

# REPORT

## Aim:

The goal of programming assignment was to write a device driver for PCI device.

## Design:

- Device driver is implemented as Loadable kernel module
- Like other modules, our module for device driver also has an initialisation and exit method.
- Initialisation method invokes PCI registration method which registers the PCI device, and it is done by `pci_register_driver(&cryptocard)`.
- In exit we have `pci_unregister_driver(&cryptocard)` to remove the device.
- All those initialisations and registration are done by probe function like enable device, allocation and mapping of device memory, registering irq etc.
- We are creating a character device named cryptocard that will show up as `/dev/cryptocard`.
- This is device file for our PCI device. This is the software interface through which we will interact with device.
- For this device file, various functions are defined like open, release, read, write, ioctl and mmap. When user program calls file related APIs from user space, then these functions are ultimately called and perform the requested operations as per user request.

## Implementation Details:

- PCI driver structure is cryptocard and I am globally maintaining some device and driver related information in `cryptocard_priv` struct like bar memory pointers etc.
- I maintained FD specific configuration setting in file `-> private_data`.
- I used ioctl function to implement `set_key`, `set_config`, `encrypt` and `decrypt` functions.
- Using using ioctl different functions are called for different configuration setting like `dma_encrypt`, `mmio_encrypt`, `dma_encrypt_interrupt` functions etc.
- To implement multiprocessing function, I have used `spin_lock` `cc_spin_lock` at time of encryption and decryption, so that only one operation will be allowed at any time.

## Testing:

- I have tested will all the possible cases with normal conditions
- Then I have tested with separate MMIO, DMA and with interrupts by changing test.c a little bit.
- As we have `set_config` function, I have checked with DMA with and without interrupt, MMIO with and without interrupt.
- By changing the arguments for the `set_config` I tested all these and for all the cases encrypted and decrypted messages are correct
- So I ensured that these are working fine
- And for multiprocess I have used used spinlocks so that I can achieve mutual exclusion

## Benchmark results:

	%system	%iowait	%user	%steal
MMIO	33	1,4	0,14	0,03
MMIO_interrupt	24,46	1,12	0,46	0,005
DMA	37,86	1,76	0,097	0,005
DMA_interrupt	6,45	1,93	0,867	0,007
MMAP	10,86	1,27	2,03	0,01
MMAP_interrupt	15,75	2,86	3,14	0,02