**ASSESSMENT 22**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | 11-06-2020 | **Name:** | Sheela Golasangi |
| **Course:** | VLSI | **USN:** | 4AL16EC068 |
| **Topic:** | MOS transistor basics-II & III | **Semester & Section:** | VIII  ‘B’ |
| **Github Repository:** | Sheela-Course |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| A **current–voltage characteristic** or **I–V curve** (current–voltage curve) is a relationship, typically represented as a [chart](https://en.wikipedia.org/wiki/Chart) or graph, between the [electric current](https://en.wikipedia.org/wiki/Electric_current) through a circuit, device, or material, and the corresponding [voltage](https://en.wikipedia.org/wiki/Voltage), or [potential difference](https://en.wikipedia.org/wiki/Potential_difference) across it. In [electronics](https://en.wikipedia.org/wiki/Electronics), the relationship between the direct [current](https://en.wikipedia.org/wiki/Electric_current) ([DC](https://en.wikipedia.org/wiki/Direct_current)) through an [electronic device](https://en.wikipedia.org/wiki/Electronic_device) and the DC [voltage](https://en.wikipedia.org/wiki/Voltage) across its terminals is called a current–voltage characteristic of the device. [Electronic engineers](https://en.wikipedia.org/wiki/Electronic_engineering) use these charts to determine basic parameters of a device and to model its behavior in an [electrical circuit](https://en.wikipedia.org/wiki/Electrical_circuit). These characteristics are also known as I–V curves, referring to the standard symbols for current and voltage.  In [electronic components](https://en.wikipedia.org/wiki/Electronic_component) with more than two terminals, such as [vacuum tubes](https://en.wikipedia.org/wiki/Vacuum_tube) and [transistors](https://en.wikipedia.org/wiki/Transistor), the current-voltage relationship at one pair of terminals may depend on the current or voltage on a third terminal. This is usually displayed on a more complex current–voltage graph with multiple curves, each one representing the current-voltage relationship at a different value of current or voltage on the third terminal.[[1]](https://en.wikipedia.org/wiki/Current%E2%80%93voltage_characteristic#cite_note-1)  For example the diagram at right shows a family of I–V curves for a [MOSFET](https://en.wikipedia.org/wiki/MOSFET#Modes_of_operation) as a function of drain voltage with overvoltage (*VGS − Vth*) as a parameter.  The simplest I–V curve is that of a [resistor](https://en.wikipedia.org/wiki/Resistor), which according to [Ohm's law](https://en.wikipedia.org/wiki/Ohm%27s_law) exhibits a [linear](https://en.wikipedia.org/wiki/Linear) relationship between the applied voltage and the resulting [electric current](https://en.wikipedia.org/wiki/Electric_current); the current is proportional to the voltage, so the I–V curve is a straight line through the [origin](https://en.wikipedia.org/wiki/Origin_(mathematics)) with positive [slope](https://en.wikipedia.org/wiki/Slope_(mathematics)). The [reciprocal](https://en.wikipedia.org/wiki/Reciprocal_(mathematics)) of the slope is equal to the [resistance](https://en.wikipedia.org/wiki/Electrical_resistance).  The I–V curve of an electrical component can be measured with an instrument called a [curve tracer](https://en.wikipedia.org/wiki/Curve_tracer). The [transconductance](https://en.wikipedia.org/wiki/Transconductance" \o "Transconductance) and [Early voltage](https://en.wikipedia.org/wiki/Early_voltage) of a [transistor](https://en.wikipedia.org/wiki/Transistor) are examples of parameters traditionally measured from the device's I–V curve.  The **subthreshold slope** is a feature of a [MOSFET](https://en.wikipedia.org/wiki/MOSFET)'s [current–voltage characteristic](https://en.wikipedia.org/wiki/Current%E2%80%93voltage_characteristic).  In the [subthreshold](https://en.wikipedia.org/wiki/Subthreshold_conduction" \o "Subthreshold conduction) region, the [drain](https://en.wikipedia.org/wiki/Field-effect_transistor#Basic_information) current behaviour – though being controlled by the [gate](https://en.wikipedia.org/wiki/Field-effect_transistor#Basic_information) terminal – is similar to the exponentially decreasing current of a [forward biased diode](https://en.wikipedia.org/wiki/Diode#Current.E2.80.93voltage_characteristic). Therefore a plot of drain current versus gate voltage with drain, [source](https://en.wikipedia.org/wiki/Field-effect_transistor#Basic_information), and [bulk](https://en.wikipedia.org/wiki/Field-effect_transistor#More_about_terminals) voltages fixed will exhibit approximately log linear behaviour in this MOSFET operating regime. Its slope is the subthreshold slope.  