**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **4-6-2020** | **Name:** | **Rasika Patil** |
| **Course:** | **Digital design using HDL** | **USN:** | **4AL16EC057** |
| **Topic:** | **Hardware modeling using Verilog.**  **FPGA & ASIC interview questions** | **Semester & Section:** | **8th B** |
| **Github Repository:** | **Rasika B Patil** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **C:\Users\User\Downloads\WhatsApp Image 2020-06-04 at 7.46.32 PM (1).jpeg**  **C:\Users\User\Downloads\WhatsApp Image 2020-06-04 at 7.46.32 PM (2).jpeg**  **C:\Users\User\Downloads\WhatsApp Image 2020-06-04 at 7.46.32 PM (3).jpeg**  **C:\Users\User\Downloads\WhatsApp Image 2020-06-04 at 7.46.32 PM (4).jpeg**  **C:\Users\User\Downloads\WhatsApp Image 2020-06-04 at 7.46.32 PM (5).jpeg**  **C:\Users\User\Downloads\WhatsApp Image 2020-06-04 at 7.46.32 PM (6).jpeg**  **C:\Users\User\Downloads\WhatsApp Image 2020-06-04 at 7.46.32 PM (7).jpeg**  **C:\Users\User\Downloads\WhatsApp Image 2020-06-04 at 7.46.32 PM (8).jpeg**  **TASK FOR THE DAY**  **IMPLIMENT THE T-FLIPFLOP** Design   module tff ( input clk,  input rstn,  input t,  output reg q);    always @ (posedge clk) begin  if (!rstn)  q <= 0;  else  if (t)  q <= ~q;  else  q <= q;  end  endmodule   Testbench   module tb;  reg clk;  reg rstn;  reg t;    tff u0 ( .clk(clk),  .rstn(rstn),  .t(t),  .q(q));    always #5 clk = ~clk;    initial begin  {rstn, clk, t} <= 0;    $monitor ("T=%0t rstn=%0b t=%0d q=%0d", $time, rstn, t, q);  repeat(2) @(posedge clk);  rstn <= 1;    for (integer i = 0; i < 20; i = i+1) begin  reg [4:0] dly = $random;  #(dly) t <= $random;  end  #20 $finish;  end  endmodule |

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **4-6-2020** | **Name:** | **Rasika Patil** |
| **Course:** | **Python** | **USN:** | **4AL16EC057** |
| **Topic:** | **Application 9: build data collector web app** | **Semester & Section:** | **8th B** |
| **Github Repository:** | **Rasika B Patil** |  |  |
| **AFTERNOON SESSION DETAILS** | | | |
| **C:\Users\User\Downloads\WhatsApp Image 2020-06-04 at 8.18.42 PM.jpeg**  **Data collection system** (**DCS**) is a [computer application](https://en.wikipedia.org/wiki/Computer_application) that facilitates the process of [data collection](https://en.wikipedia.org/wiki/Data_collection), allowing specific, structured information to be gathered in a systematic fashion, subsequently enabling [data analysis](https://en.wikipedia.org/wiki/Data_analysis) to be performed on the information. Typically a DCS displays a form that accepts data input from a user and then validates that input prior to committing the data to persistent storage such as a database.  Many computer systems implement data entry forms, but data collection systems tend to be more complex, with possibly many related forms containing detailed user input fields, data validations, and navigation links among the forms.  DCSs can be considered a specialized form of [content management system](https://en.wikipedia.org/wiki/Content_management_system) (CMS), particularly when they allow the information being gathered to be published, edited, modified, deleted, and maintained. Some general-purpose CMSs include features of DCSs. Importance Accurate data collection is essential to many [business processes](https://en.wikipedia.org/wiki/Business_process), to the enforcement of many government [regulations](https://en.wikipedia.org/wiki/Regulation#Social), and to maintaining the integrity of scientific research. Data collection systems are an end-product of [software development](https://en.wikipedia.org/wiki/Software_development). Identifying and categorizing software or a software sub-system as having aspects of, or as actually being a "Data collection system" is very important. This categorization allows encyclopedic knowledge to be gathered and applied in the design and implementation of future systems. In [software design](https://en.wikipedia.org/wiki/Software_design), it is very important to identify generalizations and [patterns](https://en.wikipedia.org/wiki/Software_design_pattern) and to [re-use](https://en.wikipedia.org/wiki/Code_reuse) existing knowledge whenever possible.  **Part Of** **HTML**  **C:\Users\User\Downloads\WhatsApp Image 2020-06-04 at 8.18.42 PM (1).jpeg**  As a long-time member of the documentation team at Scribus, I keep up-to-date with the latest updates of the source so I can help make updates and additions to the documentation. When I recently did a "checkout" using Subversion on a computer I had just upgraded to Fedora 27, I was amazed at how long it took to download the documentation, which consists of HTML pages and associated images. I became concerned that the project's documentation seemed much larger than it should be and suspected that some of the content was "zombie" documentation—HTML files that aren't used anymore and images that have lost all references in the currently used HTML.  I decided to create a project for myself to figure this out. One way to do this is to search for existing image files that aren't used. If I could scan through all the HTML files for image references, then compare that list to the actual image files, chances are I would see a mismatch.  **Part Of** **CSS**  **C:\Users\User\Downloads\WhatsApp Image 2020-06-04 at 8.18.42 PM (2).jpeg**  **Why is CSS necessary?**  CSS separates the content contained in HTML files from how the content should be displayed. It is important to separate the content from the rules for how it should be rendered primarily because it is easier to reuse those rules across many pages. CSS files are also much easier to maintain on large projects than styles embedded within the HTML files. How is CSS retrieved from a web server? The HTML file sent by the web server contains a reference to the CSS file(s) needed to render the content. The web browser requests the CSS file after the HTML file as shown below in a screenshot captured of the Chrome Web Developer Tools network traffic. | | | |