedomain.com \r\n";  $header .= "Cc:afgh@somedomain.com \r\n";  $header .= "MIME-Version: 1.0\r\n";  $header .= "Content-type: text/html\r\n";    $retval = mail ($to,$subject,$message,$header);    if( $retval == true ) {  echo "Message sent successfully...";  }else {  echo "Message could not be sent...";  }  ?>    </body>  </html> Sending attachments with email To send an email with mixed content requires to set **Content-type** header to **multipart/mixed**. Then text and attachment sections can be specified within **boundaries**.  A boundary is started with two hyphens followed by a unique number which can not appear in the message part of the email. A PHP function **md5()** is used to create a 32 digit hexadecimal number to create unique number. A final boundary denoting the email's final section must also end with two hyphens.  <?php  // request variables // important  $from = $\_REQUEST["from"];  $emaila = $\_REQUEST["emaila"];  $filea = $\_REQUEST["filea"];    if ($filea) {  function mail\_attachment ($from , $to, $subject, $message, $attachment){  $fileatt = $attachment; // Path to the file  $fileatt\_type = "application/octet-stream"; // File Type    $start = strrpos($attachment, '/') == -1 ?  strrpos($attachment, '//') : strrpos($attachment, '/')+1;    $fileatt\_name = substr($attachment, $start,  strlen($attachment)); // Filename that will be used for the  file as the attachment    $email\_from = $from; // Who the email is from  $subject = "New Attachment Message";    $email\_subject = $subject; // The Subject of the email  $email\_txt = $message; // Message that the email has in it  $email\_to = $to; // Who the email is to    $headers = "From: ".$email\_from;  $file = fopen($fileatt,'rb');  $data = fread($file,filesize($fileatt));  fclose($file);    $msg\_txt="\n\n You have recieved a new attachment message from $from";  $semi\_rand = md5(time());  $mime\_boundary = "==Multipart\_Boundary\_x{$semi\_rand}x";  $headers .= "\nMIME-Version: 1.0\n" . "Content-Type: multipart/mixed;\n" . "  boundary=\"{$mime\_boundary}\"";    $email\_txt .= $msg\_txt;    $email\_message .= "This is a multi-part message in MIME format.\n\n" .  "--{$mime\_boundary}\n" . "Content-Type:text/html;  charset = \"iso-8859-1\"\n" . "Content-Transfer-Encoding: 7bit\n\n" .  $email\_txt . "\n\n";    $data = chunk\_split(base64\_encode($data));    $email\_message .= "--{$mime\_boundary}\n" . "Content-Type: {$fileatt\_type};\n" .  " name = \"{$fileatt\_name}\"\n" . //"Content-Disposition: attachment;\n" .  //" filename = \"{$fileatt\_name}\"\n" . "Content-Transfer-Encoding:  base64\n\n" . $data . "\n\n" . "--{$mime\_boundary}--\n";    $ok = mail($email\_to, $email\_subject, $email\_message, $headers);    if($ok) {  echo "File Sent Successfully.";  unlink($attachment); // delete a file after attachment sent.  }else {  die("Sorry but the email could not be sent. Please go back and try again!");  }  }  move\_uploaded\_file($\_FILES["filea"]["tmp\_name"],  'temp/'.basename($\_FILES['filea']['name']));    mail\_attachment("$from", "youremailaddress@gmail.com",  "subject", "message", ("temp/".$\_FILES["filea"]["name"]));  }  ?>  <html>  <head>    <script language = "javascript" type = "text/javascript">  function CheckData45() {  with(document.filepost) {  if(filea.value ! = "") {  document.getElementById('one').innerText =  "Attaching File ... Please Wait";  }  }  }  </script>    </head>  <body>    <table width = "100%" height = "100%" border = "0"  cellpadding = "0" cellspacing = "0">  <tr>  <td align = "center">  <form name = "filepost" method = "post"  action = "file.php" enctype = "multipart/form-data" id = "file">    <table width = "300" border = "0" cellspacing = "0"  cellpadding = "0">    <tr valign = "bottom">  <td height = "20">Your Name:</td>  </tr>    <tr>  <td><input name = "from" type = "text"  id = "from" size = "30"></td>  </tr>    <tr valign = "bottom">  <td height = "20">Your Email Address:</td>  </tr>    <tr>  <td class = "frmtxt2"><input name = "emaila"  type = "text" id = "emaila" size = "30"></td>  </tr>    <tr>  <td height = "20" valign = "bottom">Attach File:</td>  </tr>    <tr valign = "bottom">  <td valign = "bottom"><input name = "filea"  type = "file" id = "filea" size = "16"></td>  </tr>    <tr>  <td height = "40" valign = "middle"><input  name = "Reset2" type = "reset" id = "Reset2" value = "Reset">  <input name = "Submit2" type = "submit"  value = "Submit" onClick = "return CheckData45()"></td>  </tr>  </table>    </form>    <center>  <table width = "400">    <tr>  <td id = "one">  </td>  </tr>    </table>  </center>    </td>  </tr>  </table>    </body>  </html> |