Daniel Peterson 109091561

4.Explain the tradeoffs between precise and imprecise interrupts on a superscalar machine.

**Precise interrupts leave the machine in a well-defined state. An instruction is either fully executed or not executed at all and each instruction takes place one at a time. Imprecise interrupts need to store a lot of information on their stack because the state of the machine is not known. Imprecise interrupts can be faster if backwards compatibility with precise interrupts is not required. They have a much larger chip area and a more complex design. Precise interrupts are simpler, smaller, and faster in general for practical purposes.**

6.Suppose that a system uses DMA for data transfer from disk controller to main memory. Further assume that it takes t\_1 nsec on average to acquire the bus and t\_2 nsec to transfer one word over the bus (t\_1 >> t\_2). After the CPU has programmed the DMA controller, how long will it take to transfer 1000 words from the disk controller to main memory, if:

(a) word-at-a-time mode is used

(b) burst mode is used

Assume that commanding the disk controller requires acquiring the bus to send one word and acknowledging a transfer also requires acquiring the bus to send one word.

**a) Each word takes t\_1 time to acquire and t\_2 time to send. There must also be an acknowledgement that the transfer is taking place. This results in a time of 2 \* (t\_1 + t\_2) for each word. If there are 1000 words, then the total time is 2 \* 1000 (t\_1 + t\_2) or 2000 \* (t\_1 + t\_2) = 2000\*t\_1 + 2000\*t\_2**

**b) In burst mode, only one acquire is necessary and then all words are sent one after another. So t\_1 is only performed once, while t\_2 is performed twice for each transfer. This is t\_1 + 2\*t\_2 for each word. For 1000 words, the total time is t\_1 + 2\*1000\*t\_2 or t\_1 + 2000\*t\_2**

8.Suppose that a computer can read or write a memory word in 5 nsec. Also suppose that when an interrupt occurs, all 32 CPU registers, plus the program counter and PSW are pushed onto the stack. What is the maximum number of interrupts per second this machine can process?

**32 CPU + PC + PSW = 34 writes**

**34 \* 5\*(10^-9) = 170nsec**

**1/(170 nsec) = 5,882,352.941 = 5.88 million interrupts per second**

12.A typical printed page of text contains 50 lines of 80 characters each. Imagine that a certain printer can print 6 pages per minute and that the time to write a character to the printer’s output register is so short it can be ignored. Does it make sense to run this printer using interrupt-driven I/O if each character printed requires an interrupt that takes 50 usec all-in to service?

**50 lines \* 80 chars \* 6 pages = 24,000 chars/min**

**24,000/60 = 400 chars/sec**

**50usec \* 400 = 20ms total from interrupt each second**

**1000ms – 20ms = 980ms of non-interrupt time each second**

**Very little of the CPU time is being used by the interrupt so it would make sense to use interrupt driven I/O.**

14.In which of the four I/O software layers is each of the following done.

(a) Computer the track, sector, and head for a disk read

(b)Writing commands to the device registers

(c) Checking to see if the user is permitted to use the device

(d) Converting binary integers to ASCII for printing

**a) Device driver layer**

**b) Device driver layer**

**c) Operating system layer**

**d) User level software layer**

20**.** RAID level 3 is able to correct single-bit errors using only one parity drive. What is the point of RAID level 2? After all, it also can only correct one error and takes more drives to do so.

**RAID level 2 has the ability to recover from undetected errors and from crashed drives. If a drive has a bad bit, the RAID level 2 will correct it using its Hamming code, unlike RAID level 3.**

21**.** A RAID can fail if two or more of its drives crash within a short time interval;. Suppose that the probability of one drive crashing in a given hour is p. What is the probability of a k-drive RAID failing in a given hour?

**q = 1- p**

**prob that drive fails = 1 – q^k – (k \* p \* q^(k-1)) is the probability that a k-drive RAID will fail**

28.Consider a magnetic disk consisting of 16 heads and 400 cylinders. This disk has four 100-cylinder zones with the cylinders in different zones containing 160, 200, 240 and 280 sectors, respectively. Assume that each sector contains 512 bytes, average seek time between adjacent cylinders is 1 msec, and the disk rotates at 7200 RPM. Calculate the

(a) disk capacity,

(b) optimal track skew, and

(c) maximum data transfer rate.

**a) capacity = platters \* cylinders \* sectors/track \* bytes/sector = 16 \* 100 \* zone\_size \* 512**

**zone\_size = 160: 131,072,000**

**zone\_size = 200: 163,840,000**

**zone\_size = 240: 196,608,000**

**zone\_size = 280: 229,376,999**

**b) (1msec/8.33msec) \* 220 avg num sectors + 4ms = 30.41**

**c) max data transfer rate = 7200/60 \* 280 \* 512 = 17,203,200 bytes per second. There are the most sectors in the zone with 280.**

35. In the discussion on stable storage, it was shown that the disk can be recovered to a consistent state (a write either completes or does not take place at all) if a CPU crash occurs during a write. Does this property hold if the CPU crashes again during a recovery procedure? Explain your answer.

