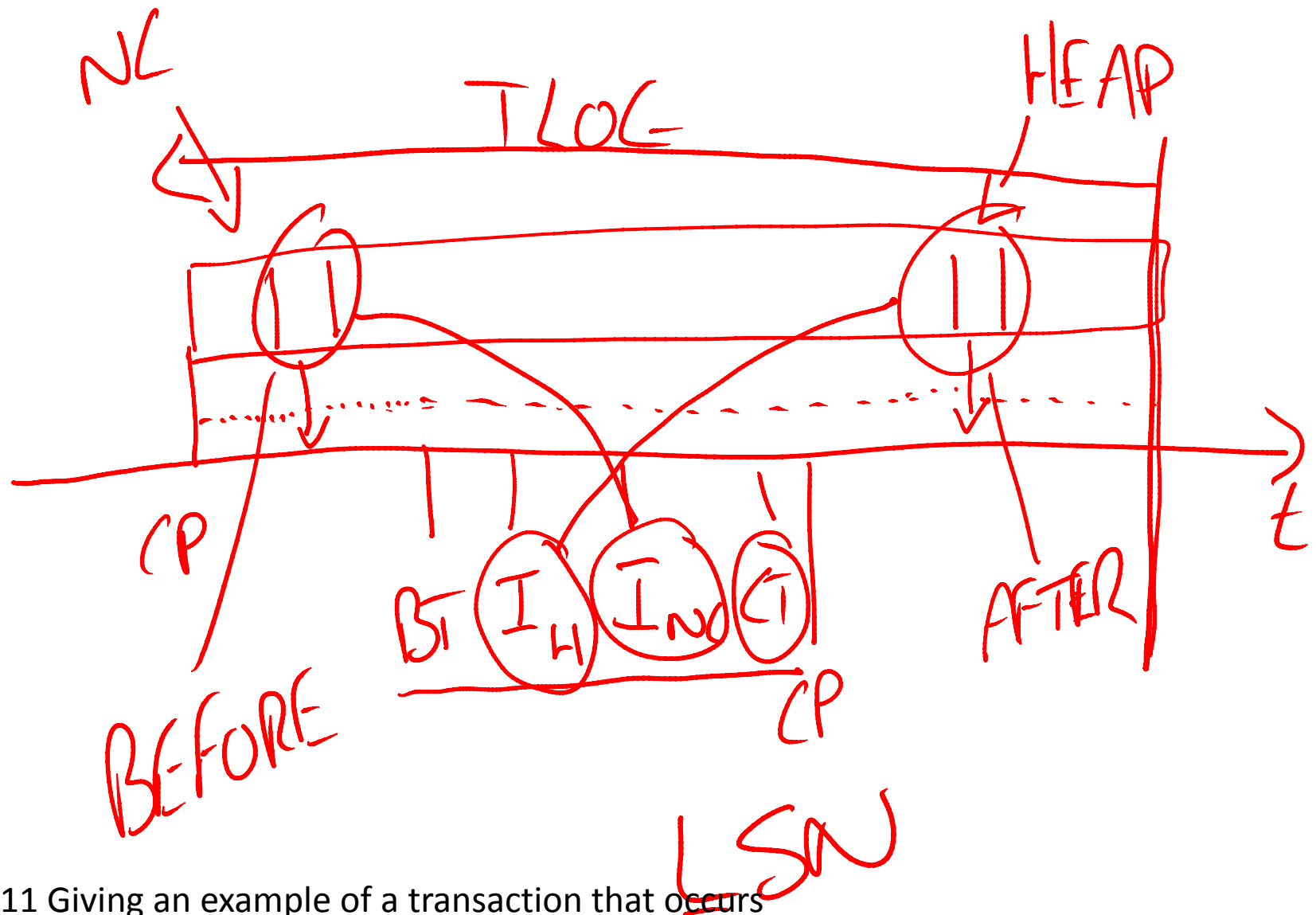
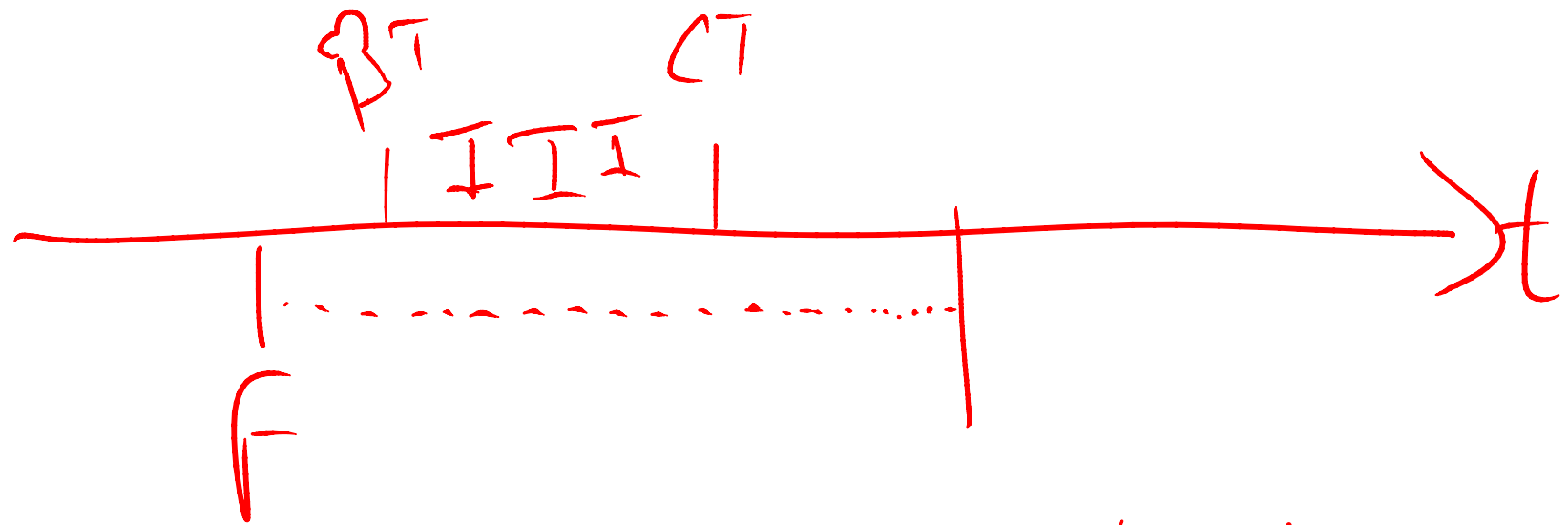




M2S11 Giving an example of a transaction that occurs during a full backup and how the log in the backup allows recovery to occur when the restore happens.

M2S11 Similar discussion to the previous slide – the same mechanism works in any recovery model.

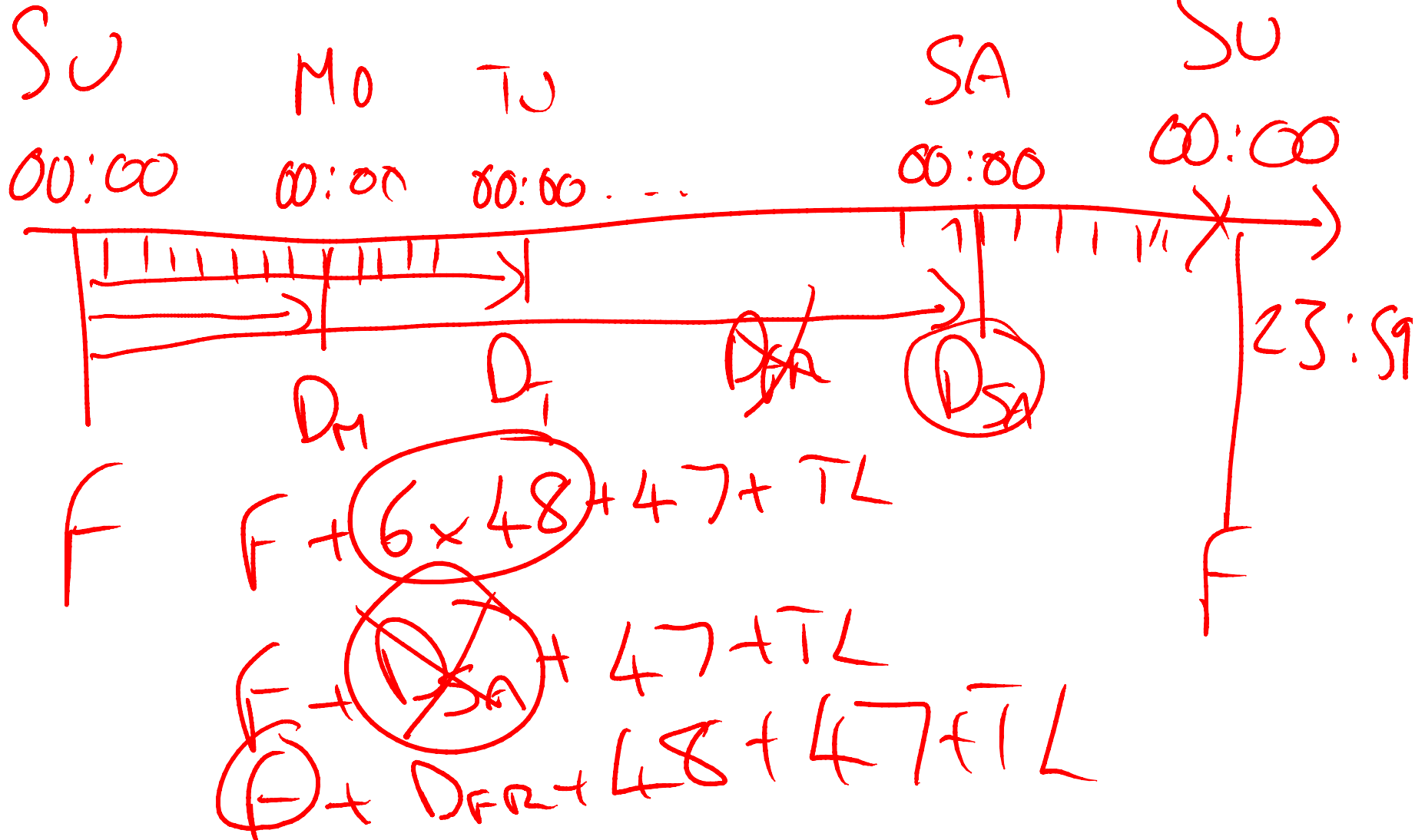
If log is filling up, query `log_reuse_wait_desc` in `sys.databases` for the reason why.



