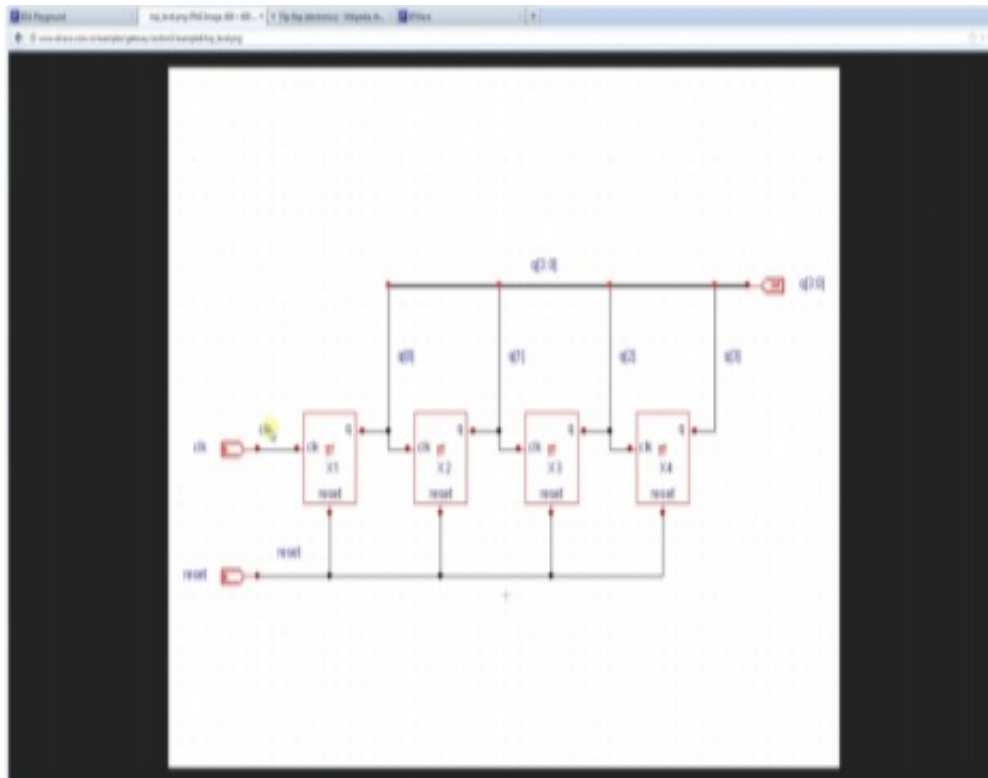


## DAILY ASSESSMENT FORMAT

Date:	03-06-2020	Name:	M V Ramya
Course:	logic design	USN:	4AL17EC045
Topic:	EDA Tool	Semester & Section:	6th sem, A sec
Github Repository:	M V Ramya-045		

### FORENOON SESSION DETAILS





## Verilog Tutorial 1 – Ripple Carry Counter

58,978 views • 11 Nov 2013

288 3 SHARE SAVE ...



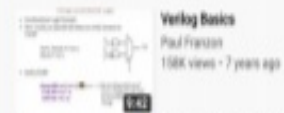
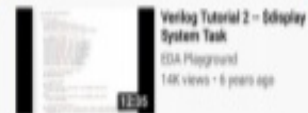
EDA Playground  
5.12K subscribers

In this Verilog tutorial, we implement a basic Ripple Carry Counter design and test using Verilog.

SUBSCRIBE

Up next

AUTOPLAY



Edit with WPS Office

3/6/2020

Task : Implement 4to1 mux using two 2to1 mux using structural modelling style and test module in online / offline compiler

```

entity mux4to1 is
    port (s1, s2, d0, d1, d2, d3 : in std_logic;
          zout : out std_logic);
end mux4to1;

architecture behavioural of mux4to1 is
    component mux2to1
    port (sx1, sx2, dx, d1 : in std_logic;
          z : out std_logic);
    end component

    component or2
    port (a, b : in std_logic;
          c : out std_logic);
    end component

    signal intr1, intr2, intr3, intr4 : std_logic;
begin
    mux1 : mux2to1 port map (s1, s2, d0, d1, intr1);
    mux2 : mux2to1 port map (s1, s2, d2, d3, intr2);
    or1 : or2 port map (intr1, intr2, z-out);
end behavioural;

entity mux is
    port (s1, s2, d0, d1 : in std_logic;
          z1, z2 : in std_logic;
          z : out std_logic);
end mux;

```

architecture behavioural of mux is

```

begin
    z1 <= d0 and (not sx1) and (not sx2);
    z2 <= d1 and (not sx1) and (not sx2);
    z <= z1 or z2;
end behavioural;

entity or-2 is
    port (a, b : in bit;
          c : out bit);
end or-2;

architecture behavioural of or-2 is
begin
    c <= a or b;
end behavioural;

```

Date: 03 June 2020

Course: python

Topic: Application  
7 python for web  
scraping

Name: MV Ramya

USN: 4AL17EC045

Semester & 6th sem Asec  
Section:

#### AFTERNOON SESSION DETAILS

Image of session



Edit with WPS Office

**Udemy** The Python Mega Course: Build 10 Real World Applications

4.5 (1,000+ ratings)

**About this course**

A complete Python course for both beginners and intermediates! Master Python 3 by making 10 amazing Python apps.

**Course content**

- 239. Request Headers
- 240. Loading the Webpage in Python
- 241. Extracting <div> Tags
- 242. Extracting Addresses and Property Details
- 243. Extracting Elements without Unique Identifiers
- 244. Saving the Extracted Data in CSV Files
- 245. Crawling Through Webpages

## Python for Web Scraping?

3161202D

In this session we were taught how to scrap data from real estate website.

- Easy of use : Python is simple today code . You do not have to add semi-colon ";" or curly-braces "{}" anywhere . This makes it less messy and easy to use.
- Large collection of libraries : Python has a huge collection of libraries such as Numpy, Matplotlib, Pandas etc which provides methods and services for various purposes. Hence, it is suitable for web scraping and for further manipulation of extracted data.
- Dynamic typed :- In python, you don't have to define datatype for variables, you can directly use the variables wherever required. This saves time and makes your job faster.
- Easily Understandable Syntax :- Python syntax is easily understandable mainly because reading a python code is very similar to reading a statement in English. It is expressive and easily readable, and the indentation used in python also helps the user to diff btwn scope/blocks in the code.
- Small code, large tasks :- Web scraping is used to save time. But what's the use if you spend more time writing the code? Well, you don't have to. In python, you can write small code to do large tasks.
- Community :- Python community has one of the biggest and most active communities, where you seek help for

--

