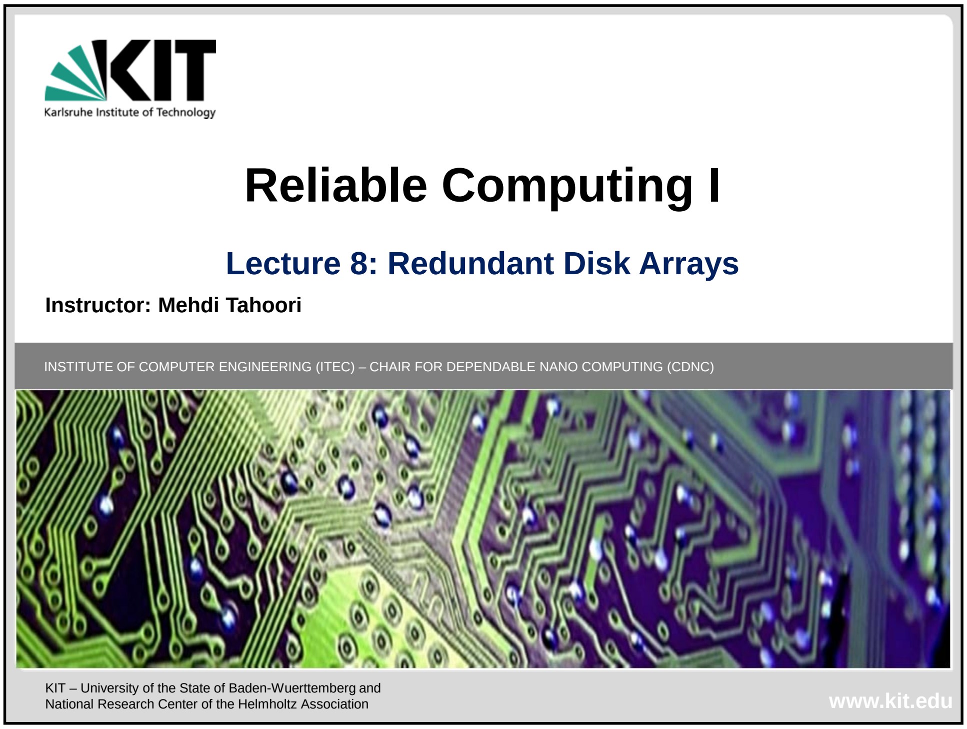


Reliable Computing I

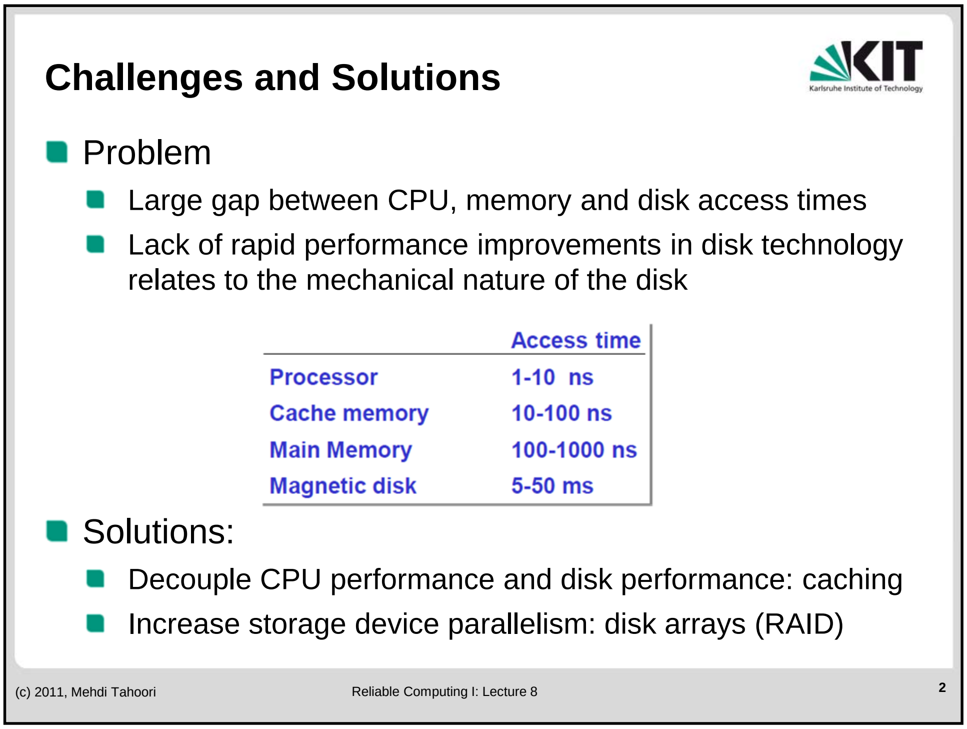
Lecture 8: Redundant Disk Arrays

Instructor: Mehdi Tahoori

INSTITUTE OF COMPUTER ENGINEERING (ITEC) – CHAIR FOR DEPENDABLE NANO COMPUTING (CDNC)

