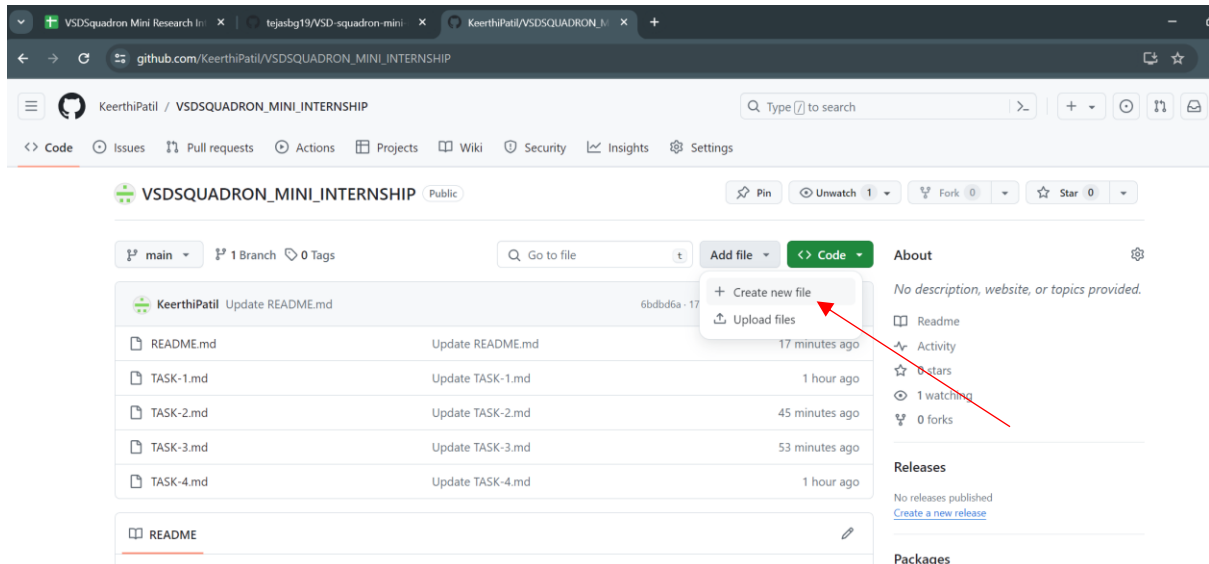


## 1. To insert the image:

Copy on the image and paste it on the place where you want to place.

## 2. To create an file :



Name as **filename.md**

### 3. To make bold

select the sentence and press ctrl + B / add \*\* before and after sentence

VSDSQUADRON\_MINI\_INTERNSHIP / TASK-1.md

EditPreview

1    **\*\*TASK-1:\*\***

2

3    1.Download Oracle Virtual machine and allocated 8GB RAM,100GB HDD

4    2.Installation of Ubuntu 22.04 using Virtual Machine

5    3.Installation of RISC-V GNU Toolchain

6    4.Installation of Yosys

7    5.Installation of iverilog

8    6.Installation of gtkwave

9

EditPreview

**TASK-1:**

1.Download Oracle Virtual machine and allocated 8GB RAM,100GB HDD

2.Installation of Ubuntu 22.04 using Virtual Machine

3.Installation of RISC-V GNU Toolchain

4.Installation of Yosys

5.Installation of iverilog

6 Installation of gtkwave

## 4. To make Italian font

select the sentence and press ctrl + I / add \_ before and after the sentence

EditPreview

```
1  _TASK-1:_
2
3  1.Download Oracle Virtual machine and allocated 8GB RAM,100GB HDD
4  2.Installation of Ubuntu 22.04 using Virtual Machine
5  3.Installation of RISC-V GNU Toolchain
6  4.Installation of Yosys
7  5.Installation of iverilog
8  6.Installation of gtkwave
```

