CS 61C: Great Ideas in Computer Architecture (Machine Structures)
Dependability and RAID

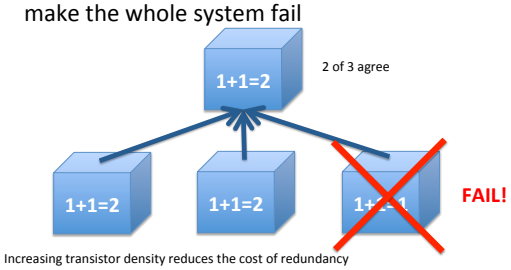
Instructors:
Krste Asanovic, Randy H. Katz
http://inst.eecs.Berkeley.edu/~cs61c/fa12

Review - 6 Great Ideas in Computer Architecture

1. Layers of Representation/Interpretation
2. Moore's Law
3. Principle of Locality/Memory Hierarchy
4. Parallelism
5. Performance Measurement & Improvement
6. **Dependability via Redundancy**

Review - Great Idea #6: Dependability via Redundancy

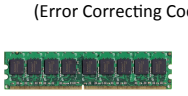
- Redundancy so that a failing piece doesn't make the whole system fail

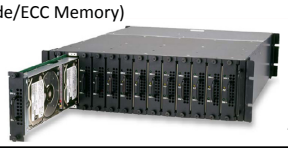


Increasing transistor density reduces the cost of redundancy

Review - Great Idea #6: Dependability via Redundancy

- Applies to everything from datacenters to memory
  - Redundant datacenters so that can lose 1 datacenter but Internet service stays online
  - Redundant routes so can lose nodes but Internet doesn't fail
  - Redundant disks so that can lose 1 disk but not lose data (Redundant Arrays of Independent Disks/RAID)
  - Redundant memory bits of so that can lose 1 bit but no data (Error Correcting Code/ECC Memory)





Agenda

- Dependability, Reliability, Availability metrics
- Codes for Redundancy
- Administrivia
- Error Detection and Correction in Memory
- RAID

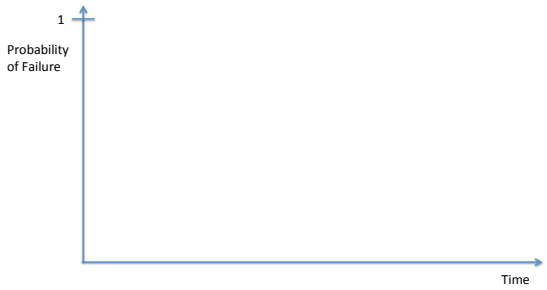
Agenda

- Dependability, Reliability, Availability metrics
- Codes for Redundancy
- Administrivia
- Error Detection and Correction in Memory
- RAID

Dependability Measures

- Reliability: Mean Time To Failure (MTTF)
- Service interruption: Mean Time To Repair (MTTR)
- Mean time between failures (MTBF)
  - MTBF = MTTF + MTTR
- Availability = MTTF / (MTTF + MTTR)
- Improving Availability
  - Increase MTTF: More reliable hardware/software + Fault Tolerance
  - Reduce MTTR: improved tools and processes for diagnosis and repair

Understanding MTTF



Availability Measures

- Availability = MTTF / (MTTF + MTTR) as %
  - MTTF, MTBF usually measured in hours
- Since hope rarely down, shorthand is "number of 9s of availability per year"
- 1 nine: 90% => 36 days of repair/year
- 2 nines: 99% => 3.6 days of repair/year
- 3 nines: 99.9% => 526 minutes of repair/year
- 4 nines: 99.99% => 53 minutes of repair/year
- 5 nines: 99.999% => 5 minutes of repair/year

Dependability Design Principle

- Design Principle: No single points of failure
  - "Chain is only as strong as its weakest link"
- Dependability Corollary of Amdahl's Law
  - Doesn't matter how dependable you make one portion of system
  - Dependability limited by part you do not improve

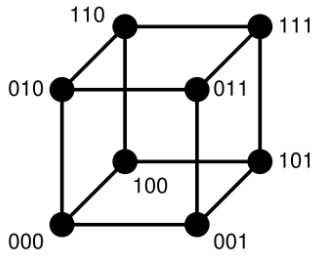
Agenda

- Dependability, Reliability, Availability metrics
- Codes for Redundancy
- Administrivia
- Error Detection and Correction in Memory
- RAID