KIT – University of the State of Baden-Wuerttemberg and
National Research Center of the Helmholtz Association

www.kit.edu



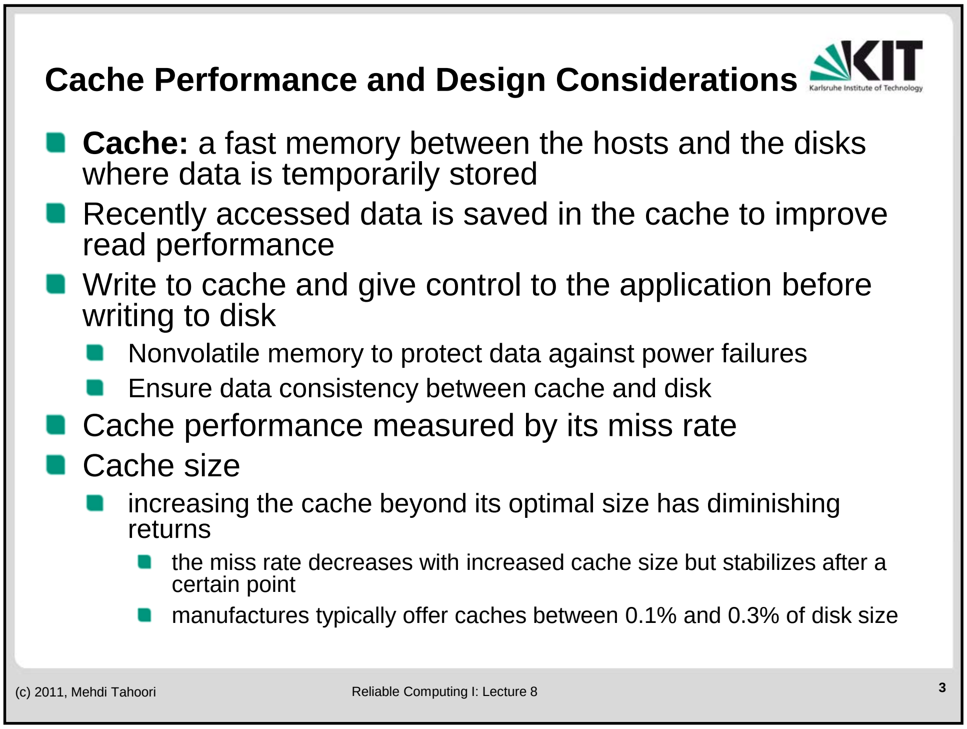
Challenges and Solutions

- Problem
  - Large gap between CPU, memory and disk access times
  - Lack of rapid performance improvements in disk technology relates to the mechanical nature of the disk

	Access time
Processor	1-10 ns
Cache memory	10-100 ns
Main Memory	100-1000 ns
Magnetic disk	5-50 ms

- Solutions:
  - Decouple CPU performance and disk performance: caching
  - Increase storage device parallelism: disk arrays (RAID)