EditPreview

*TASK-1:*

- 1.Download Oracle Virtual machine and allocated 8GB RAM,100GB HDD
- 2.Installation of Ubuntu 22.04 using Virtual Machine
- 3.Installation of RISC-V GNU Toolchain
- 4.Installation of Yosys
- 5.Installation of iverilog
- 6.Installation of gtkwave

## 5. To add heading

Add # and space in front of the heading

[Preview](#) [Code](#) [Blame](#) 63 lines (44 loc) · 2.26 KB

```
1  ✓  # TASK-1:
2
3      1.Download Oracle Virtual machine and allocated 8GB RAM,100GB HDD
4      2.Installation of Ubuntu 22.04 using Virtual Machine
5      3.Installation of RISC-V GNU Toolchain
6      4.Installation of Yosys
7      5.Installation of iverilog
8      6.Installation of gtkwave
```

[Preview](#) [Code](#) [Blame](#) 63 lines (44 loc) · 2.26 KB

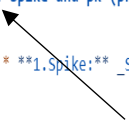
### TASK-1:

- 
- 1.Download Oracle Virtual machine and allocated 8GB RAM,100GB HDD
  - 2.Installation of Ubuntu 22.04 using Virtual Machine
  - 3.Installation of RISC-V GNU Toolchain
  - 4.Installation of Yosys
  - 5.Installation of iverilog
  - 6.Installation of gtkwave

## 6. To add as subheading

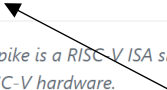
Add ## and space in front of the heading

```
# TASK-4:
## Spike and pk (proxy kernel) are components of the RISC-V ISA (Instruction Set Architecture) ecosystem.
> * **1.Spike:** _Spike is a RISC-V ISA simulator. It emulates a RISC-V processor and allows to run RISC-V programs on computer without needing actual RISC
> * **2.Proxy Kernel (pk):** _Proxy Kernel is a small software layer that provides an interface between the RISC-V hardware and a RISC-V operating system. :
```



### TASK-4:

**Spike and pk (proxy kernel) are components of the RISC-V ISA (Instruction Set Architecture) ecosystem.**

- 1.Spike: *Spike is a RISC-V ISA simulator. It emulates a RISC-V processor and allows to run RISC-V programs on computer without needing actual RISC-V hardware.*
  - 2.Proxy Kernel (pk): *Proxy Kernel is a small software layer that provides an interface between the RISC-V hardware and a RISC-V operating system. It is used when running an operating system on Spike. pk initializes the system, sets up memory, handles system calls, and provides other essential functionalities required by the operating system.*
- 

## 7. To add dot

Put >space\*space before statement ==>

> \* example

```
Preview Code Blame 228 lines (139 loc) · 8.27 KB Raw Copy Download Edit View Source
1  ✓ # TASK-4:
2  ✓ ## Spike and pk (proxy kernel) are components of the RISC-V ISA (Instruction Set Architecture) ecosystem.
3
4  > * **1.Spike:** _Spike is a RISC-V ISA simulator. It emulates a RISC-V processor and allows to run RISC-V programs on computer without needing actual RISC-V
5
6  > * **2.Proxy Kernel (pk):** _Proxy Kernel is a small software layer that provides an interface between the RISC-V hardware and a RISC-V operating system. I
7
8
```

Preview Code Blame 228 lines (139 loc) · 8.27 KB Raw Copy Download Edit View Source

### TASK-4:

**Spike and pk (proxy kernel) are components of the RISC-V ISA (Instruction Set Architecture) ecosystem.**

- 1.Spike: *Spike is a RISC-V ISA simulator. It emulates a RISC-V processor and allows to run RISC-V programs on computer without needing actual RISC-V hardware.*
- 2.Proxy Kernel (pk): *Proxy Kernel is a small software layer that provides an interface between the RISC-V hardware and a RISC-V operating system. It is used when running an operating system on Spike. pk initializes the system, sets up memory, handles system calls, and provides other essential functionalities required by the operating system.*

## 8. To add box to some statements

Add `` symbol at starting and end of statement.