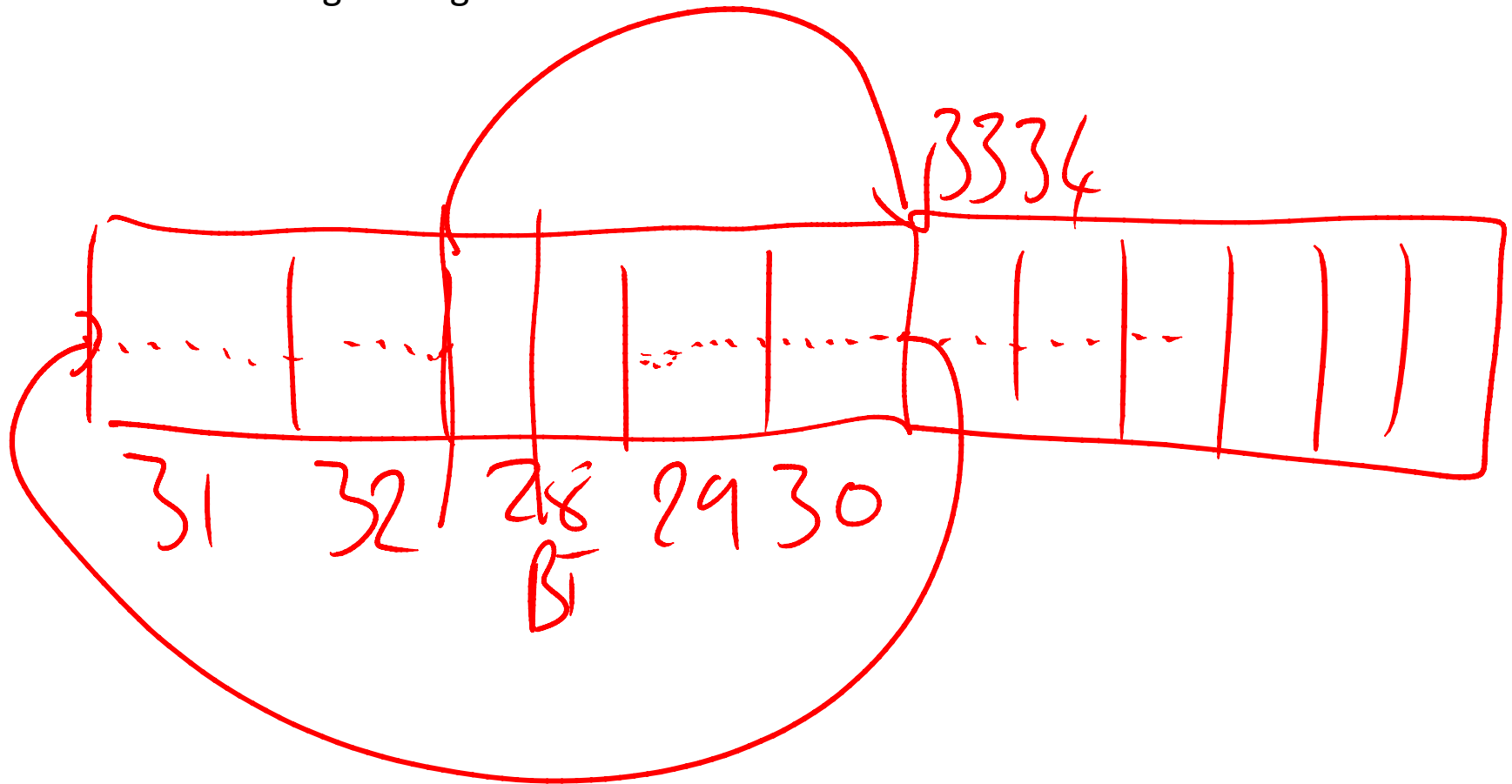
`log_reuse_wait_desc`

(F)

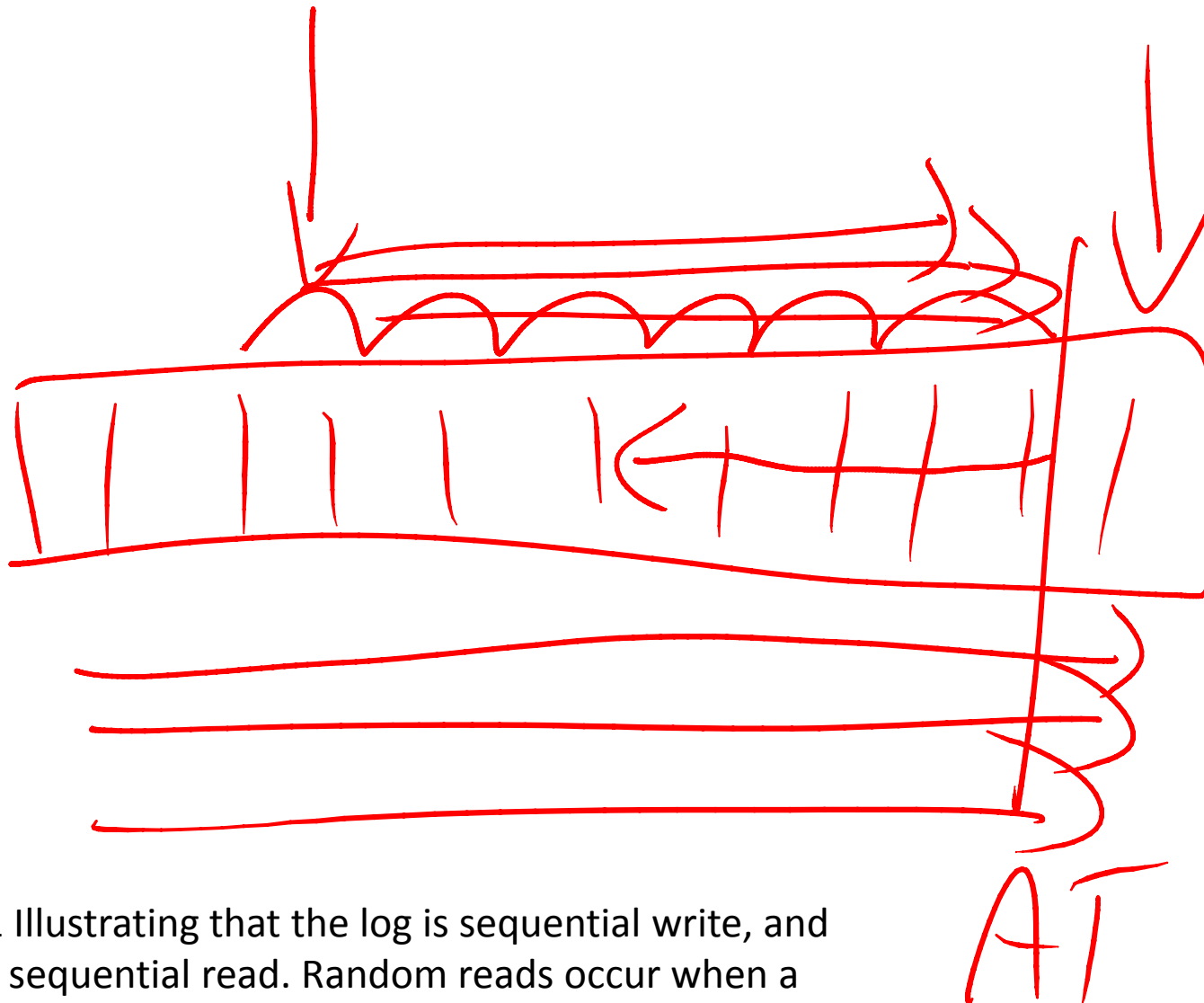
M2S14+28+34 Discussing how the various backup strategies play out when a crash occurs and a week's worth of changes need to be restored. Diff backups short-circuit *lots* of log backups.



M2S21 Illustrating VLF sequence numbers and how operations on the log follow the sequence numbers when reading the log