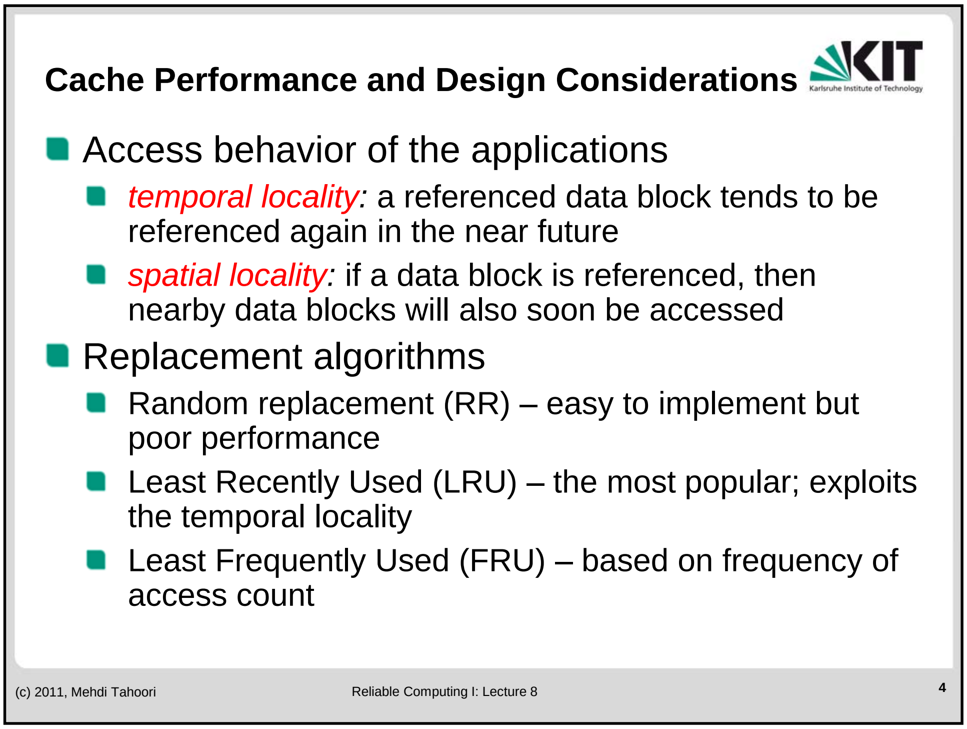
(c) 2011, Mehdi Tahoori Reliable Computing I: Lecture 8 2



Cache Performance and Design Considerations

- **Cache:** a fast memory between the hosts and the disks where data is temporarily stored
- Recently accessed data is saved in the cache to improve read performance
- Write to cache and give control to the application before writing to disk
  - Nonvolatile memory to protect data against power failures
  - Ensure data consistency between cache and disk
- Cache performance measured by its miss rate
- Cache size
  - increasing the cache beyond its optimal size has diminishing returns
    - the miss rate decreases with increased cache size but stabilizes after a certain point
    - manufactures typically offer caches between 0.1% and 0.3% of disk size

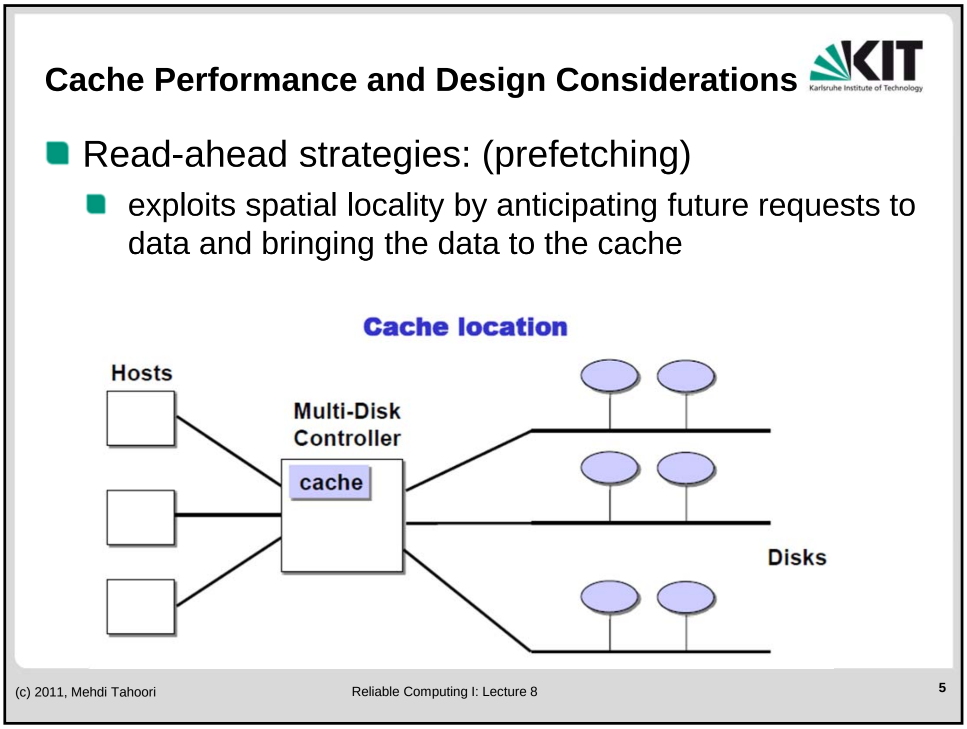
(c) 2011, Mehdi Tahoori Reliable Computing I: Lecture 8 3