The subthreshold slope is also the [reciprocal value](https://en.wikipedia.org/wiki/Reciprocal_(mathematics)) of the **subthreshold swing** *Ss-th* which is usually given as:  {\displaystyle S\_{s-th}=\ln(10){kT \over q}\left(1+{C\_{d} \over C\_{ox}}\right)}{\displaystyle C\_{d}}[depletion layer](https://en.wikipedia.org/wiki/Depletion_region) capacitance  {\displaystyle C\_{ox}}gate-oxide capacitance  {\displaystyle {kT \over q}}thermal Voltage  The minimum subthreshold swing of a conventional device can be found by letting {\displaystyle \textstyle {C\_{d}}\rightarrow 0}and/or {\displaystyle \textstyle {C\_{ox}}\rightarrow \infty }, which yield {\displaystyle S\_{s-th,\min }=\ln(10){kT \over q}}(known as thermionic limit) and 60 mV/dec at room temperature (300 K). A typical experimental subthreshold swing for a scaled MOSFET at room temperature is ~70 mV/dec, slightly degraded due to short-channel MOSFET parasitics.  A *dec* (decade) corresponds to a 10 times increase of the drain current *ID*.  A device characterized by steep subthreshold slope exhibits a faster transition between off (low current) and on (high current) states.  **Short-channel effects** occur when the channel length is the same order of magnitude as the depletion-layer widths of the source and drain junction. In MOSFETs, channel lengths must be greater than the sum of the drain and source depletion widths to avoid edge effects. Otherwise, a number of effects appear.  Among the reported effects cited by a number of researchers at universities around the globe are: 1. “Off-state” leakage current. 2. Impact ionization, in which a charge carrier can be affected by other charge carriers; 3. Velocity saturation/mobility degradation; 4. Drain-induced barrier lowering (DIBL), which is caused by encroachment of the drain depletion region into the channel; 5. Drain punch through, whereby current flows regardless of gate voltage-a phenomenon that can occur if the drain is at high enough voltage compared to the source and the depletion region around the drain extends to the source; 6. Surface scattering; 7. Channel length modulation; 8. Threshold voltage roll-off.  Analysis of MOSFET circuits is based on three possible operating modes: cutoff, triode (aka linear), and saturation. (The subthreshold region is a fourth mode, but we don’t need to worry about that for this article.)  In cutoff, the gate-to-source voltage is not greater than the threshold voltage, and the MOSFET is inactive.  In triode, the gate-to-source voltage is high enough to allow current flow from drain to source, and the nature of the induced channel is such that the magnitude of the drain current is influenced by the gate-to-source voltage and the drain-to-source voltage. As the drain-to-source voltage increases, the triode region transitions to the saturation region, in which drain current is (ideally) independent of drain-to-source voltage and thus influenced only by the physical characteristics of the FET and the gate-to-source voltage.  The saturation-region relationship between gate-to-source voltage (VGS) and drain current (ID) is expressed as follows:  ID=12μnCoxWL(VGS−VTH)2ID=12μnCoxWL(VGS−VTH)2    The transition to saturation mode occurs because the channel gets “pinched off” at the drain end:        Unfortunately, the “pinching off” isn’t the end of the influence exerted by the drain-to-source voltage. Further increases continue to affect the channel because the pinch-off point moves closer to the source:        The resistance of the channel is proportional to its width-to-length ratio; reducing the length leads to decreased resistance and hence higher current flow. Thus, channel-length modulation means that the saturation-region drain current will increase slightly as the drain-to-source voltage increases.  So we need to modify the saturation-region drain-current expression to account for channel-length modulation. We do this by incorporating the incremental channel-length reduction into the original expression: |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | 11-06-2020 | **Name:** | Sheela Golasangi |
| **Course:** | Java Tutorial for Complete Beginners | **USN:** | 4AL16EC068 |