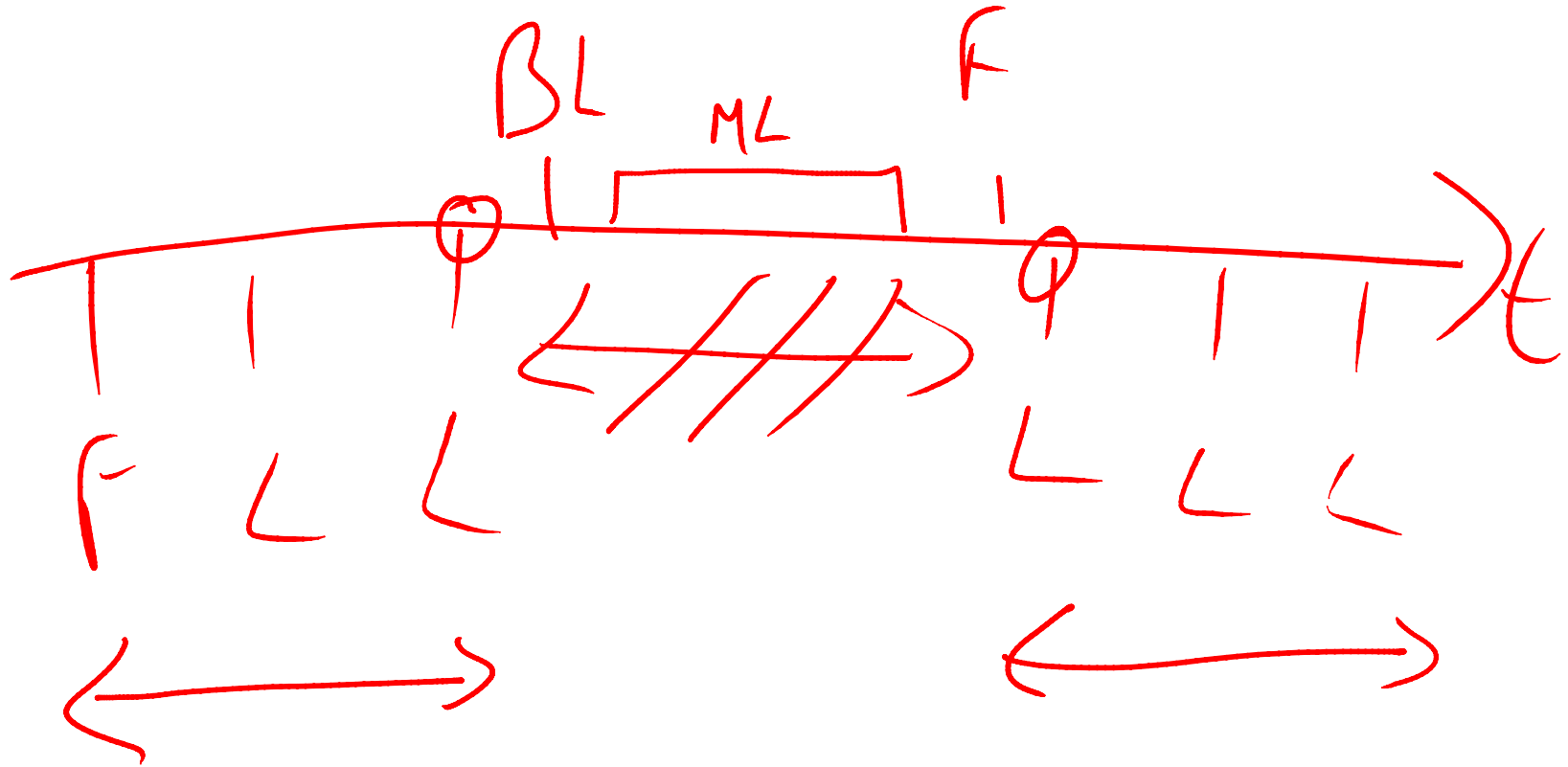


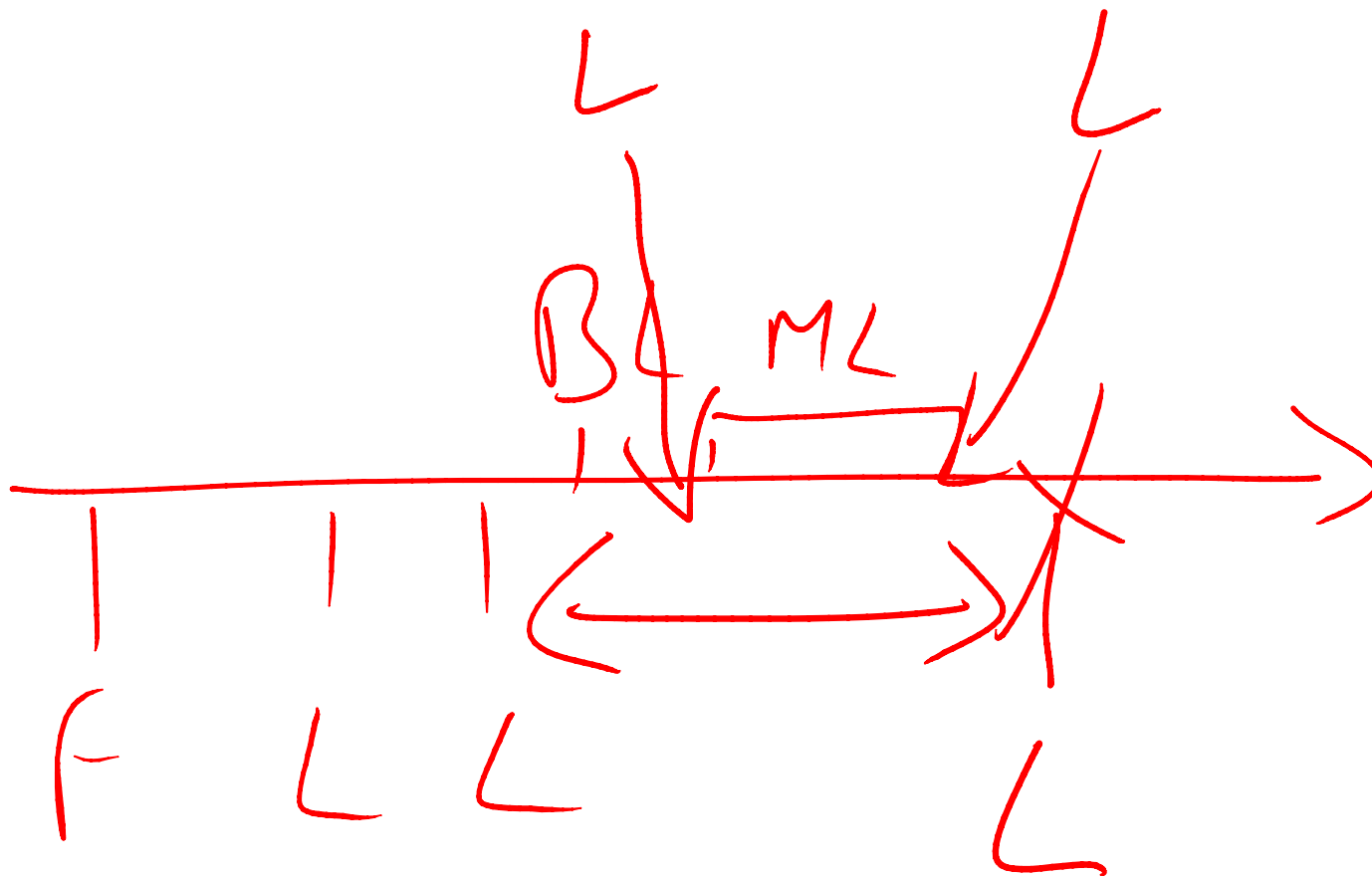
In this example, a transaction started in VLF 28, the log wrapped at VLF 30, and then the log had to grow when the end of VLF 32 is reached, jumping over VLFs 28-30.



M2S21 Illustrating that the log is sequential write, and mostly sequential read. Random reads occur when a transaction rolls back or crash recovery runs.

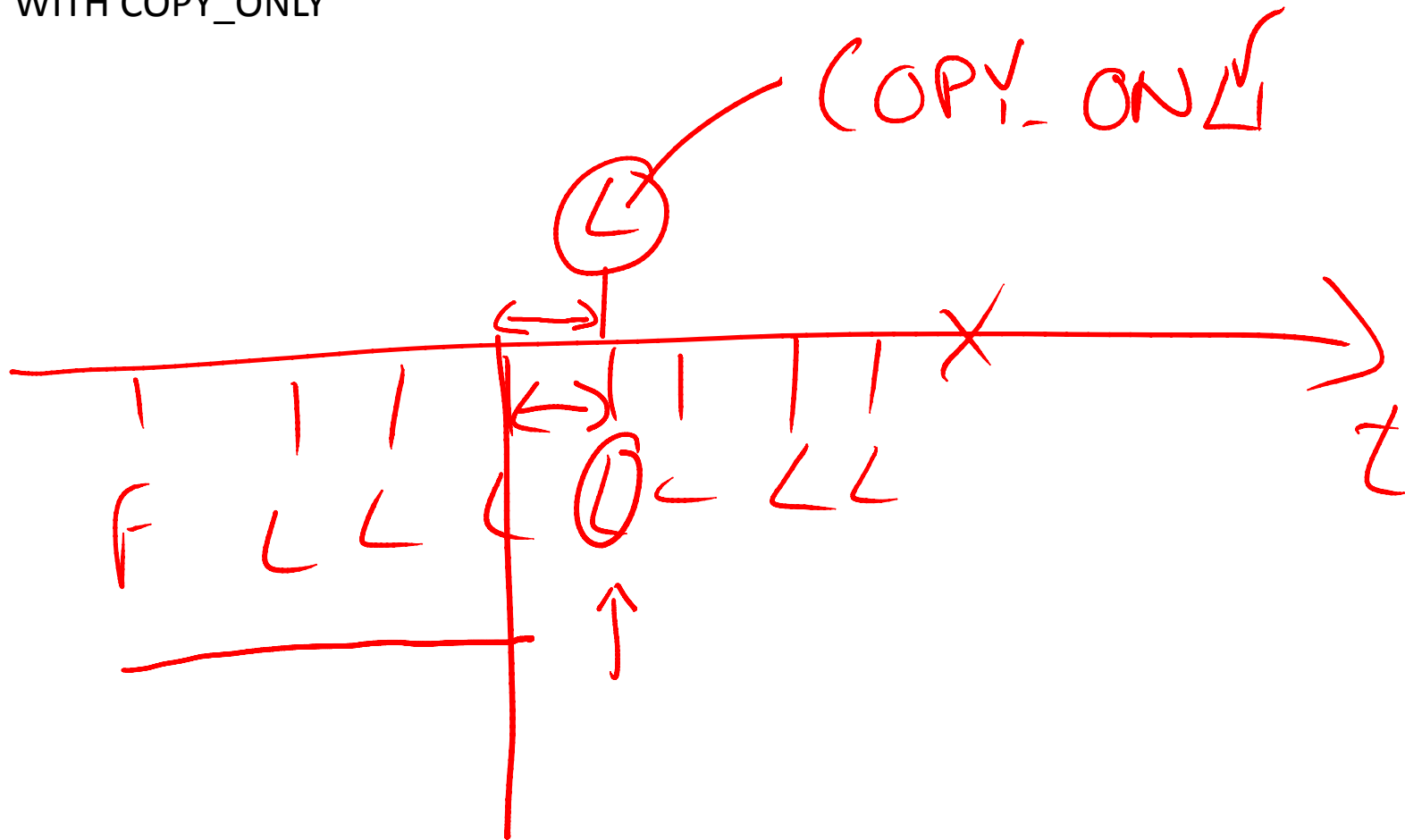
M2S26 Showing that you can not do a point-in-time restore to any time period covered by a log backup that contains a minimally-logged operation.