Cache Performance and Design Considerations

- Access behavior of the applications
  - *temporal locality*: a referenced data block tends to be referenced again in the near future
  - *spatial locality*: if a data block is referenced, then nearby data blocks will also soon be accessed
- Replacement algorithms
  - Random replacement (RR) – easy to implement but poor performance
  - Least Recently Used (LRU) – the most popular; exploits the temporal locality
  - Least Frequently Used (FRU) – based on frequency of access count

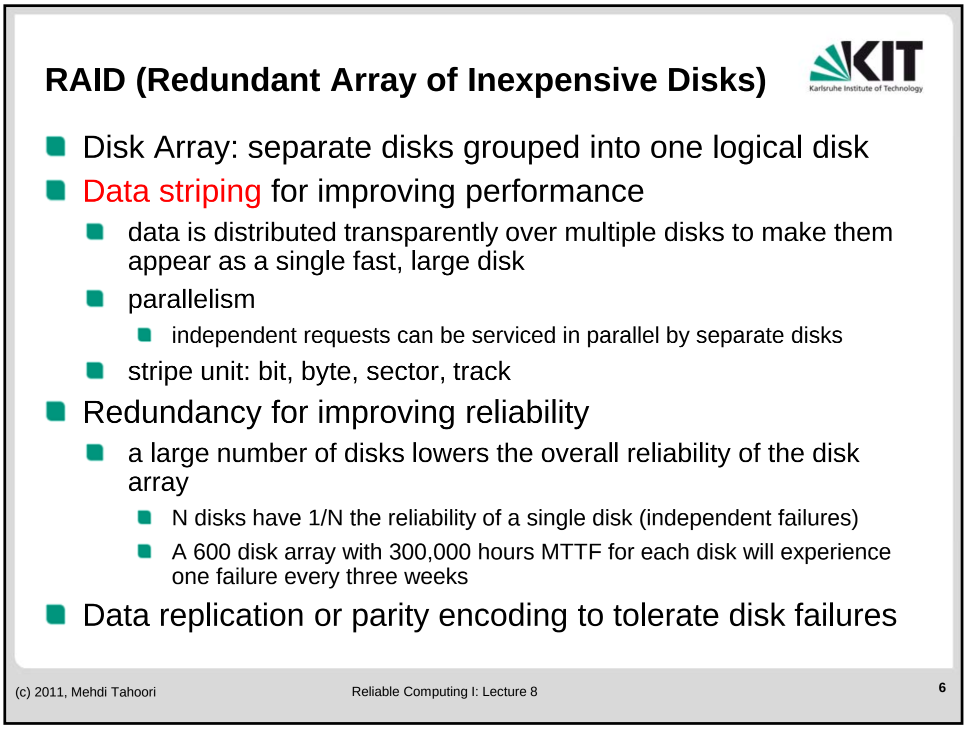
(c) 2011, Mehdi Tahoori Reliable Computing I: Lecture 8 4



Cache Performance and Design Considerations

- Read-ahead strategies: (prefetching)
  - exploits spatial locality by anticipating future requests to data and bringing the data to the cache