``

Xyz

Xyz

Trm

``

**## Installation of spike:**

``

```
$ riscv64-unknown-elf-gcc -Ofast -mabi=lp64 -march=rv64i -o sum1ton.o sum1ton.c
```

```
$gcc sum1ton.c
```

```
$/a.out
```

```
$ riscv64-unknown-elf-gcc -Ofast -mabi=lp64 -march=rv64i -o sum1ton.o sum1ton.c
```

``

Installation of spike:

```
$ riscv64-unknown-elf-gcc -Ofast -mabi=lp64 -march=rv64i -o sum1ton.o sum1ton.c
$gcc sum1ton.c
$/a.out
$ riscv64-unknown-elf-gcc -Ofast -mabi=lp64 -march=rv64i -o sum1ton.o sum1ton.c
```



## 9. To add line after statement

Add ----- after the statement

*Xyzzzzzzzzzzzzzzzzzz*

-----

I have downloaded to VDI file which has spike,pk and riscv installaton and followed the Task-3.

```
> * Method 1 explains Manual Installation.  
> * Method 2 is Pre-installed VDI.  
> * Method 1 discusses installing Spike and pk, while Method 2 provides a VirtualBox disk image (VDI) with Spike and pk already installed.
```

```
# Compiling the .C using GCC Compiler:  
....
```

## METHOD-2:

I have downloaded to VDI file which has spike,pk and riscv installaton and followed the Task-3.

- Method 1 explains Manual Installation.
- Method 2 is Pre-installed VDI.
- Method 1 discusses installing Spike and pk, while Method 2 provides a VirtualBox disk image (VDI) with Spike and pk already installed.

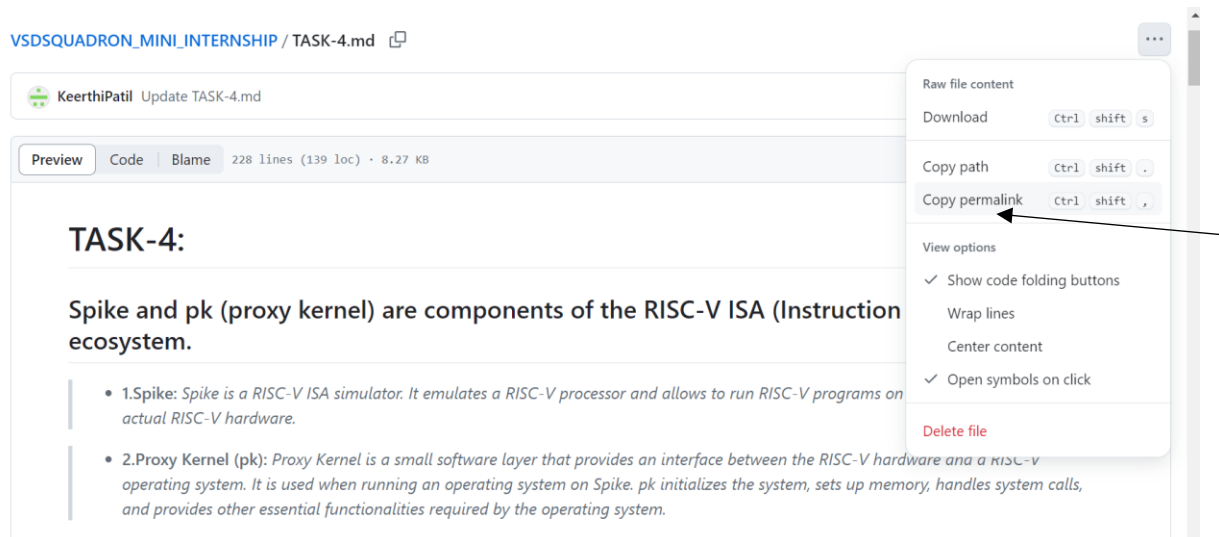


## 10. To add hyperlinks

Add this `<a href="url">link text</a>`

Add the url here

Add the name which u want to display



Select copy permalink add in url

```
12  
13 <a href="https://github.com/KeerthiPatil/VSDSQUADRON_MINI_INTERNSHIP/blob/013d3afb9cf6bb129c51351145294a6a20d56b6d/TASK-4.md">Task-4</a>
```