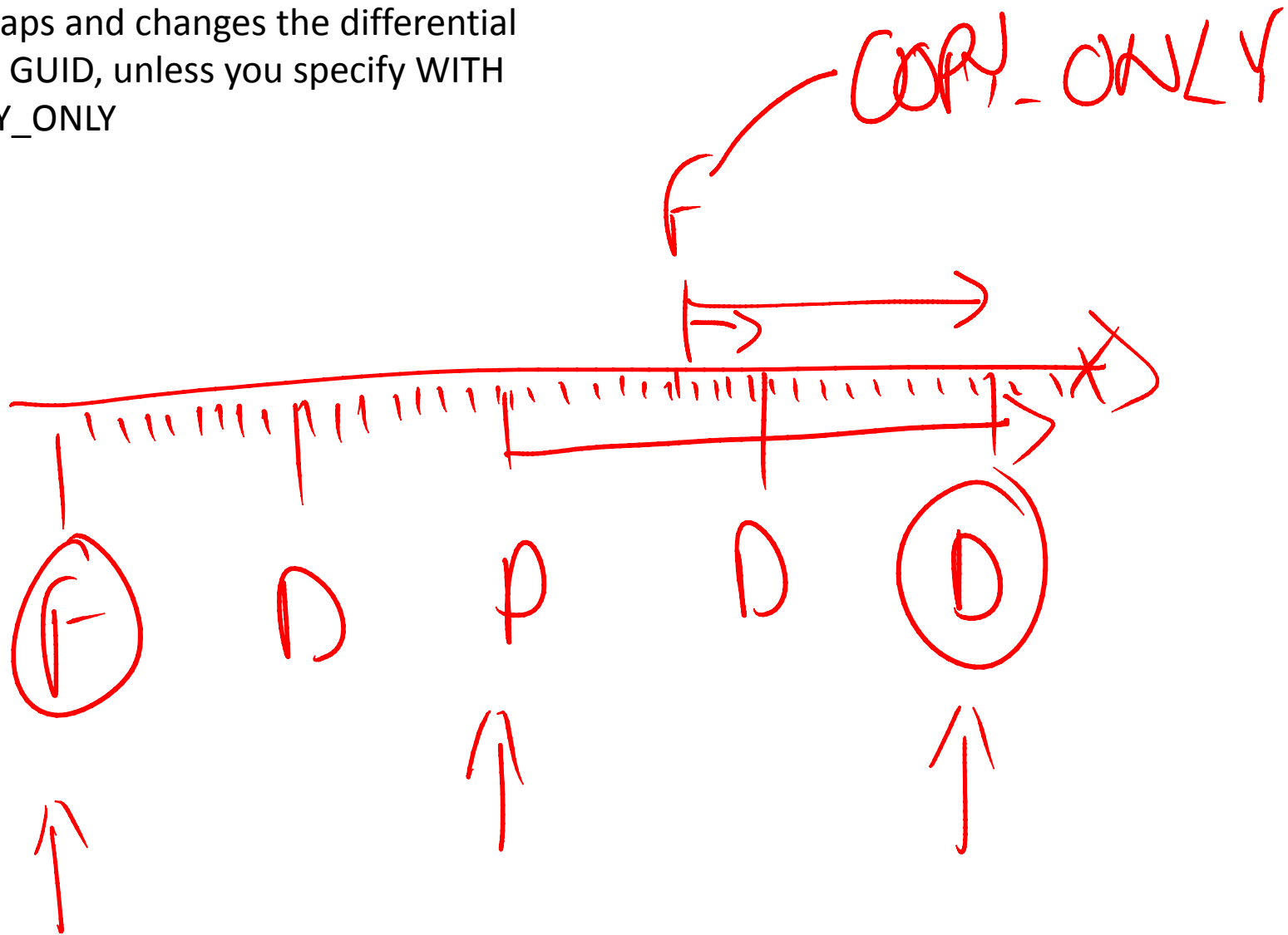


M2S26 Showing that if you want to use bulk-logged, bracket the minimally-logged operation with log backups to minimize the time you're exposed to not being able to perform a tail-of-the-log backup if the system crashes and the data files are damaged.

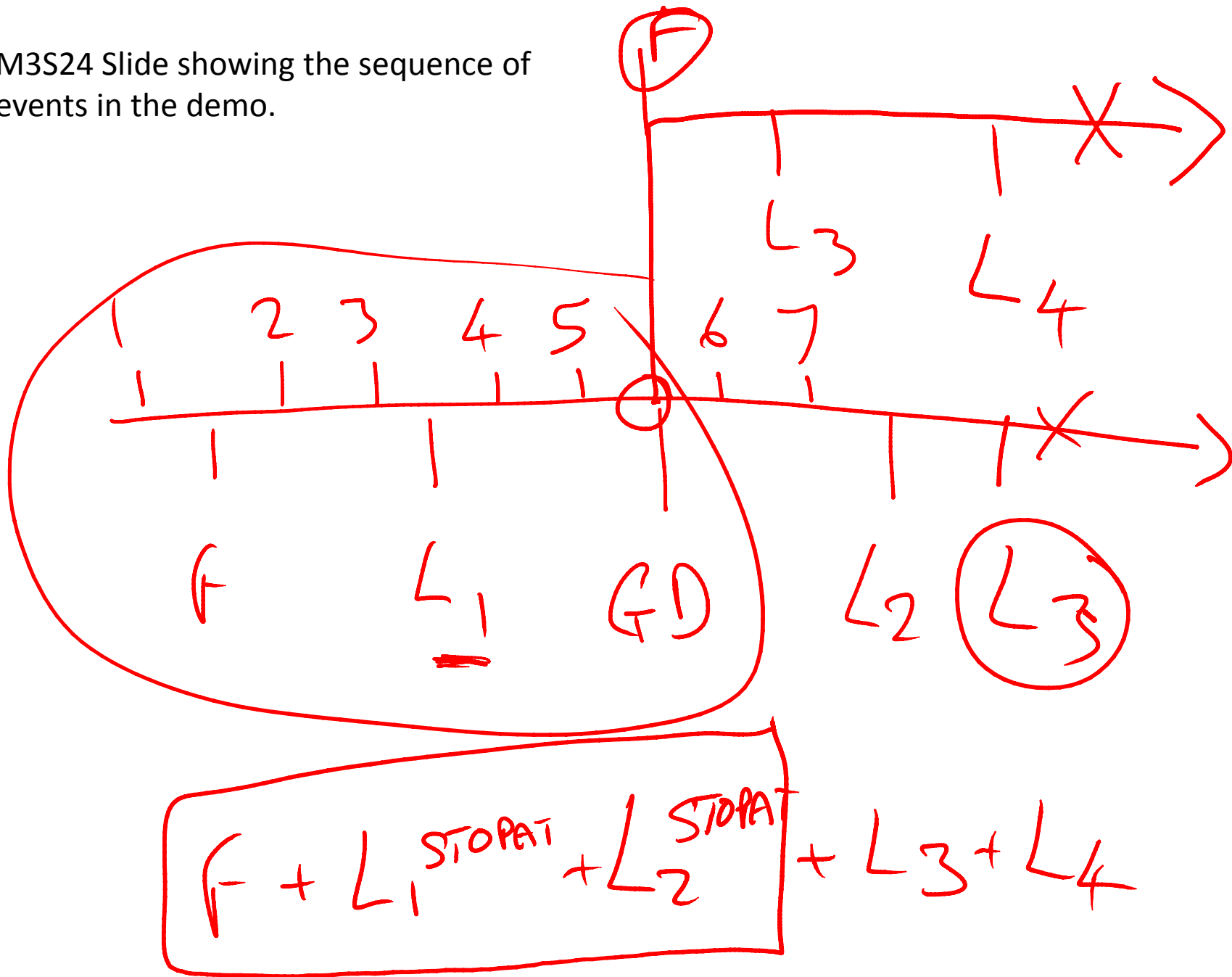
M2S38 Showing how an out-of-band log backup forms part of the log backup chain unless you specify WITH COPY_ONLY



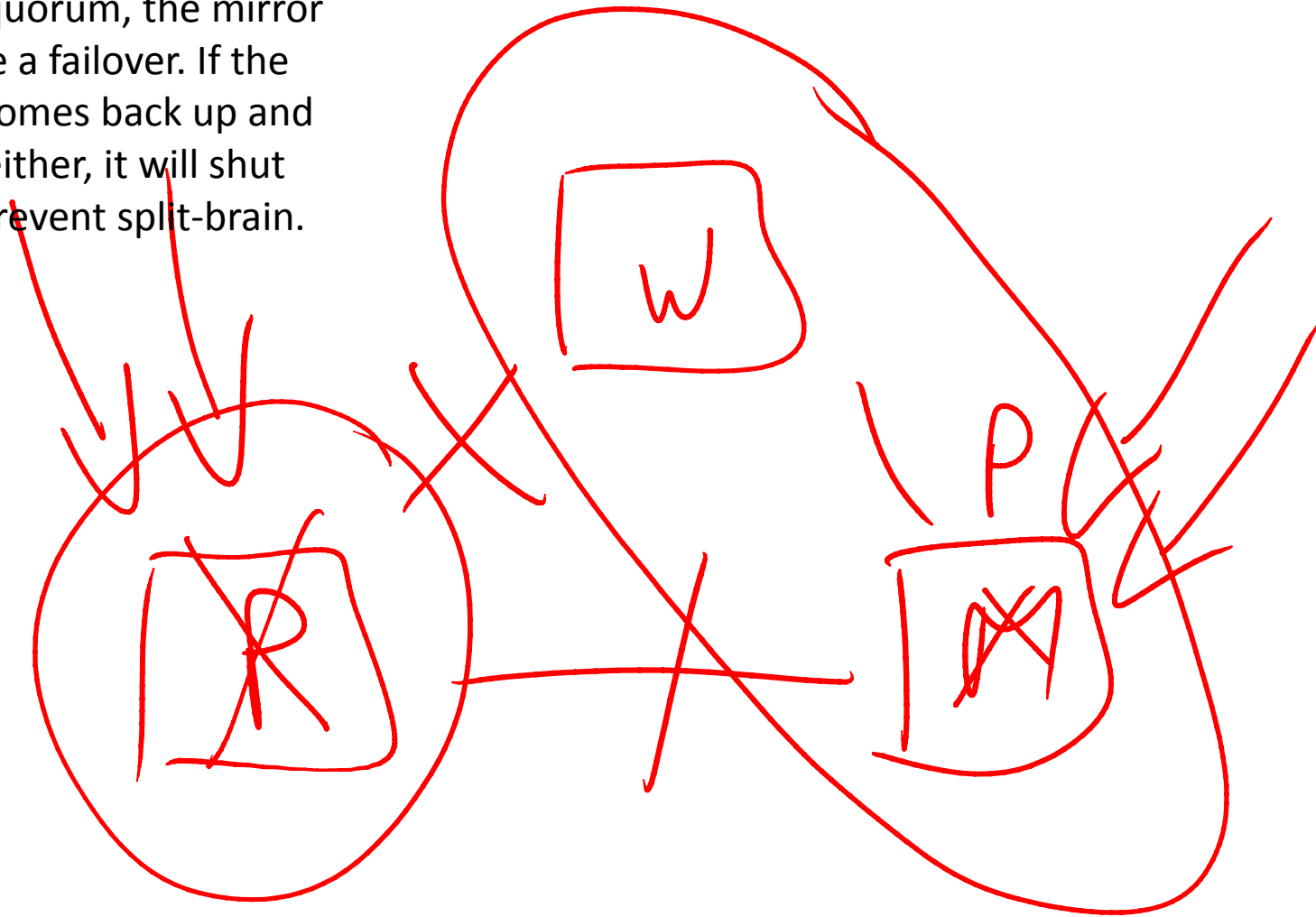
M2S38 Showing how an out-of-band full backup resets the differential bitmaps and changes the differential base GUID, unless you specify WITH COPY_ONLY