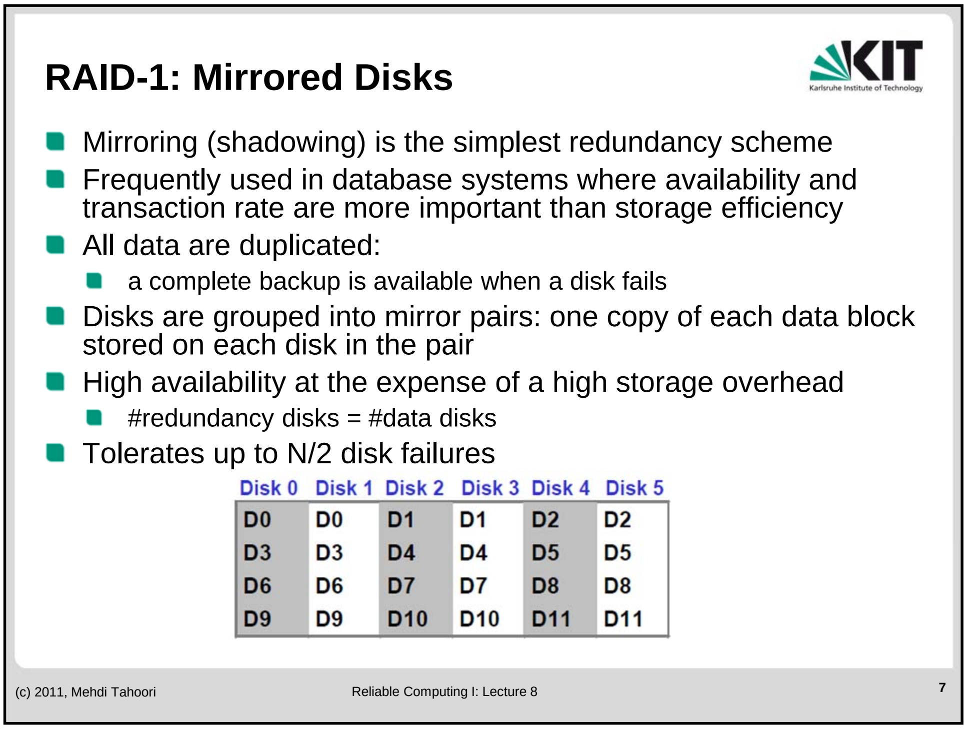
(c) 2011, Mehdi Tahoori   Reliable Computing I: Lecture 8   5



RAID (Redundant Array of Inexpensive Disks)

- Disk Array: separate disks grouped into one logical disk
- Data striping for improving performance
  - data is distributed transparently over multiple disks to make them appear as a single fast, large disk
  - parallelism
    - independent requests can be serviced in parallel by separate disks
  - stripe unit: bit, byte, sector, track
- Redundancy for improving reliability
  - a large number of disks lowers the overall reliability of the disk array
    - N disks have 1/N the reliability of a single disk (independent failures)
    - A 600 disk array with 300,000 hours MTTF for each disk will experience one failure every three weeks
- Data replication or parity encoding to tolerate disk failures

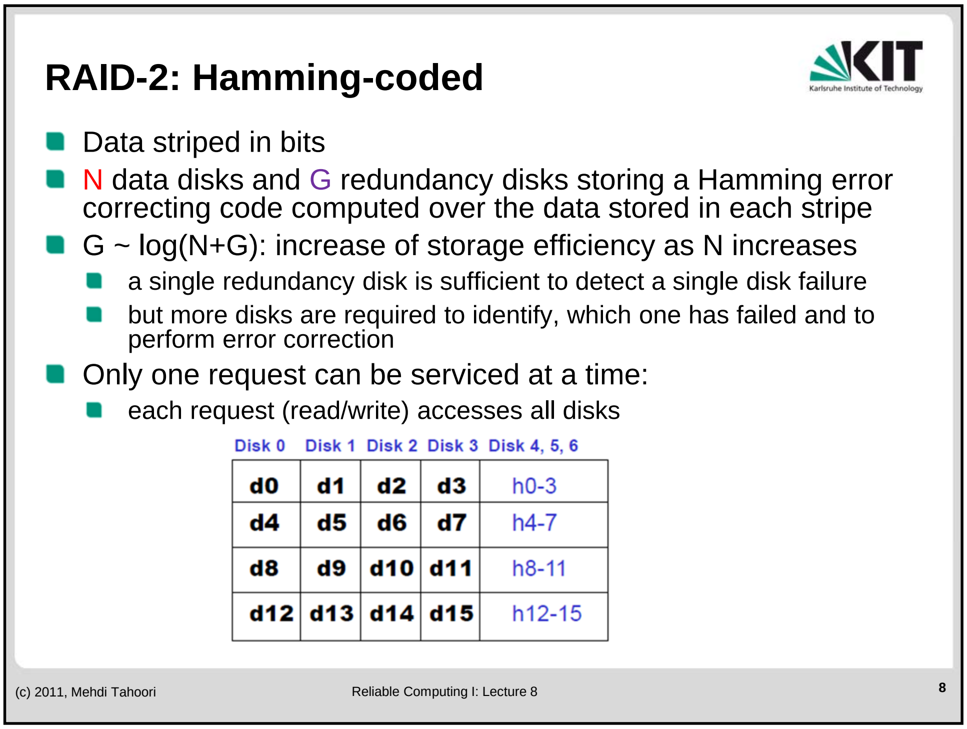
(c) 2011, Mehdi Tahoori   Reliable Computing I: Lecture 8   6