| **Topic:** | Programming core java   1. The toString Method 2. Inheritance 3. Packages 4. Interfaces 5. Public, Private, Protected 6. Polymorphism 7. Encapsulation and the API Docs 8. Casting Numerical Values 9. Upcasting and Downcasting 10. Using Generics | **Semester & Section:** | VIII  ‘B’ |
| **Github Repository:** | Sheela-Course |  |  |
| **AFTERNOON SESSION DETAILS** | | | |
| C:\Users\india\Pictures\Screenshots\Screenshot (400).pngC:\Users\india\Pictures\Screenshots\Screenshot (402).pngJava toString() method If you want to represent any object as a string, **toString() method** comes into existence.  The toString() method returns the string representation of the object.  If you print any object, java compiler internally invokes the toString() method on the object. So overriding the toString() method, returns the desired output, it can be the state of an object etc. depends on your implementation. Advantage of Java toString() method By overriding the toString() method of the Object class, we can return values of the object, so we don't need to write much code. Understanding problem without toString() method Let's see the simple code that prints reference.   1. **class** Student{ 2. **int** rollno; 3. String name; 4. String city; 6. Student(**int** rollno, String name, String city){ 7. **this**.rollno=rollno; 8. **this**.name=name; 9. **this**.city=city;    } Private access modifier The scope of private modifier is limited to the class only.   1. Private Data members and methods are only accessible within the class 2. Class and [Interface](https://beginnersbook.com/2013/05/java-interface/) cannot be declared as private 3. If a class has [private constructor](https://beginnersbook.com/2013/12/java-private-constructor-example/) then you cannot create the object of that class from outside of the class.   Let’s see an example to understand this: Private access modifier example in java This example throws compilation error because we are trying to access the private data member and method of class ABC in the class Example. The private data member and method are only accessible within the class.  class ABC{  privatedoublenum=100;  privateint square(int a){  return a\*a;  }  }  publicclassExample{  publicstaticvoid main(Stringargs[]){  ABC obj=new ABC();  System.out.println(obj.num);  System.out.println(obj.square(10));  }  } Protected Access Modifier Protected data member and method are only accessible by the classes of the same package and the subclasses present in any package. You can also say that the protected access modifier is similar to default access modifier with one exception that it has visibility in sub classes. Classes cannot be declared protected. This access modifier is generally used in a parent child relationship. Protected access modifier example in Java In this example the class Test which is present in another package is able to call the addTwoNumbers() method, which is declared protected. This is because the Test class extends class Addition and the protected modifier allows the access of protected members in subclasses (in any packages). **Addition.java**  packageabcpackage;  publicclassAddition{  protectedintaddTwoNumbers(int a,int b){  returna+b;  }  }  **Test.java**  packagexyzpackage;  importabcpackage.\*;  classTestextendsAddition{  publicstaticvoid main(Stringargs[]){  Testobj=newTest();  System.out.println(obj.addTwoNumbers(11,22));  }  } Public access modifier The members, methods and classes that are declared public can be accessed from anywhere. This modifier doesn’t put any restriction on the access. public access modifier example in java Lets take the same example that we have seen above but this time the method addTwoNumbers() has public modifier and class Test is able to access this method without even extending the Addition class. This is because public modifier has visibility everywhere. Addition.java  packageabcpackage;  publicclassAddition{  publicintaddTwoNumbers(int a,int b){  returna+b;  }  }  Test.java  packagexyzpackage;  importabcpackage.\*;  classTest{  publicstaticvoid main(Stringargs[]){  Additionobj=newAddition();  System.out.println(obj.addTwoNumbers(100,1));  }  }    **public** **static** **void** main(String args[]){     Student s1=**new** Student(101,"Raj","lucknow");     Student s2=**new** Student(102,"Vijay","ghaziabad");       System.out.println(s1);//compiler writes here s1.toString()     System.out.println(s2);//compiler writes here s2.toString()   }  } | | | |