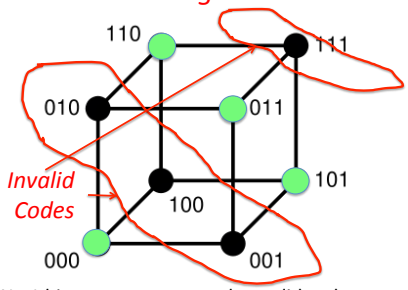
Error Detection/Correction Codes

- Memory systems generate errors (accidentally flipped-bits)
  - DRAMs store very little charge per bit
  - "Soft" errors occur occasionally when cells are struck by alpha particles or other environmental upsets
  - "Hard" errors can occur when chips permanently fail.
  - Problem gets worse as memories get denser and larger
- Memories protected against failures with EDC/ECC
- Extra bits are added to each data-word
  - Used to detect and/or correct faults in the memory system
  - Each data word value mapped to unique *code word*
  - A fault changes valid code word to invalid one, which can be detected

Hamming Distance: 8 code words

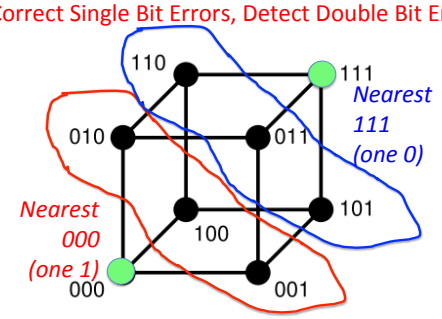


Hamming Distance 2: Detection
Detect Single Bit Errors



- No 1 bit error goes to another valid code
- ½ codes are valid

Hamming Distance 3: Correction
Correct Single Bit Errors, Detect Double Bit Errors



- No 2 bit error goes to another valid code; 1 bit error near
- 1/8 codes are valid

Agenda

- Dependability, Reliability, Availability metrics
- Codes for Redundancy
- Administrivia
- Error Detection and Correction in Memory
- RAID

Administrivia

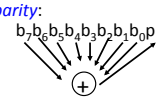
- Final Exam
  - Monday, December 10, 11:30-2:30
  - 10 questions x 10 points = 100 points/minutes
  - Room 220/230/242 Hearst Gym (assigned by course account login)
  - Comprehensive, but concentrated on material since midterm examination
  - Closed book/note, open crib sheet as before, MIPS Green Card provided
  - *Special consideration students, please contact*

Agenda

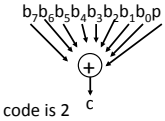
- Dependability, Reliability, Availability metrics
- Codes for Redundancy
- Administrivia
- Error Detection and Correction in Memory
- RAID

Parity: Simple Error Detection Coding

- Each data value, before it is written to memory is "tagged" with an extra bit to force the stored word to have *even parity*:



- Each word, as it is read from memory is "checked" by finding its parity (including the parity bit).



- Minimum Hamming distance of parity code is 2
- A non-zero parity indicates an error occurred:
  - 2 errors (on different bits) are not detected
  - nor any even number of errors, just odd numbers of errors are detected

Parity Example

- Data 0101 0101
- 4 ones, even parity now
- Write to memory: 0101 0101 0 to keep parity even
- Data 0101 0111
- 5 ones, odd parity now
- Write to memory: 0101 0111 1 to make parity even
- Read from memory 0101 0101 0
- 4 ones => even parity, so no error
- Read from memory 1101 0101 0
- 5 ones => odd parity, so error
- What if error in parity bit?

Suppose Want to Correct 1 Error?

- Richard Hamming came up with simple to understand mapping to allow Error Correction at minimum distance of 3
  - Single error correction, double error detection
- Called "Hamming ECC"
  - Worked weekends on relay computer with unreliable card reader, frustrated with manual restarting
  - Got interested in error correction; published 1950
  - R. W. Hamming, "Error Detecting and Correcting Codes," *The Bell System Technical Journal*, Vol. XXVI, No 2 (April 1950) pp 147-160.