RAID-1: Mirrored Disks

- Mirroring (shadowing) is the simplest redundancy scheme
- Frequently used in database systems where availability and transaction rate are more important than storage efficiency
- All data are duplicated:
  - a complete backup is available when a disk fails
- Disks are grouped into mirror pairs: one copy of each data block stored on each disk in the pair
- High availability at the expense of a high storage overhead
  - #redundancy disks = #data disks
- Tolerates up to N/2 disk failures

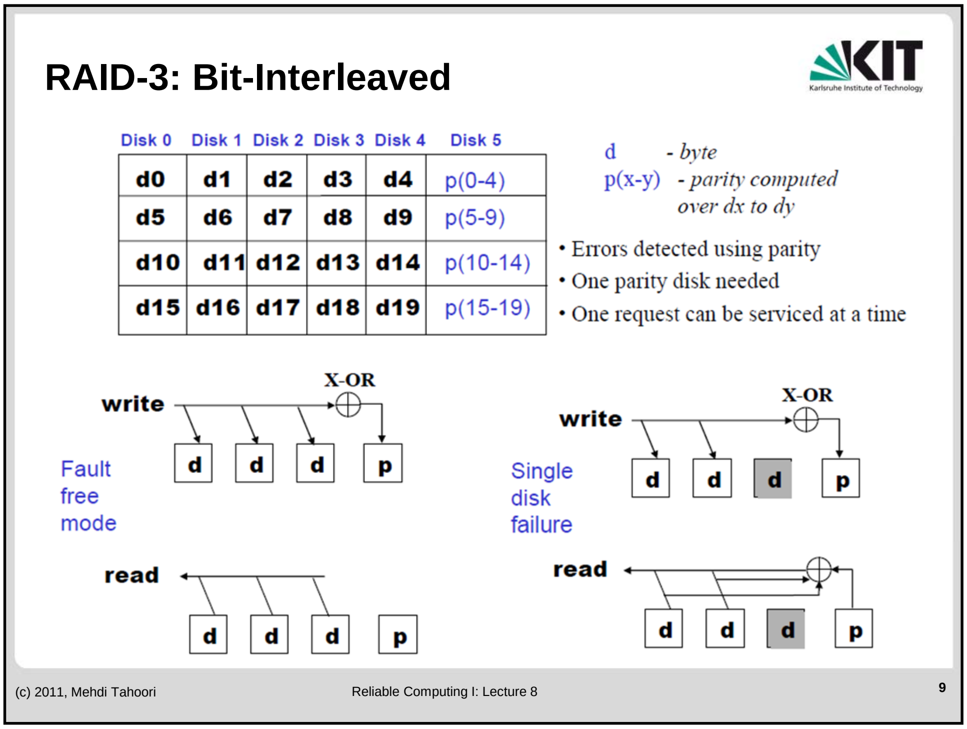
Disk 0	Disk 1	Disk 2	Disk 3	Disk 4	Disk 5
D0	D0	D1	D1	D2	D2
D3	D3	D4	D4	D5	D5
D6	D6	D7	D7	D8	D8
D9	D9	D10	D10	D11	D11



RAID-2: Hamming-coded

- Data striped in bits
- N data disks and G redundancy disks storing a Hamming error correcting code computed over the data stored in each stripe
- G ~ log(N+G): increase of storage efficiency as N increases
  - a single redundancy disk is sufficient to detect a single disk failure
  - but more disks are required to identify, which one has failed and to perform error correction
- Only one request can be serviced at a time:
  - each request (read/write) accesses all disks

Disk 0	Disk 1	Disk 2	Disk 3	Disk 4, 5, 6
d0	d1	d2	d3	h0-3
d4	d5	d6	d7	h4-7
d8	d9	d10	d11	h8-11
d12	d13	d14	d15	h12-15