**Yes, this property holds even if the CPU crashes during the recovery because a write will only occur completely, it will not partially write incorrect data. A write will only take place when both blocks can be checked to ensure that they are both good.**

36**.** In the discussion on stable storage, a key assumption is that a CPU crash that corrupts a sector leads to an incorrect ECC. What problems might arise in the five crash-recovery scenarios show in Figure 5-27 if this assumption does not hold?

**In the first and fifth scenario, nothing changes because the two blocks are the same and the ECC is not checked. In the second and fourth scenarios, if the ECC error is not shown, then the value of block 1 will not be overwritten by block 2 because it appears to be valid, which leads to incorrect data. In the third scenario, the first block would overwrite block 2 regardless of any errors.**

39**.** A system simulates multiple clocks by chaining all pending clock requests together as shown in Fig. 5-30. Suppose the current time is 5000 and there are pending clock requests for time 5008, 5012, 5015, 5029, and 5037. Show the values of Clock header, Current time, and Next signal at times 5000, 5005, and 5013. Suppose a new (pending) signal arrives at time 5017 for 5033. Show the values of Clock header, Current time and Next signal at time 5023.

**5000: Clock header = [8][] -> [4][] -> [3][] -> [14][] -> [8][x], Current time = 5000, Next Signal = 8**

**5005: Clock header = [8][] -> [4][] -> [3][] -> [14][] -> [8][x], Current time = 5005, Next Signal = 3**

**5013: Clock header = [3][] -> [14][] -> [8][x] , Current time = 5013, Next Signal = 2**

**5023: Clock header = [14][] -> [4][] -> [4][x] , Current time = 5023, Next Signal = 6**

40**.** Many versions of UNIX use an unsigned 32-bit integer to keep track of the time as the number of seconds since the origin of time. When will these systems wrap around (year and month)? Do you expect this to actually happen?

**The signed 32-bit integer systems will wrap around in January of 2038 because the MSB of the integer will switch from 0 to 1. If the 32-bit integer is unsigned, then the year that it will run over becomes January of 2106. This will only happen in systems which used an unsigned 32-bit integer. Systems which use 64-bit integers will not run into this problem until thousands of years into the future.**

44. The designers of a computer system expected that the mouse could be moved a maximum rate of 20 cm/sec. If a mickey is 0.1 mm and each mouse message is 3 bytes, what is the maximum data rate of the mouse assuming that each mickey is reported separately?

**20cm/sec \* 10 = 200mm/sec**

**(200mm/sec / 0.1mm) \* 3 bytes = 6000 bytes/sec**

47.Assume that it takes 2 nsec to copy a byte, how much time does it take to completely rewrite the screen of an 80 character x 25 line text mode memory-mapped screen? What about a 1024x768 pixel graphics screen with 24-bit color?

**80 \* 25 \* 2 = 4000 bytes**

**4000bytes \* 2nsec = 8000nsec for text mode memory-mapped screen**

**24 bit colors = 3 bytes per pixel (R G B)**

**1024 \* 768 \* 3 = 2,359,296 bytes**

**2,359,296 \* 2nsec = 4,718,592nsec for graphics screen**

50**.** A thin-client terminal is used to display a Web page containing an animated cartoon of size 400 pixels x 160 pixels running at 10 frames/sec. What fraction of a 100-Mbps Fast Ethernet is consumed by displaying the cartoon?

**400 \* 160 = 64,000 pixels**

**64,000 \* 10 = 640,000 pixels per second**

**640,000 \* 8bits = 5,120,000 for B/W**

**640,000 \* 24bits = 122,880,000 for RGB**

**10^8 bits/sec cannot handle the RGB video so it must be B/W**

**5,120,000/100,000,000 = 0.0512 or 5.12% of the Ethernet is used**

53**.** If a CPU’s maximum voltage, *V*, is cut to *V*/*n*, its power consumption drops to 1/*n* of its original value and its clock speed drops to 1/*n* of its original value. Suppose that a user is typing at 1 char/sec, but the CPU time required to process each character is 100 msec. What is the optimal value of *n* and what is the corresponding energy saving in

percent compared to not cutting the voltage? Assume that an idle CPU consumes no energy at all.

**1 char/sec = 1 char / 1000msec input**

**1 char / 100msec process time**

**1000 – 100 = 900 idle time**

**Power can be cut by n = 10 (1/10) to drop clock speed to 100 msec**

**This way, the processing speed matching the input speed and the power is now 1/10th of its original. 10% of power used when cutting the voltage**