Graphic of Hamming Code

Bit position		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Encoded data bits		p1	p2	d1	p4	d2	d3	d4	p8	d5	d6	d7	d8	d9	d10	d11
Parity bit coverage	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

- http://en.wikipedia.org/wiki/Hamming_code

Hamming ECC

5. Set parity bits to create even parity for each group
- A byte of data: 10011010
- Create the coded word, leaving spaces for the parity bits:
- _ _ 1 _ 0 0 1 _ 1 0 1 0

  0 0 0 0 0 0 0 0 0 1 1 1

  1 2 3 4 5 6 7 8 9 0 1 2
- Calculate the parity bits

Hamming ECC

- Position 1 checks bits 1,3,5,7,9,11 (bold): **?** _ **1** _ **0** 0 **1** _ **1** 0 **1** 0. set position 1 to a _: **_** _ **1** _ **0** 0 **1** _ **1** 0 **1** 0
- Position 2 checks bits 2,3,6,7,10,11 (bold): 0 **?** **1** _ 0 **0** **1** _ 1 **0** **1** 0. set position 2 to a _: 0 **_** **1** _ 0 **0** **1** _ 1 **0** **1** 0
- Position 4 checks bits 4,5,6,7,12 (bold): 0 1 1 **?** **0** **0** **1** _ 1 0 1 **0**. set position 4 to a _: 0 1 1 **_** **0** **0** **1** _ 1 0 1 **0**
- Position 8 checks bits 8,9,10,11,12: 0 1 1 1 0 0 1 **?** **1** **0** **1** **0**. set position 8 to a _: 0 1 1 1 0 0 1 **_** **1** **0** **1** **0**

Hamming ECC

- Final code word: 01110010 1010
- Data word: 1 001 1010

Hamming ECC Error Check

- Suppose receive 01110010 1110

		0	1	1	1	0	0	1	0	1	1	1	0			
Bit position		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Encoded data bits		p1	p2	d1	p4	d2	d3	d4	p8	d5	d6	d7	d8	d9	d10	d11
Parity bit coverage	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

Hamming ECC Error Correct

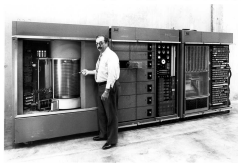
- Flip the incorrect bit …
  01110010 1010

Agenda

- Dependability, Reliability, Availability metrics
- Codes for Redundancy
- Administrivia
- Error Detection and Correction in Memory
- RAID

Evolution of the Disk Drive

IBM 3390K, 1986

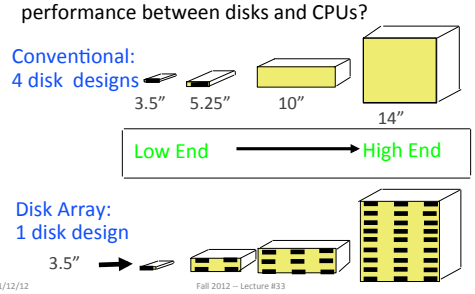


IBM RAMAC 305, 1956

Apple SCSI, 1986

Arrays of Small Disks

Can smaller disks be used to close gap in performance between disks and CPUs?



Replace Small Number of Large Disks with Large Number of Small Disks! (1988 Disks)

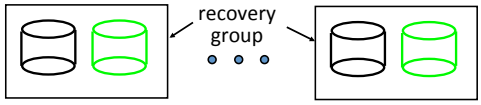
	IBM 3390K	IBM 3.5" 0061	x70	
Capacity	20 GBytes	320 MBytes	23 GBytes	
Volume	97 cu. ft.	0.1 cu. ft.	11 cu. ft.	9X
Power	3 KW	11 W	1 KW	3X
Data Rate	15 MB/s	1.5 MB/s	120 MB/s	8X
I/O Rate	600 I/Os/s	55 I/Os/s	3900 IOs/s	6X
MTTF	250 KHrs	50 KHrs	??? Hrs	
Cost	$250K	$2K	$150K	

Disk Arrays have potential for large data and I/O rates, high MB per cu. ft., high MB per KW, but what about reliability?

RAID: Redundant Arrays of (Inexpensive) Disks

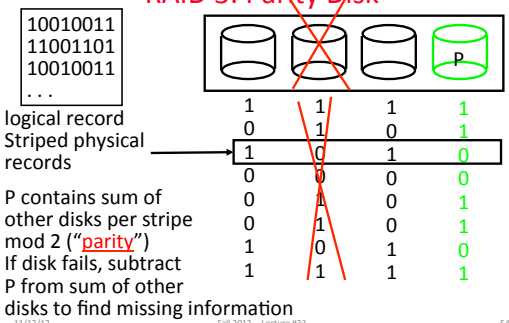
- Files are "striped" across multiple disks
- Redundancy yields high data availability
  - Availability: service still provided to user, even if some components failed
- Disks will still fail
- Contents reconstructed from data redundantly stored in the array
  - ⇒ Capacity penalty to store redundant info
  - ⇒ Bandwidth penalty to update redundant info

