Lecture 5 Device Driver

2025.12.10

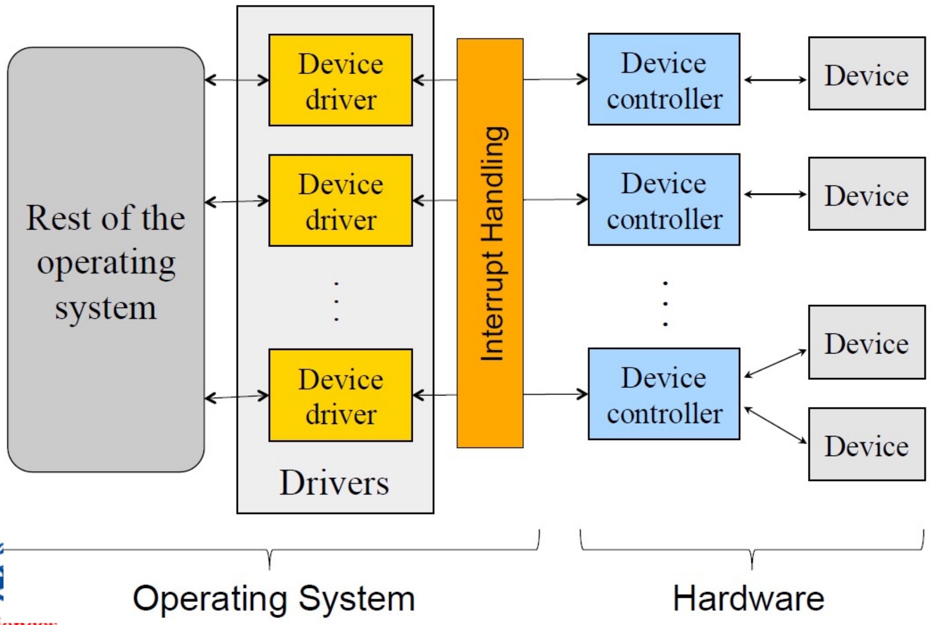
中国科学院大学
University of Chinese Academy of Sciences

Project 5 Device Driver

- Requirement
  - Implement driver for network card
    - Setup MAC controller registers to allow network card initialization, and sending/receiving data without interrupt
    - Implement network interrupt handler to send/receive data

Project 5 Device Driver

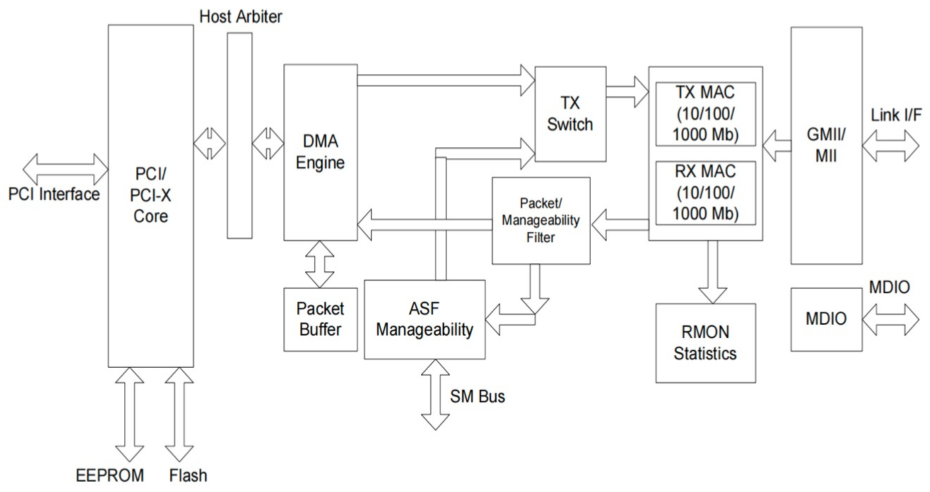
- Device driver
  - Interact with device controller
  - Issue commands to device controller to finish data reads and writes
    - e.g. network and disk



University of Chinese Academy of Sciences

Project 5 Device Driver

- MAC controller for E1000 NIC
  - Physical layer
  - Data link layer
    - MAC controller
  - Networking layer
  - Transportation layer
  - Application layer