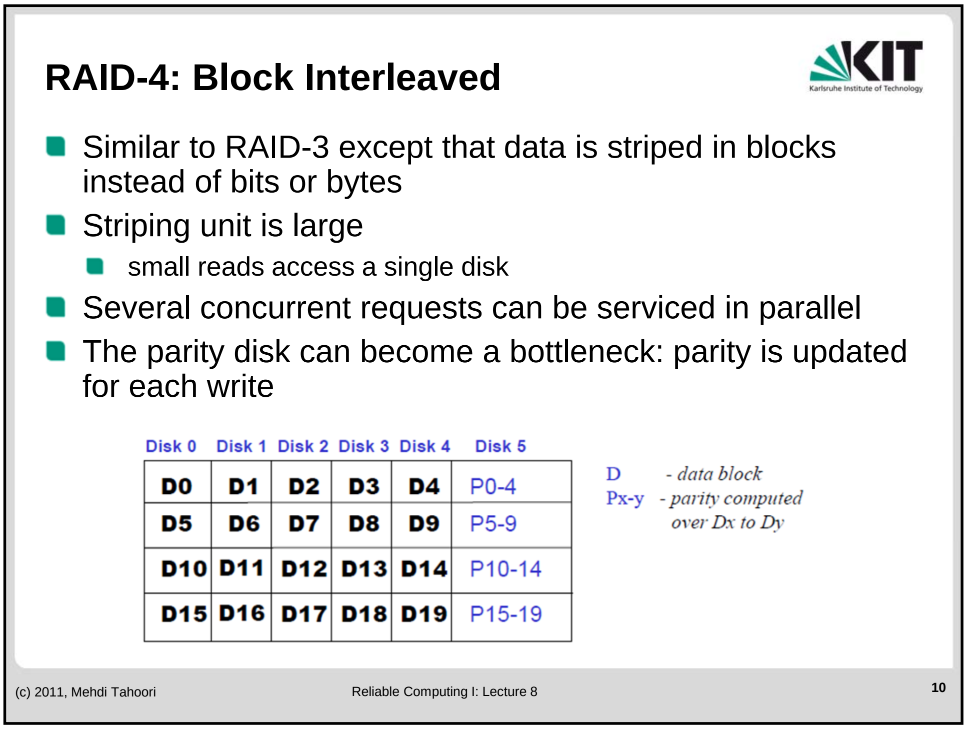


RAID-3: Bit-Interleaved

Disk 0	Disk 1	Disk 2	Disk 3	Disk 4	Disk 5
d0	d1	d2	d3	d4	p(0-4)
d5	d6	d7	d8	d9	p(5-9)
d10	d11	d12	d13	d14	p(10-14)
d15	d16	d17	d18	d19	p(15-19)

d - *byte*
p(x-y) - *parity computed over dx to dy*

- Errors detected using parity
- One parity disk needed
- One request can be serviced at a time

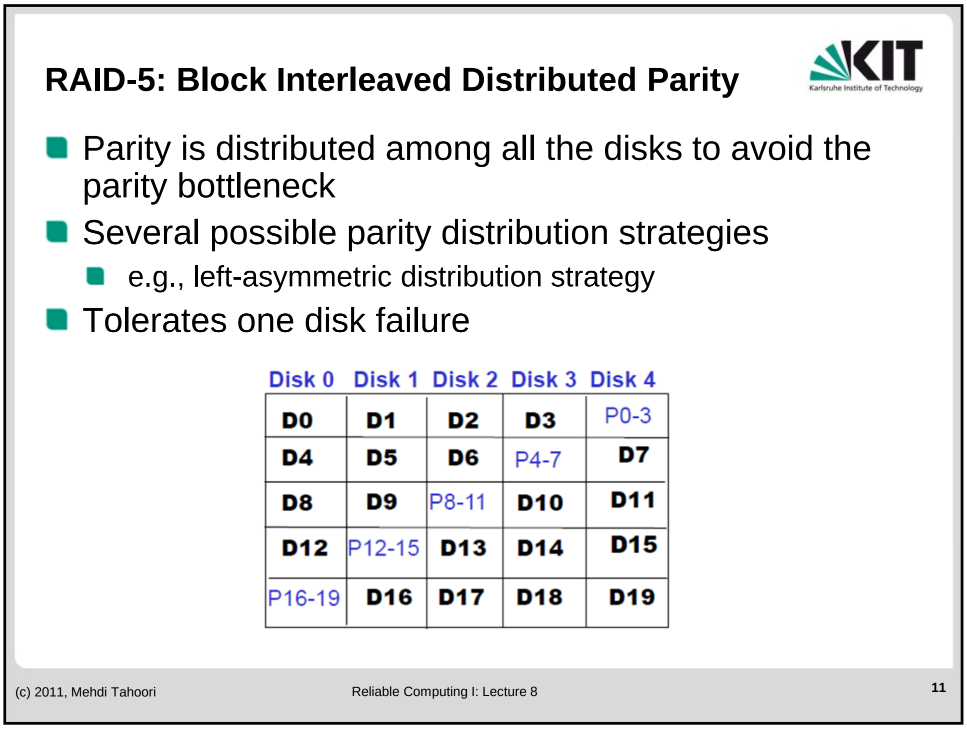


RAID-4: Block Interleaved

- Similar to RAID-3 except that data is striped in blocks instead of bits or bytes
- Striping unit is large
  - small reads access a single disk
- Several concurrent requests can be serviced in parallel
- The parity disk can become a bottleneck: parity is updated for each write