Redundant Arrays of Inexpensive Disks RAID 1: Disk Mirroring/Shadowing

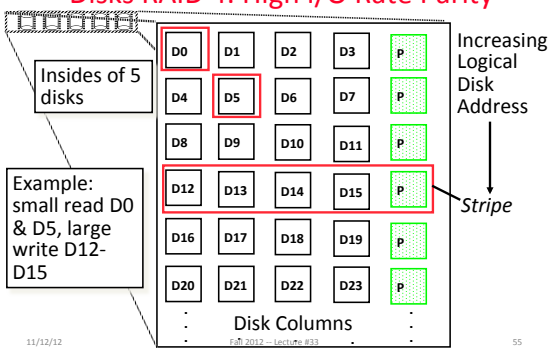


- Each disk is fully duplicated onto its "mirror"
  Very high availability can be achieved
- Bandwidth sacrifice on write:
  Logical write = two physical writes
  Reads may be optimized
- Most expensive solution: 100% capacity overhead

Redundant Array of Inexpensive Disks RAID 3: Parity Disk

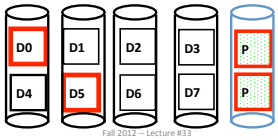


Redundant Arrays of Inexpensive Disks RAID 4: High I/O Rate Parity



Inspiration for RAID 5

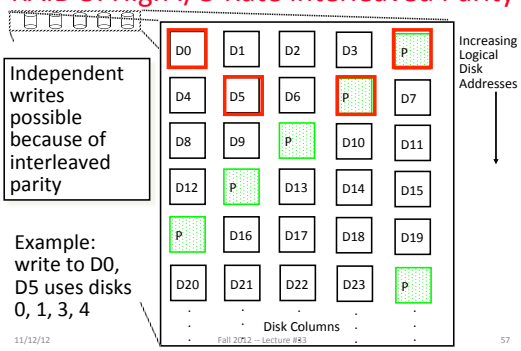
- RAID 4 works well for small reads
- Small writes (write to one disk):
  - Option 1: read other data disks, create new sum and write to Parity Disk
  - Option 2: since P has old sum, compare old data to new data, add the difference to P
- Small writes are limited by Parity Disk: Write to D0, D5 both also write to P disk



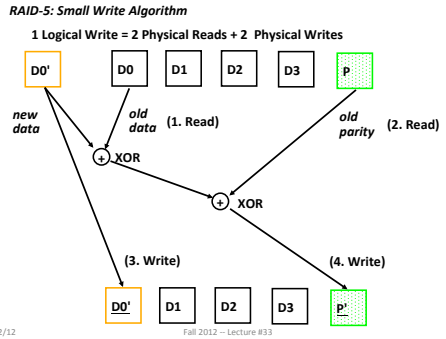
RAID 5: High I/O Rate Interleaved Parity

Independent writes possible because of interleaved parity

Example: write to D0, D5 uses disks 0, 1, 3, 4



Problems of Disk Arrays: Small Writes



RAID-I

- RAID-I (1989)
  - Consisted of a Sun 4/280 workstation with 128 MB of DRAM, four dual-string SCSI controllers, 28 5.25-inch SCSI disks and specialized disk striping software

RAID II

- 1990-1993
- Early Network Attached Storage (NAS) System running a Log Structured File System (LFS)
- Impact:
  - $25 Billion/year in 2002
  - Over $150 Billion in RAID device sold since 1990-2002
  - 200+ RAID companies (at the peak)
  - Software RAID a standard component of modern OSs

RAID II

And, in Conclusion, …

- Great Idea: Redundancy to Get Dependability
  - Spatial (extra hardware) and Temporal (retry if error)
- Reliability: MTTF & Annualized Failure Rate (AFR)
- Availability: % uptime (MTTF-MTTR/MTTF)
- Memory
  - Hamming distance 2: Parity for Single Error Detect
  - Hamming distance 3: Single Error Correction Code + encode bit position of error
  - Hamming distance 4: SEC/Double Error Detection
- CRC for many bit detection, Reed Solomon per disk sector for many bit error detection/correction