Project 5 Device Driver

- MAC controller
  - MAC registers
    - register group, including a number of registers
    - *bios_read_fdt()* returns the base address of the register group (physical address)
  - ioremap
    - You need to map the above physical address to kernel virtual address
    - You need to flush TLB after finishing ioremap
  - Accessing a register
    - Base address + *i* * 4
    - *i* indicates i^th^ register (*i* starts from 0)

Project 5 Device Driver

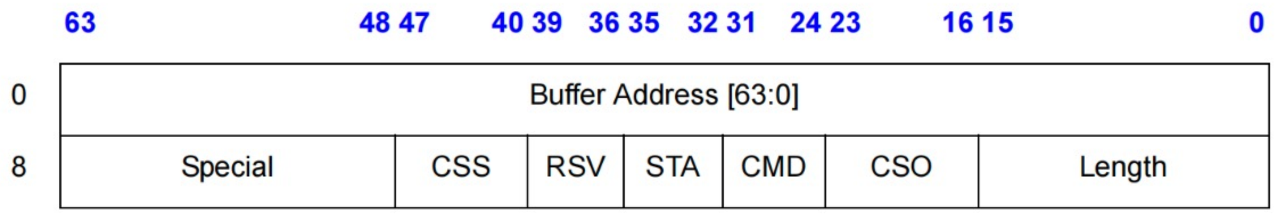
- Direct Memory Access
  - Exchange data between host and device bypassing CPU
  - DMA transmit descriptor
  - DMA receive descriptor
  - Use physical addresses

Project 5 Device Driver

- DMA descriptor
  - – Each descriptor is 16 bytes
  - – Transmit descriptor for transmitting data
  - – Receive descriptor for receiving data
  - – One descriptor for one network frame

Project 5 Device Driver

- DMA descriptor - Tx descriptor
  - Buffer address: physical address of the buffer for transmitting data
  - Length: data length



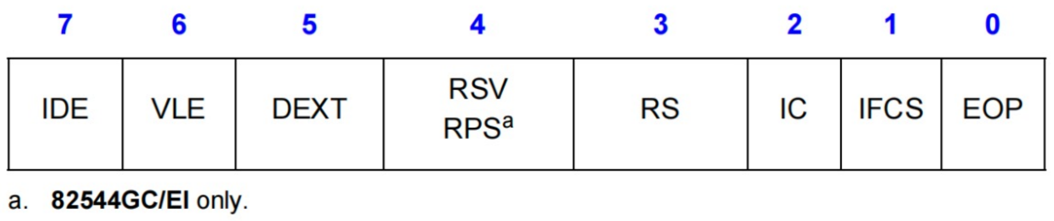
	63 48	47 40	39 36	35 32	31 24	23 16	15 0
0	Buffer Address [63:0]						
8	Special	CSS	RSV	STA	CMD	CSO	Length

Tx Descriptor

图 P5-3: E1000 发送描述符 (Legacy)

Project 5 Device Driver

- DMA descriptor - Tx descriptor
  - CMD
    - DEXT: 0, indicates Legacy format
    - RS: 1, DMA controller records transmission status in STA field
    - EOP: indicates whether the current descriptor is the last one of the current packet



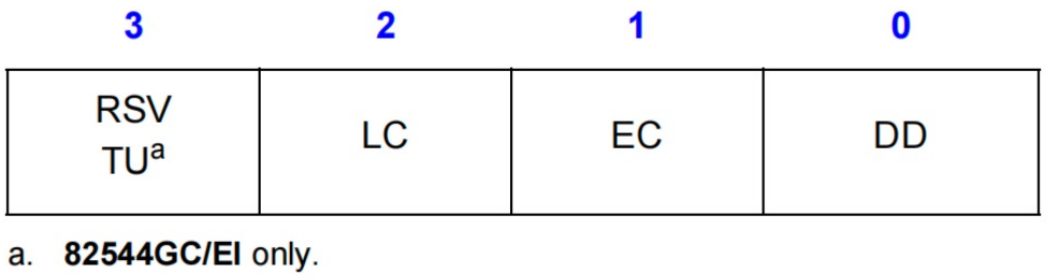
7	6	5	4	3	2	1	0
IDE	VLE	DEXT	RSV RPS[a]	RS	IC	IFCS	EOP

a. **82544GC/EI** only.

