Discussion ~~13~~ 14: ECC/Parity, RAID, I/O

Hamming ECC

even parity = even # of 1's

2^n

Recall the basic structure of a Hamming code. Given bits 1, . . . , 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	P1	P2	D1	P4	D2	D3	D4	P8	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

i. How many bits do we need to add to 0011_2 to allow single error correction?

3 bits

ii. Which locations in 0011_2 would parity bits be included?

P1 P2 D1 P4 D2 D3 D4 ⇒ PPDPDDD

iii. Which bits does each parity bit cover in 0011_2?

P1: 1 3 5 7 ...

P2: 2 3, 6 7 ...

P4: 4 5 6 7, 12 13 14 15 ...

iv. Write the completed coded representation for 0011_2 to enable single error correction.

1 0 0 0 0 1 1

1 2 2 4 4 4 4

0000 ↔ 0001 ham dist = 4

v. How can we enable an additional double error detection on top of this?

add 1 more bit to cover entire sequence

7 + P

vi. Find the original bits given the following SEC Hamming Code: 0110111_2

P1 + P4 groups are wrong

⇒ error is at 1+4=5

0110011

0 1 1 1 X P1

1 1 1 1 ✓ P2

0 1 1 1 X P4

vii. Find the original bits given the following SEC Hamming Code: 1001000_2

1001000

1 0 0 0 X P4

0 0 0 0 ✓ P2

1 0 0 0 X P1

1

P1 + P4 groups are wrong

error = 1+4=5

1001100 → 1 0100

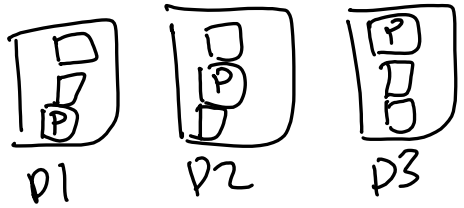
1011

viii. Find the original bits given the following SEC Hamming Code: 0100110100001102

RAID

Fill out the following table:

	Configuration	Pro / Good for…	Con / Bad for…
RAID 0	Data disks	No overhead, fast read/write	No Redundancy
RAID 1	Mirroring Data disk + copy	Redundancy, avail. -fast recovery	Cost
RAID 2	with bitwise parity + parity disk	less overhead - 1 parity disk	writing slower - parity is bottleneck
RAID 3	byte wise parity + parity disk		
RAID 4	blockwise parity + parity disk		
RAID 5	distributed parity - no parity disk	higher writing throughput	



I/O

1. Fill this table of polling and interrupts.

Operation	Definition	Pro / Good for…	Con / Bad for…

Polling	hardware waits for data	frequent data lower overhead when handling data	always waiting
Interrupts	i/o device triggers an interrupt ("exception")	infrequent data other processes while "waiting"	higher overhead

2. Memory Mapped I/O
   Certain memory addresses correspond to registers in I/O devices and not normal memory.
   0xFFFF0000 – Receiver Control:
   LSB is the ready bit, there may be other bits set that we don't need right now.
   0xFFFF0004 – Receiver Data:
   Received data stored at lowest byte.
   0xFFFF0008 – Transmitter Control
   LSB is the ready bit, there may be other bits set that we don't need right now.
   0xFFFF000C – Transmitter Data
   Transmitted data stored at lowest byte.

   Write RISC-V code to read a byte from the receiver and immediately send it to the transmitter.