CS61C Summer 2014 Discussion 15 – Dependability and I/O (Solutions)

Exercise 1 (Hamming Codes): Recall the basic structure of a Hamming code. Given bits $1, \ldots, m$, the bit at position 2^n (starting at $n = 0$, the first bit) is parity for all the bits with a 1 in position n. For example, the first bit is chosen such that the sum of all odd-numbered bits is even.

1. Suppose you had the bits 0011 and want to add some bits to allow the correction of single-bit errors.
    a. How many bits do we have to add? **3**
    b. Which bits are parity bits? **XX0X011**
    c. Which bits does each parity bit cover? **Bit 0: 0, 2, 4, 6; Bit 1: 1, 2, 5, 6; Bit 3: 3, 4, 5, 6**
    d. Write the completed coded representation for 0011. **1000011**
    e. What do we need to make this into a SEC-DED code? **Add another parity bit.**
2. Find the original bits given the following SEC Hamming codes.
    a. 0110111
    Group 0 & 3 have an error; bit 4 should be 0: 1011
    b. 1001000
    Group 0 & 3 have an error; bit 4 should be a 1: 0100
3. If we only wanted SED but not SEC, how many bits would you need to add to 4 bits? 16 bits?
    1, 1

The idea of RAID is to use an array of smaller disks to simulate a larger disk (and in some cases provide better performance, reliability, and redundancy).

RAID 0	Data striping
RAID 1	Disk mirroring
RAID 2	Bit-striping with ECC disks
RAID 3	Byte-striping with dedicated parity disk
RAID 4	Block-striping with dedicated parity disk
RAID 5	Block-striping with interleaved parity
RAID 6	Block-striping with two interleaved parity disks for DEC

Which is faster for reading data (in a similar setup): **RAID 0** or RAID 5? What about **RAID 0** or RAID 1?

Exercise 3 (Memory Mapped I/O): Certain memory addresses correspond to registers in I/O devices rather than physical memory. The control register indicates if the I/O device is ready to transmit/receive data in the data register.

Register	Location	Contains
Receiver Control	0xFFFF0000	Lowest bit is ready bit
Receiver Data	0xFFFF0004	Lowest byte is received data
Transmitter Control	0xFFFF0008	Lowest bit is ready bit
Transmitter Data	0xFFFF000C	Lowest byte is data to transmit

Write MIPS code to read a byte from the receiver as soon as it becomes ready and send it to the transmitter.

```
        lui $t0, 0xFFFF
receive_wait:
        lw $t1, 0(4t0)
        andi $t1, $t1, 1
        beq $t1, $0, receive_wait
        lb $t2, 4($t0)
transmit_wait:
        lw $t1, 8($t0)
        andi $t1, $t1, 1
        beq $t1, $0, transmit_wait
        sb $t2, 12($t0)
```

Exercise 4 (Polling and Interrupts): Fill in the table below:

Operation	Definition	Pros/Cons	Ideal Use
Polling	Pretty much the above; forces hardware to wait on ready bits (alternatively, if timing of device is known – the ready bit can be polled at the frequency of the device). It basically means manually checking the ready bit	Pros: -easy to write -poll handler does not have excessively high overhead -deterministic -doesn't require additional hardware Cons: -unfeasable on hardware with fast transfer rates that is actually rarely ready	Slow devices: Keyboards, mice
Interrupt-Driven I/O	Hardware fires an "exception" when it becomes ready. CPU changes $PC to execute code in the interrupt handler when this occurs	Pros: -Necessary for fast devices that are rarely Con: -nondeterministic when interrupt occurs -interrupt handler has some overhead (saves all registers), meaning polling can actually be faster for slow, often ready devices such as mice ready.	Fast devices: Network devices, hard drives