图 P5-4: E1000 发送描述符 CMD 字段

Project 5 Device Driver

- DMA descriptor - Tx descriptor
  - STA
    - DD: indicates the transmission of current frame is finished



3	2	1	0
RSV TU[a]	LC	EC	DD

a. **82544GC/EI** only.

图 P5-5: E1000 发送描述符 STA 字段

Project 5 Device Driver

- Circular buffer for Tx descriptors
  - TDBAL, TDBAH registers: the low and high 32-bits of the address of the circular buffer
  - TDLEN register: the bytes of the circular buffer
  - Head: the position where DMA controller is currently processing
  - Tail: the position where OS is currently processing

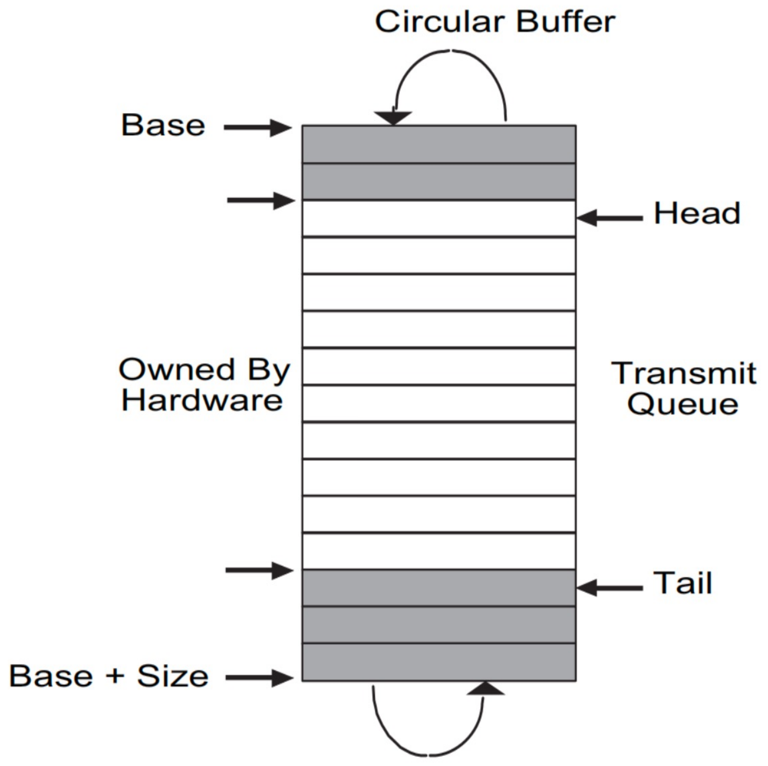


图 P5-6: E1000 发送描述符循环数组

Project 5 Device Driver

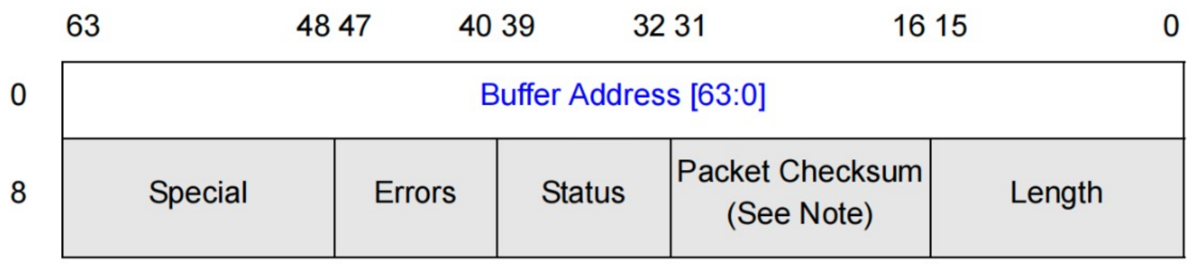
- Circular buffer for Tx descriptors
  - TDH register: stores the index of the descriptor element pointed by Head in the circular buffer
  - TDT register: stores the index of the descriptor element pointed by Tail in the circular buffer

Project 5 Device Driver

- Enabling data transmission
  - TCTL register: 1, enable data transmission
  - Once TDH is not equals TDT, DMA controller starts to transmit data

Project 5 Device Driver

- DMA descriptor - Rx descriptor
  - Buffer address: physical address of the buffer for receiving data
  - Length: data length