Disk 0	Disk 1	Disk 2	Disk 3	Disk 4	Disk 5
D0	D1	D2	D3	D4	P0-4
D5	D6	D7	D8	D9	P5-9
D10	D11	D12	D13	D14	P10-14
D15	D16	D17	D18	D19	P15-19

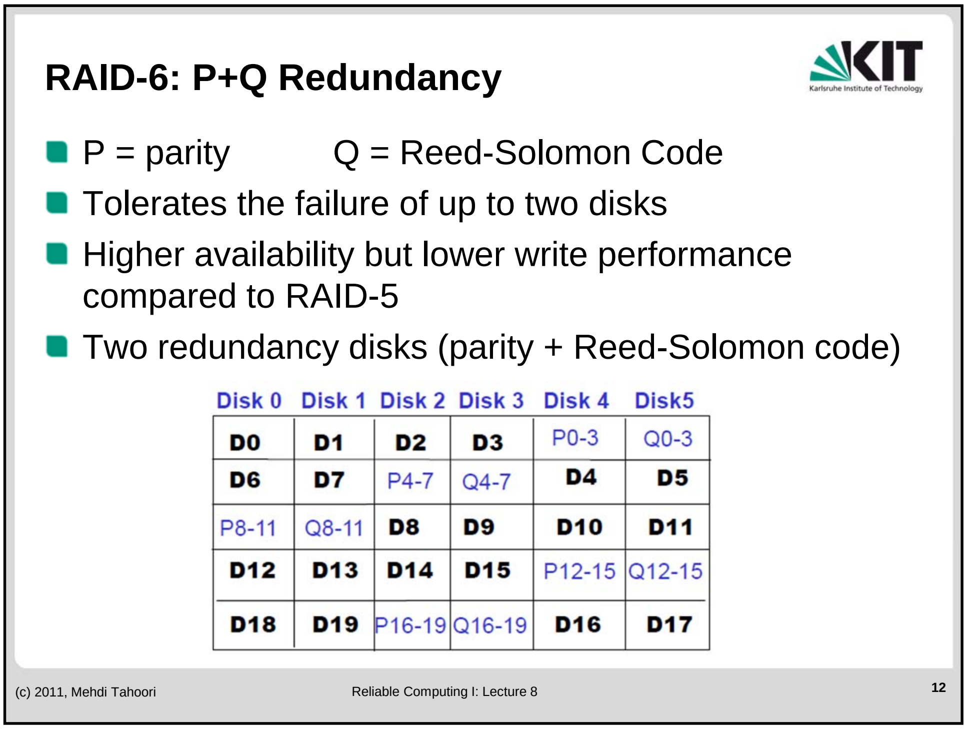
D - *data block*
Px-y - *parity computed over Dx to Dy*



RAID-5: Block Interleaved Distributed Parity

- Parity is distributed among all the disks to avoid the parity bottleneck
- Several possible parity distribution strategies
  - e.g., left-asymmetric distribution strategy
- Tolerates one disk failure

Disk 0	Disk 1	Disk 2	Disk 3	Disk 4
D0	**D1**	**D2**	**D3**	P0-3
D4	**D5**	**D6**	P4-7	**D7**
D8	**D9**	P8-11	**D10**	**D11**
D12	P12-15	**D13**	**D14**	**D15**
P16-19	**D16**	**D17**	**D18**	**D19**



RAID-6: P+Q Redundancy

- P = parity Q = Reed-Solomon Code
- Tolerates the failure of up to two disks
- Higher availability but lower write performance compared to RAID-5
- Two redundancy disks (parity + Reed-Solomon code)

Disk 0	Disk 1	Disk 2	Disk 3	Disk 4	Disk5
D0	**D1**	**D2**	**D3**	P0-3	Q0-3
D6	**D7**	P4-7	Q4-7	**D4**	**D5**
P8-11	Q8-11	**D8**	**D9**	**D10**	**D11**
D12	**D13**	**D14**	**D15**	P12-15	Q12-15
D18	**D19**	P16-19	Q16-19	**D16**	**D17**