

## Discussion 14: I/O, ECC/Parity, RAID

### Hamming ECC

Recall the basic structure of a Hamming code. Given bits  $1, \dots, m$ , the bit at position  $2^n$  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.

Bit	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Data	<u>P1</u>	<u>P2</u>	D1	<u>P4</u>	D2	D3	D4	<u>P8</u>	D5	D6	D7	D8	D9	D10	D11
P1	X		X		X		X		X		X		X		X
P2		X	X			X	X			X	X			X	X
P4				X	X	X	X					X	X	X	X
P8								X	X	X	X	X	X	X	X

- How many bits do we need to add to  $0011_2$  to allow single error correction?
- Which locations in  $0011_2$  would parity bits be included?
- Which bits does each parity bit cover in  $0011_2$ ?
- Write the completed coded representation for  $0011_2$  to enable single error correction.
- How can we enable an additional double error detection on top of this?
- Find the original bits given the following SEC Hamming Code:  $0110111_2$
- Find the original bits given the following SEC Hamming Code:  $1001000_2$
- Find the original bits given the following SEC Hamming Code:  $010011010000110_2$

## RAID

Fill out the following table:

	Configuration	Pro / Good for...	Con / Bad for...
RAID 0			
RAID 1			
RAID 4			
RAID 5			

**Note:** RAID 2 and 3 are conceptually the same as RAID 4, but with bit-striping and byte-striping instead.

## I/O

1. Fill this table of polling and interrupts.

Operation	Definition	Pro / Good for...	Con / Bad for...
Polling			
Interrupts			

2. Memory Mapped I/O

Certain memory addresses correspond to registers in I/O devices and not normal memory.

**0xFFFF0000 – Receiver Control:**

Lowest two bits are interrupt enable bit and ready bit.

**0xFFFF0004 – Receiver Data:**

Received data stored at lowest byte.

**0xFFFF0008 – Transmitter Control**

Lowest two bits are interrupt enable bit and ready bit.

**0xFFFF000C – Transmitter Data**

Transmitted data stored at lowest byte.

Write MIPS code to read a byte from the receiver and immediately send it to the transmitter.