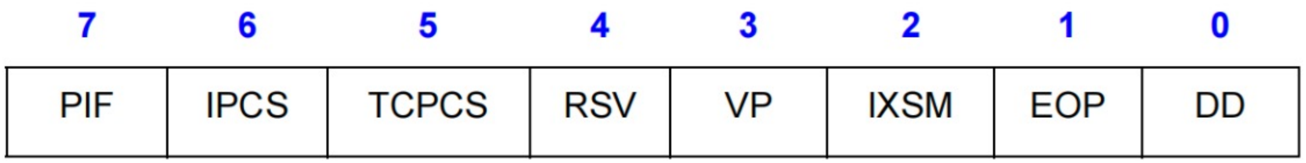
	63 48	47 40	39 32	31 16	15 0
0	Buffer Address [63:0]				
8	Special	Errors	Status	Packet Checksum (See Note)	Length

Rx Descriptor

图 P5-13: E1000 接收描述符

Project 5 Device Driver

- DMA descriptor - Rx descriptor
  – STA (Status)
    - EOP: indicates whether the current descriptor is the last one
    - DD: indicates current frame is finished



7	6	5	4	3	2	1	0
PIF	IPCS	TCPCS	RSV	VP	IXSM	EOP	DD

图 P5-14: E1000 接收描述符 STA 字段

Project 5 Device Driver

- Circular buffer for Rx descriptors
  - RDBAL, RDBAH registers: the low and high 32-bits of the circular buffer
  - RDLEN register: the bytes of the circular buffer
  - Head: the position where DMA controller is currently processing
  - Tail: the position where OS is currently processing

