Exercise 1

843ACBDD5h

This is an address of a 2-levels paged system. What is the memory size? If the page size is 16 KB how many slots do level 1 and 2 tables consist of? What are the hexadecimal values of level 1 and level 2 tables and of the in-page byte of this address. What is the decimal value of level 1 table? In the processor word parallelism is 8 bytes how many bits for each table entry can be used for status and other process related information

The address is 36 bit. The memory size is therefore 64GB. 16 KB => 14 bit. There are 4 millions 16K pages in 64GB. 36-14=22 and therefore level 1 and level 2 are 11 bit each that is 2048 slots each.

843ACBDD5h => 1000 0100 0011 1010 1100 1011 1110 1110 0101 => 11 1110 1110 0101 => 421h 6B2h 3DD5h

Level 1 table decimal value is 421h= 4x256d + 2x16d + 1= 1057d

Processor word size is 64 bits. Tables are paged and aligned and therefore the 14 LSBs bits (page size 16KB - 2^{14}) of the their initial addresses stored in the tables are implicitly zero. Therefore there are 64-(36-14) = 42 bits available for status and other process related information for level 1 and 2 table entries

Exercise 2

How does the interrupt mechanism in a protected x86 environment work? Under what conditions an interrupt is acknowledged?

An interrupt INT is acknowledged if the IEN flag is set and the protection level is correct. Upon activation of INT signal an INTA* signal is twice activated by the processor; during the second activation a byte externally provided is read (interrupt identificator) by the processor which is internally multiplied by 8 (descriptor size in bytes). This constitutes the index in a table of interrupt descriptors which is pointed by a processor internal register (IDTR). The selected interrupt descriptor contains the virtual address of the interrupt response program.

Exercise 3

In a DLX processor what types of hazards are present in this instruction flow and is it possible to rearrange it in order to remove them without hardware modification?

```
LW      R1, 64(R29)
LW      R16, 32(R16)
MUL     R24, R2,R2
ADD     R16,R16,R24
SW      4(R28), R17
BR      +100, R6
MUL     R8, R16, R24
```

There is a data hazard (ADD R16,R16,R24) with MUL R24, R2,R2 and a control hazard (BR +100, R6).

Rearrangement

```
        LW      R16, 32(R16)
MUL     R24, R2,R2
BR      +100, R6        ; Delayed branch
LW      R1, 64(R29)     ! Delayed instruction
SW      4(28), R17
ADD     R16,R16,R24   ;
```

; The three last (moved) instructions are always executed and the ADD is possible because the RF (R16 and R24) have been already updated

```
MUL     R8, R16, R24    ; executed if not branch
```