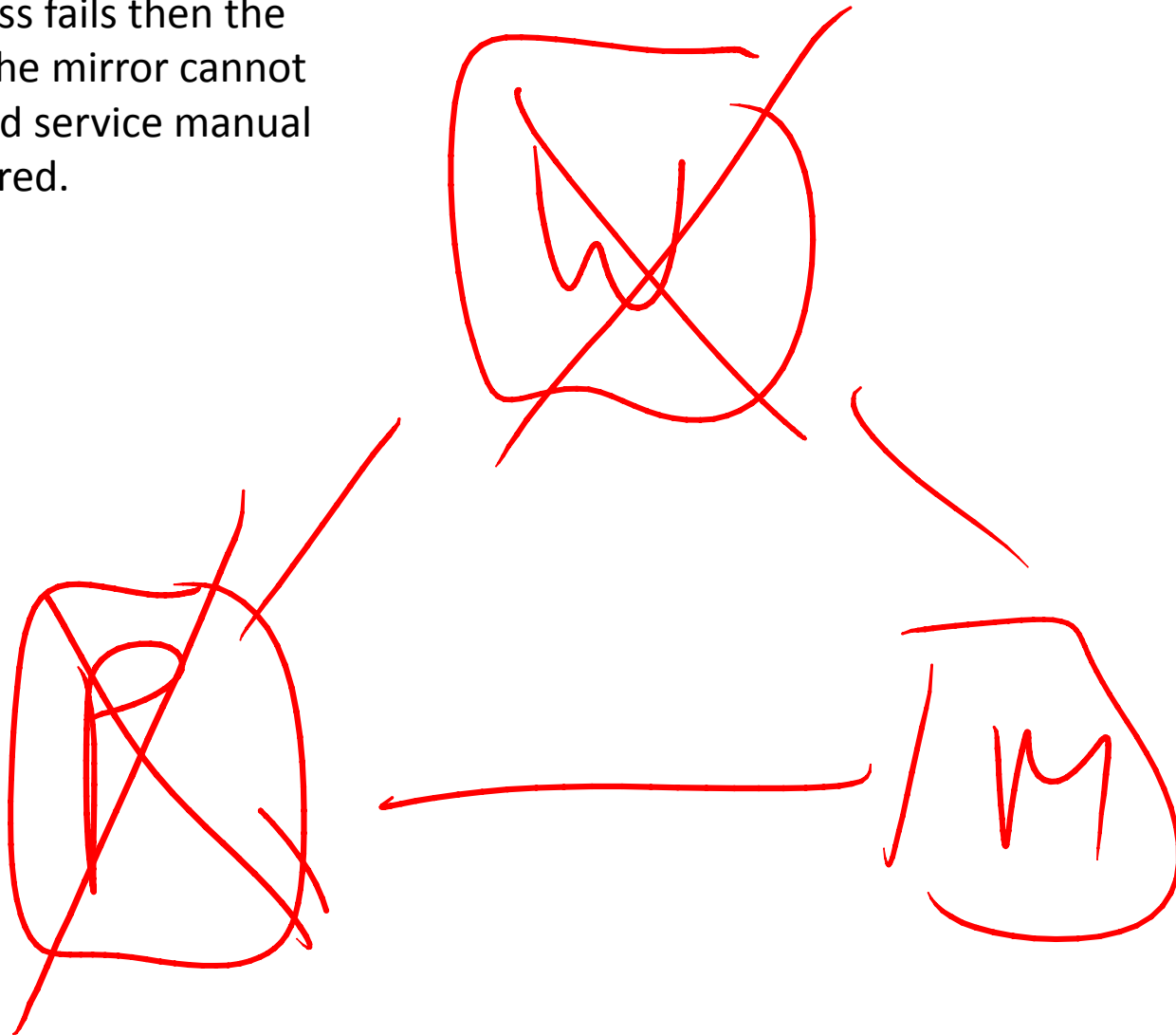
M3S24 Slide showing the sequence of events in the demo.



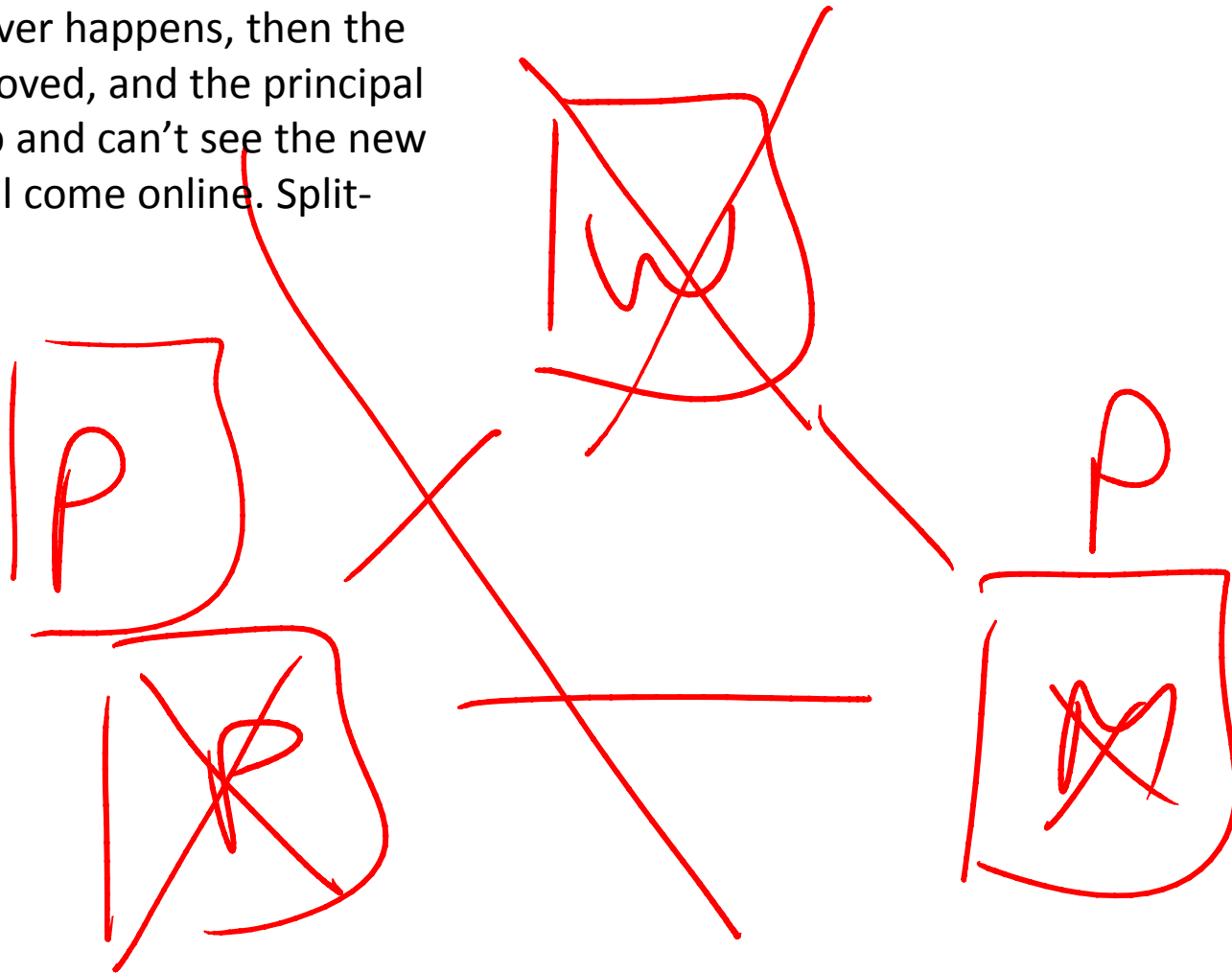
M7 If the witness and mirror can form quorum, the mirror will initiate a failover. If the principal comes back up and can't see either, it will shut down to prevent split-brain.



M7 If the witness fails then the principal fails, the mirror cannot failover. A forced service manual failover is required.



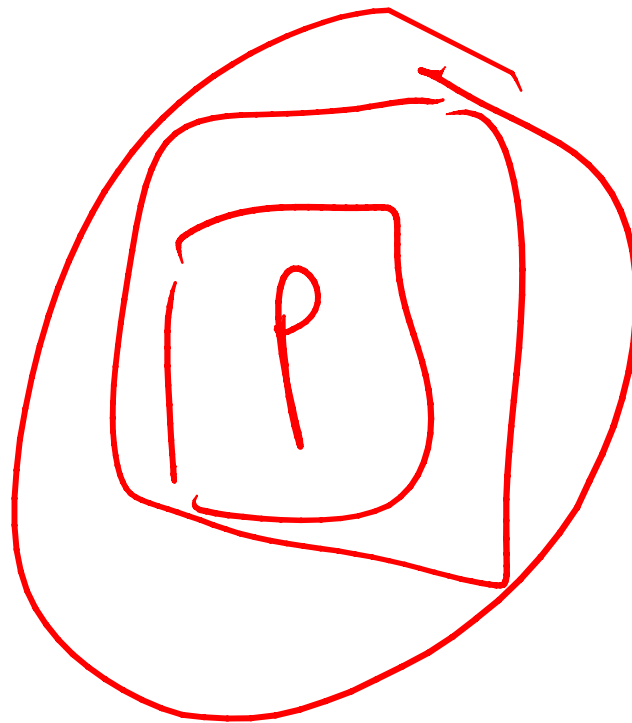
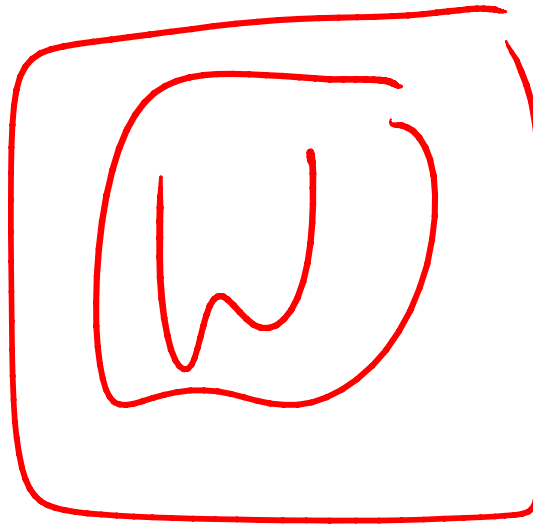
M7 If the principal crashes and a mirroring failover happens, then the witness is removed, and the principal comes back up and can't see the new principal, it will come online. Split-brain!