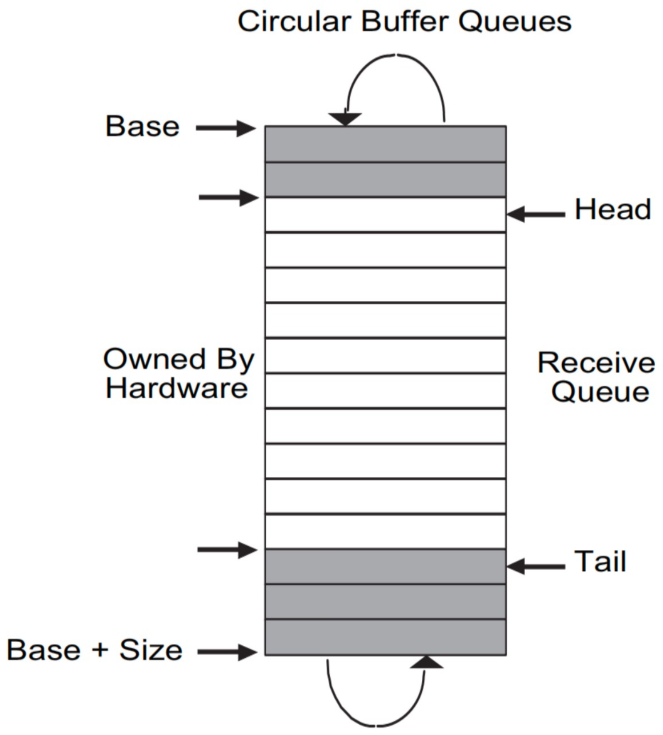


图 P5-15: E1000 接收描述符循环队列

Project 5 Device Driver

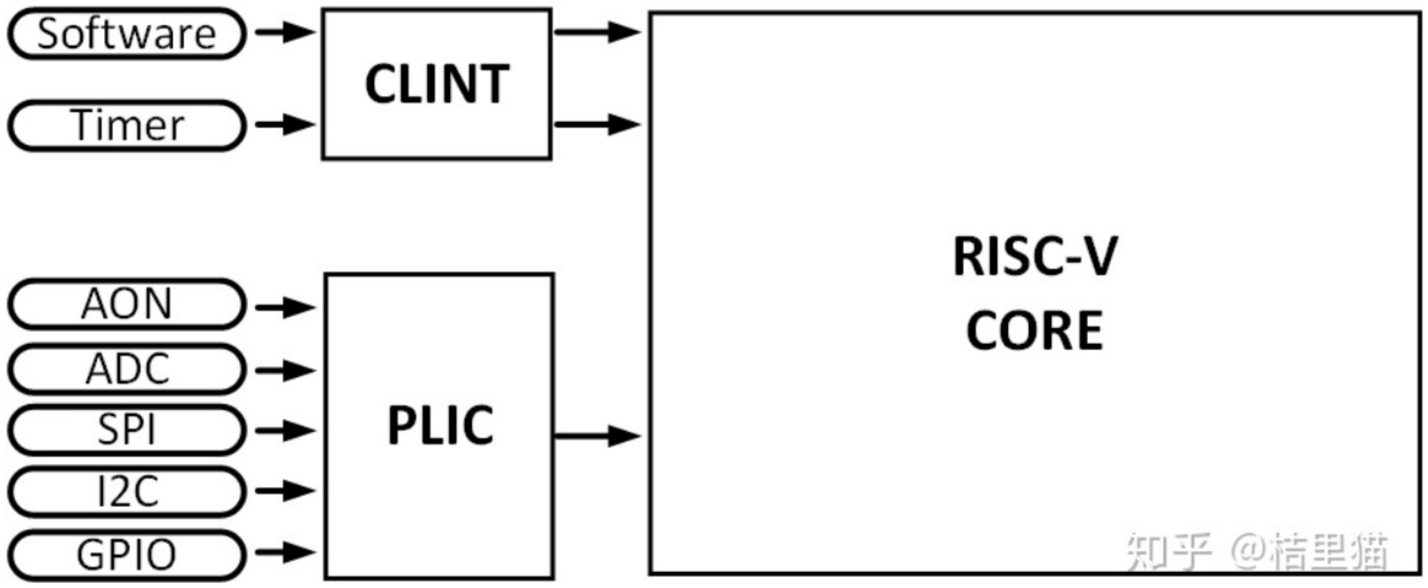
- Circular buffer for Rx descriptors
  - RDH register: stores the index of the descriptor element pointed by Head in the circular buffer
  - RDT register: stores the index of the descriptor element pointed by Tail in the circular buffer

Project 5 Device Driver

- Enabling data receiving
  - RCTL register: 1, enable data receiving
  - NIC only receives the packets with the matched MAC address
  - You need to fill MAC address into RAL0 and RAH0 registers before receiving data

Project 5 Device Driver

- MAC interrupt handler
  - PLIC(Platform Level Interrupt Controler)

