Lab – writing a character device driver

In this lab you will write a device driver to access a custom hardware (implemented on FPGA)

The device implement a cyclic counter from start value to end value

Device physical address: 0x101e9000

Device IRQ line: 11

Device registers

Offset address(hex)	Name	Description
0-99	REC	Data generated by external system (256 byte/packet)
100	DATA	Current counter value
104	MATCH	Generate interrupt on this value
108	START	Start value for the counter
10C	END	End value for the counter
110	ENABLE	Write 1 to enable counter
114	DISABLE	Write 1 to disable counter
118	STATUS	Interrupt status register

- Add a platform device object to the board specific code (arch/arm/mach-versatile/versatile_pb.c)
    - Add IO and IRQ resources
    - Register the device on board init code
    - Compile the kernel
- Write a kernel module to implement the FPGA driver with the following components
    - Platform driver object to match the device
    - fpga_init function to register the device
    - fpga_exit function to unregister the device
    - fpga_probe function to register the IO address, IRQ and build the char device interface
    - fpga_remove function to undo the probe process
    - Character device functions
        - fpga_read – read a packet (256 bytes)
        - fpga_mmap – map the device registers to the user
        - fpga_ioctl – custom operations:
            - set start value
            - set stop value
            - enable/disable the counter
            - read current value
            - set match value
            - get number of generated interrupts
            - block for interrupt
    - fpga_irq – interrupt handler to handle and acknowledge the interrupt
- Write a user space application to test all the functionality above
- Add the driver as a static module under drivers/mfd
    - Edit Kconfig and Makefile