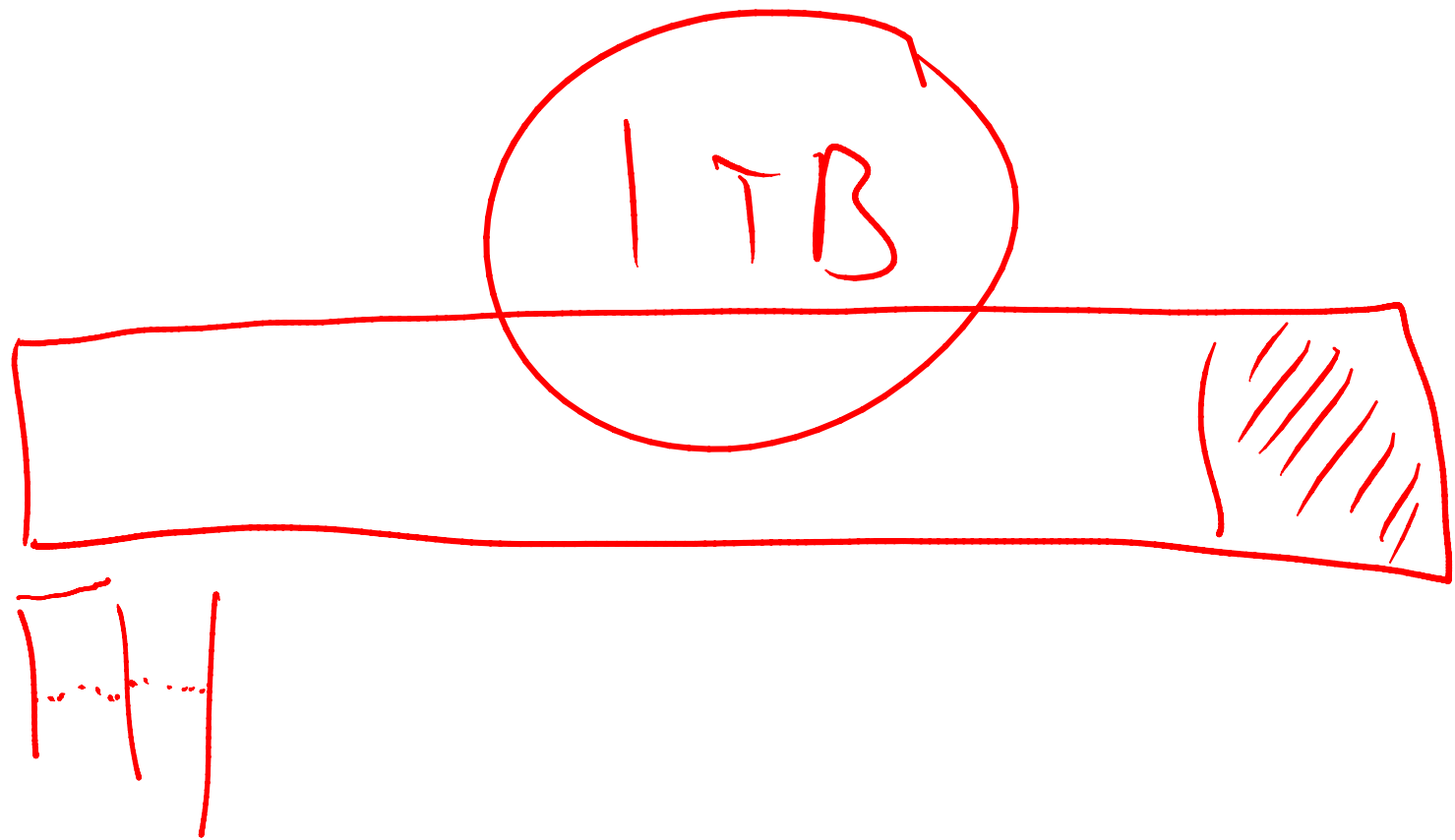




M7 To avoid unnecessary shutdowns of the principal to avoid split-brain, put the witness in the same DC as the principal (if only two DCs are available), so a network outage does not cause principal to lose contact with the witness.

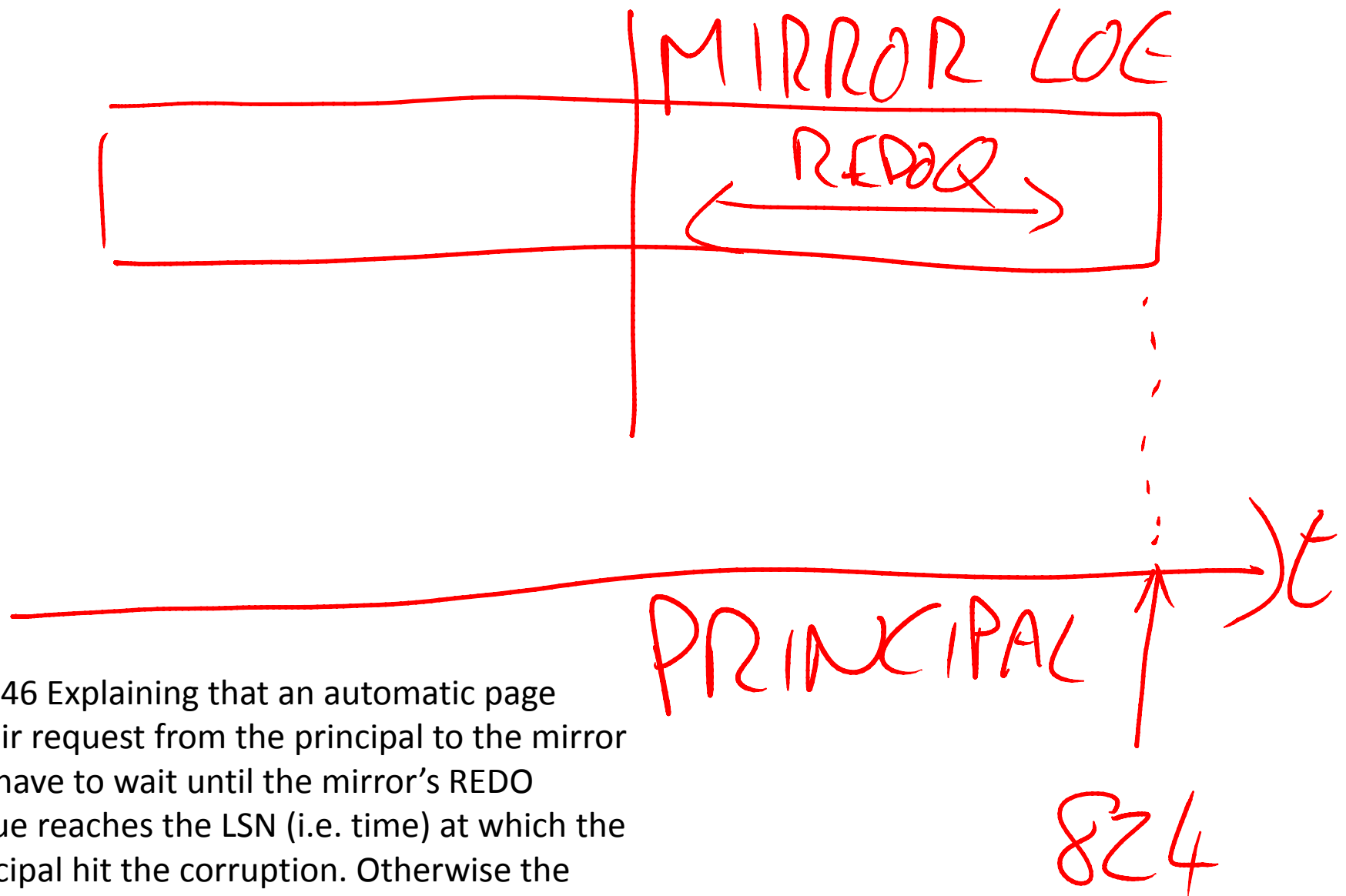
M7 Ultimate in availability, each server in mirroring topology in own DC, possibly on a failover cluster.