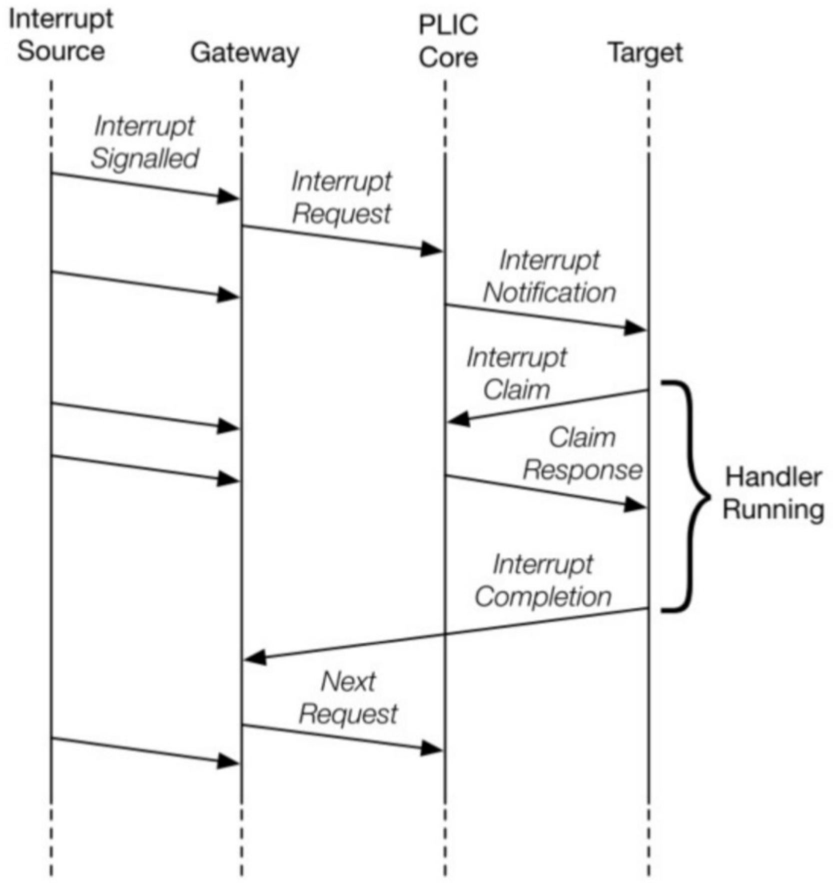


Project 5 Device Driver

- MAC interrupt handler
  - Use scause register to judge IRQ_S_EXT interrupt
  - Use registers provided by PLIC to judge and process NIC interrupt
    - claim/complete register
      - read device ID from claim register and indicate accepting the interrupt from PLIC
        - » NIC device ID: 33 (qemu), 3(board)
      - write device ID to complete register to indicate finishing interrupt handling

Project 5 Device Driver

- MAC interrupt handler



Project 5 Device Driver

- MAC interrupt handler
  - Use registers provided by PLIC to judge and process NIC interrupt
    - IMS register (Interrupt Mask Set): write 1 to it to enable NIC interrupt
    - IMC register (Interrupt Mask Clear): write 1 to it to clear the corresponding bit in IMS register, then NIC interrupt can be disabled

Project 5 Device Driver

- MAC interrupt handler
  - Use registers provided by PLIC to judge and process NIC interrupt
    - ICR register (Interrupt Cause Read): when NIC interrupt occurs, the corresponding bit in ICR is set
    - ICS register (Interrupt Cause Set): write 1 to it then the corresponding bit in ICR is also set
      - We do not use this register in P5

Project 5 Device Driver

- MAC interrupt handler
  - Use registers provided by PLIC to judge and process NIC interrupt
    - TXQE interrupt for data transmission
      - When TDH equals to TDT, this interrupt is triggered
    - RXDMT0 interrupt for receiving data
      - Set RCTL.RXDMT to 00
      - When the Rx descriptors owned by NIC is less than half, this interrupt is triggered

Project 5 Device Driver

- Step-by-step: Task1
  - Initialize DMA descriptors for transmitting data
  - Set corresponding DMA registers to allow transmitting data
  - Implement the syscall for sending data through NIC
  - The number of packets you send needs to be larger than the number of Tx descriptors in the circular buffer

Project 5 Device Driver

- Step-by-step: Task2
  - Initialize DMA descriptors for receiving data
  - Set corresponding DMA registers to allow receiving data
  - Implement the syscall for receiving data through NIC
  - The number of packets you receives needs to be larger than the number of Rx descriptors in the circular buffer

Project 5 Device Driver

- Step-by-step: Task 3
  - Enable MAC interrupt
  - Implement MAC interrupt handler for receiving data and transmitting data
  - The receive thread blocks itself when there is less than half of total Rx descriptors.
  - The transmit thread blocks itself after all descriptors are used. Once NIC finishes transmission, it is waked up

Project 5 Device Driver

- Step-by-step: Task 4
  - Design and implement a new syscall *sys_net_recv_stream* to receive *n* bytes in sequence
  - You need to handle packet loss or out-of-order packet arriving
  - Please refer to Sec. 10 in the guidebook to implement a reliable transfer protocol, supporting DAT, ACK and RSD packet

Project 5 Device Driver

- Requirement for S/A/C-Core

Core type	Task requirements
S-Core	Task 1, 2
A-Core	Tasks 1~3
C-Core	Tasks 1~4

Project 5 Device Driver

- Task testing
  - For transmitting data
    - Use wireshark (Windows) or tcpdump (Linux) to examine if the transmit thread successfully transmit data
    - Please read our reference documents
  - For receiving data
    - Use pktRxTx to transmit data from host to your development card to test receiving data

Project 5 Device Driver

- Requirements for design review
  - 请分别展示发送和接收DMA描述符的示例，并介绍所填充字段的含义
  - 请介绍网卡中断的处理流程，包含使用哪些寄存器、如何使能、判断和处理中断
  - 任何想讨论的问题

Project 5 Device Driver

- P5 schedule
  - Design review
    - 17th Dec.
  - C core due
    - 24th Dec.
  - A core due
    - 31st Dec.

Project 5 Device Driver

- P6 schedule
  - Assignment
    - 24th Dec.
  - Design review
    - 31st Dec.
  - C core due
    - 7th Jan.
  - Final due
    - 14th Jan.