The above thing displays

[Task-4](https://github.com/KeerthiPatil/VSDSQUADRON_MINI_INTERNSHIP/blob/013d3afb9cf6bb129c51351145294a6a20d56b6d/TASK-4.md)

## 10.<hr> to give line

Tasks Assigned: To create a block representation to identify :

- Input Port
- Input Waveform
- Output Port
- Output Waveform

```
89      ## Meeting was conducted on 20th of February 2024 at 6PM IST
90      <hr>
91      <b>Tasks Assigned:</b>
92      To create a block representation to identify :
93
94      * Input Port
95      * Input Waveform
96      * Output Port
97      * Output Waveform
98      <hr>
```

## 11.To align at centre

Universal Asynchronous Receiver Transmitter Protocol based hardware transmitter

```
99      <b><p align="center">
100      Universal Asynchronous Receiver Transmitter Protocol based hardware transmitter
```

## 12.To align at left side image

Protocol Overview



The Idle state refers to that the transmission has not begun. It is represented through a

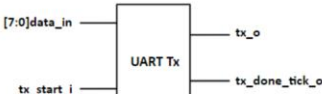
```
89      ## Meeting was conducted on 20th of February 2024 at 6PM IST

## Protocol Overview
<p align="left">

</p>
<b>The Idle state refers to that the transmission has not begun. It is represented through a high pulse
1.Wait until incoming signal becomes 0 (start bit), then start the sampling tick counter<br>
2.When tick counter reaches 7 (middle of start bit), clear tick counter and restart<br>
3.When counter reaches 15 (middle of first data bit), shift bit value into register & restart tick counter<br>
4. Repeat step 3 (N-1) more times to retrieve the remaining data bits<br>
</details>
<details>
```

## 13.To align at centre image

special bits to synchronize the sending and receiving inputs. These bits define the beginning and end of the data packet so the receiving UART knows when to start and stop reading the bits. These special bits are (START, DATA, PARITY, STOP) bits.



```
UART transmits asynchronously which means there is no need to transmit clock signal with the transmitter. Instead of clock, the transmitter transmits data with some special bits to synchronize the sending and receiving. These bits define the beginning and end of the data packet so the receiving UART knows when to start and stop reading the bits. These special bits are (START, DATA, PARITY, STOP) bits.
<p align="center">

</p>
```

## 14.To make bold

To generate waveform

vvp uart\_wav  
gtkwave dump.vcd



```
...  
...  
<b>To generate waveform</b>  
...  
vvp uart_wav  
gtkwave dump.vcd  
...
```

## 15.To make point

Meeting was conducted on 22nd of February 2024 at 6PM IST

Tasks Assigned:

- Simulation of code and testbench file
- Generate the waveform

```
### Meeting was conducted on 22nd of February 2024 at 6PM IST  
<br>  
<b>Tasks Assigned:</b>  
  
* Simulation of code and testbench file  
* Generate the waveform  
<br>  
<b>To generate the code and testbench file:</b>  
...
```

<https://github.com/kmkalpana2001/DIGITAL-VLSI-SOC-DESIGN-AND-PLANNING>

[https://github.com/Pradyumna1312/FPGA\\_Workshop\\_VSD](https://github.com/Pradyumna1312/FPGA_Workshop_VSD)

<https://github.com/deepsita/vsdBasicPD>

<https://github.com/britovski/vsdBasicPD>

[https://github.com/manjunathrv/VSD\\_Advanced\\_Physical\\_Design\\_with\\_sky130nmPDK](https://github.com/manjunathrv/VSD_Advanced_Physical_Design_with_sky130nmPDK)

<https://github.com/eugene-tarassov/vivado-risc-v/blob/master/.gitmodules>