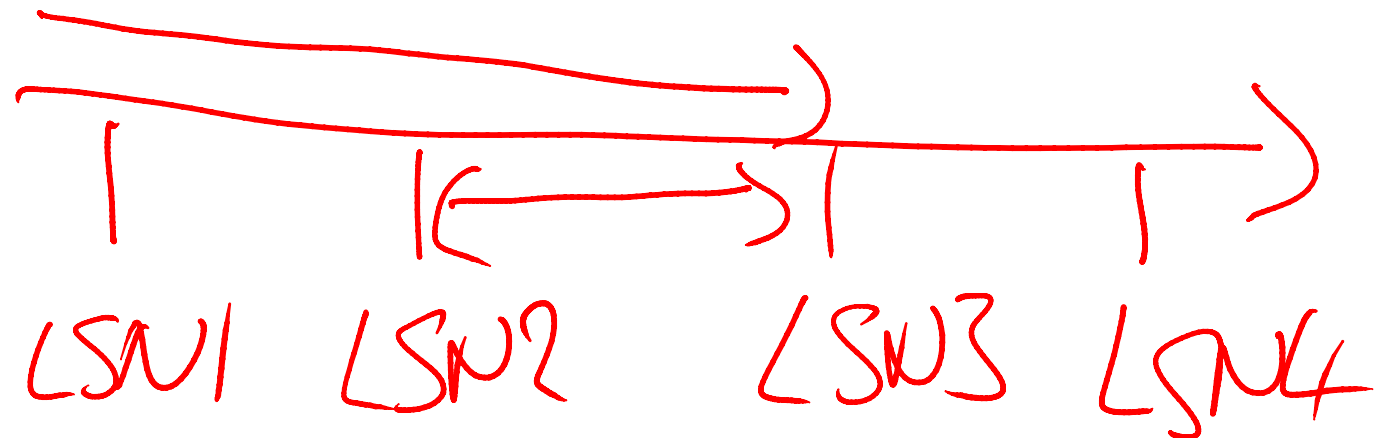


M7S11 One in full recovery model, rebuilding of large indexes becomes unpalatable. Instead consider staggered ALTER INDEX ... REORGANIZE, running for an hour or two every night.

AI REORG



M7S46 Explaining that an automatic page repair request from the principal to the mirror will have to wait until the mirror's REDO queue reaches the LSN (i.e. time) at which the principal hit the corruption. Otherwise the page image returned from the mirror may be old.

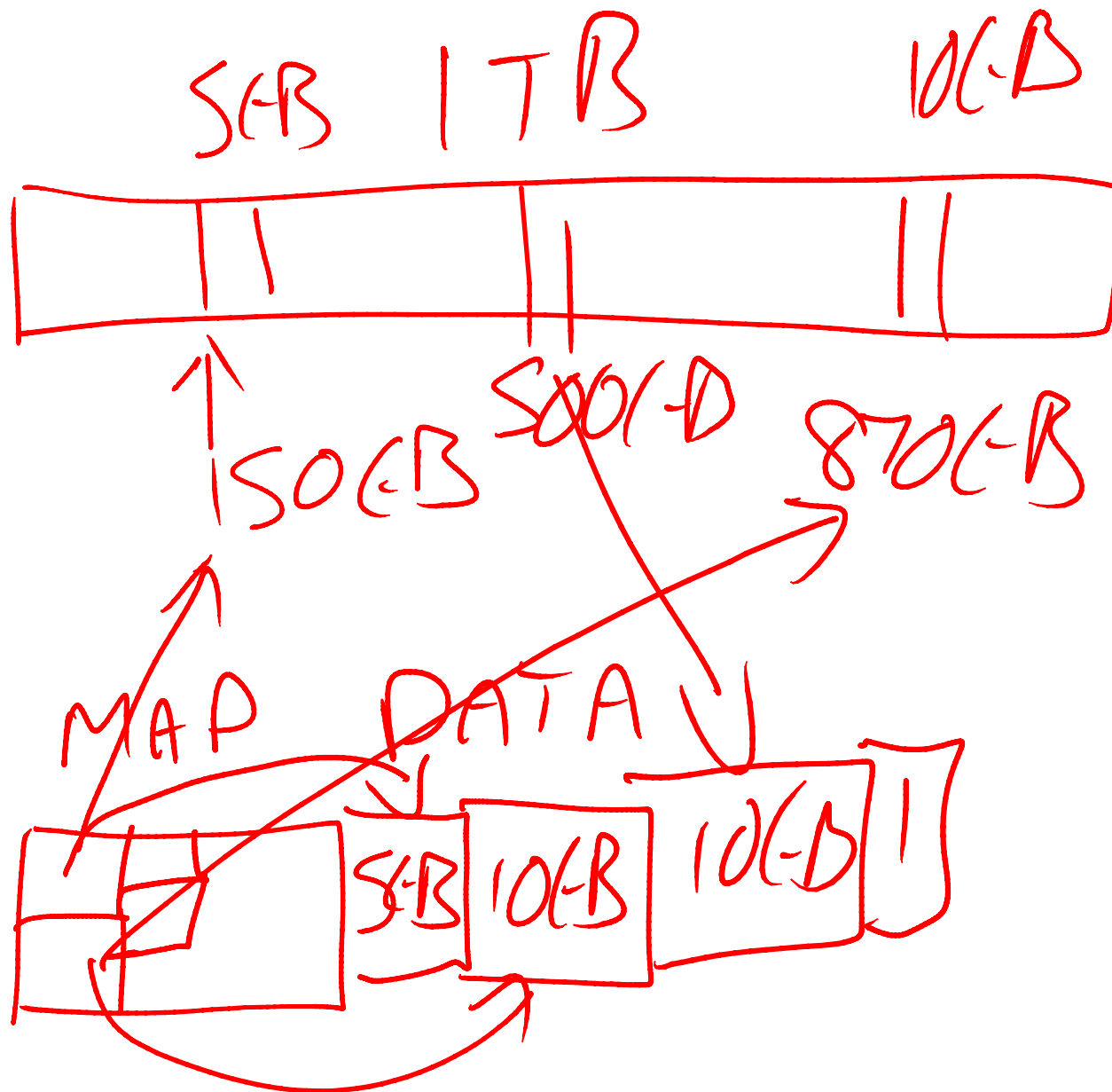


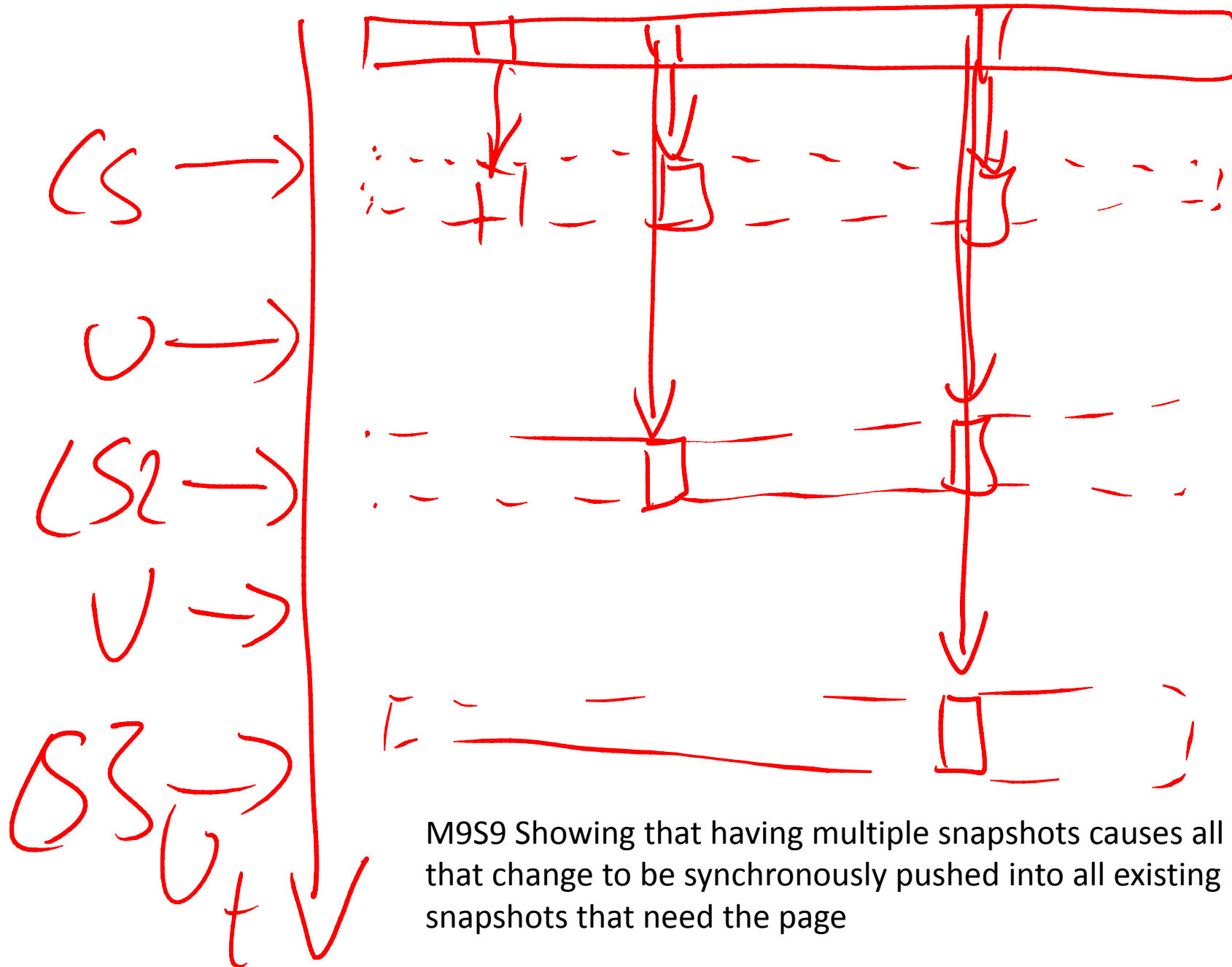
M7S55 Explaining the various LSNs involved in initialize-from-LSN creation of subscription after mirroring failover of subscriber database in SQL 2008. See the repl/dbm whitepaper for more details.

MIRRORED
TO MIRROR

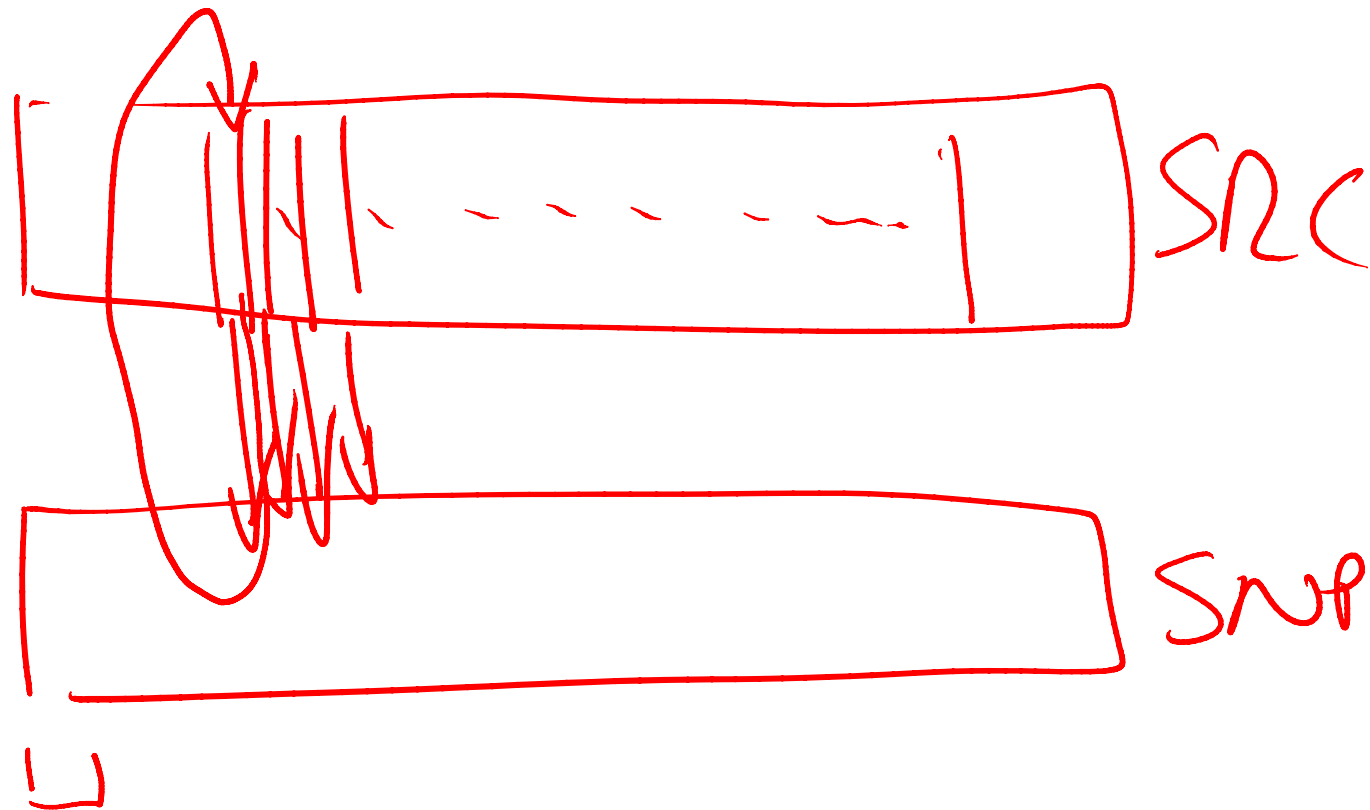
OLD
PRINCIPAL

M9S5
showing how
NTFS sparse
files work



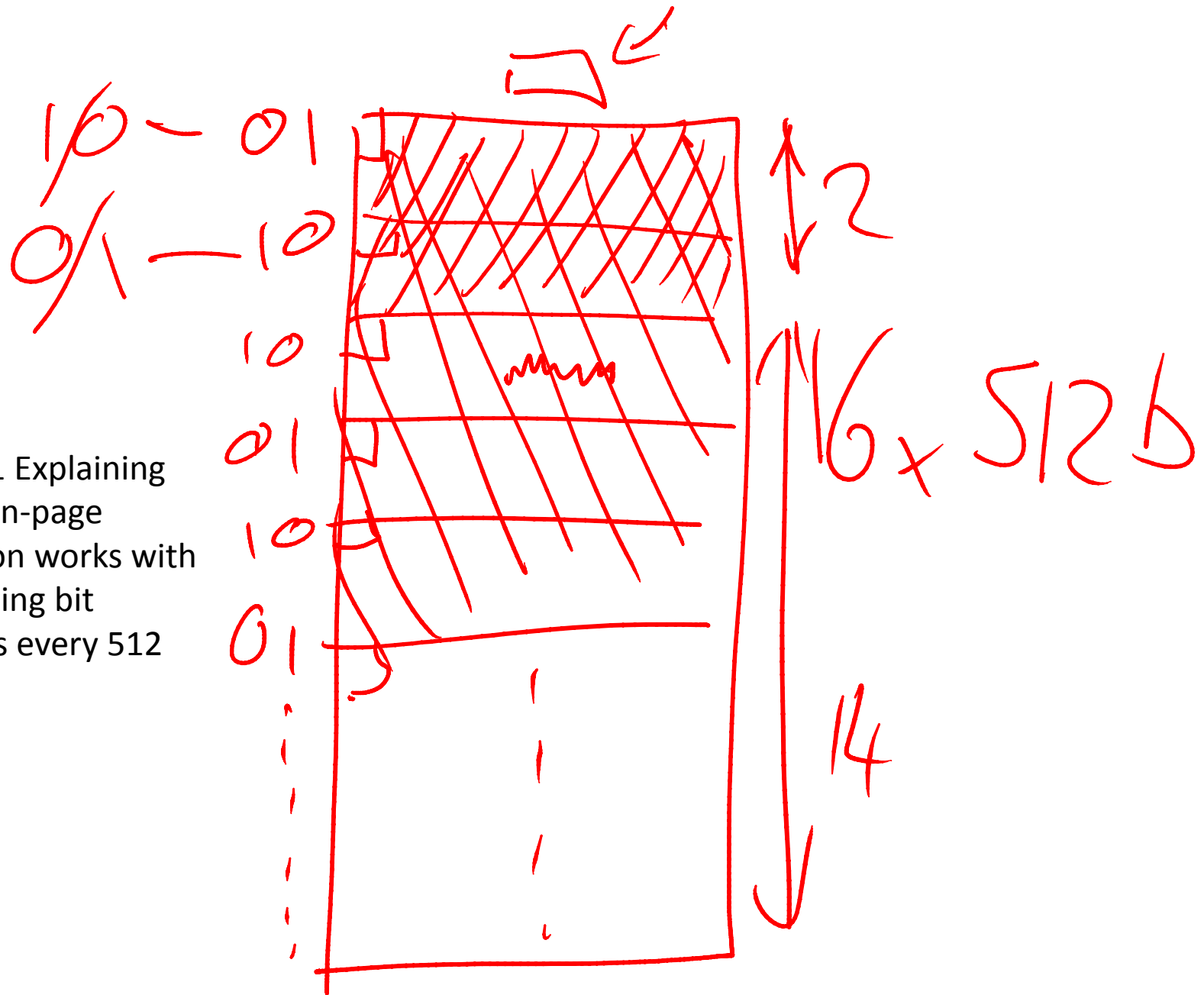


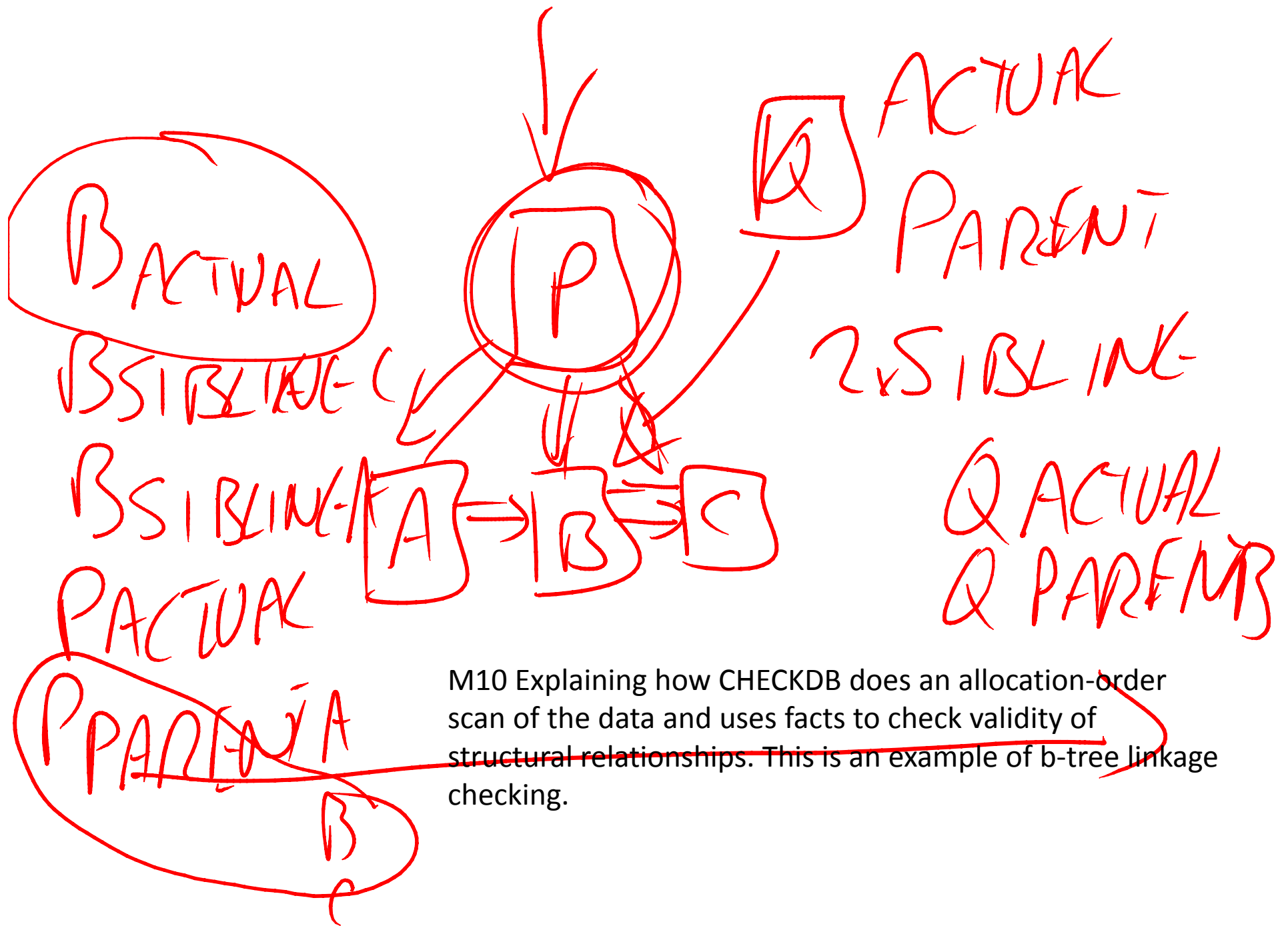
M9S9 Showing that having multiple snapshots causes all pages that change to be synchronously pushed into all existing snapshots that need the page

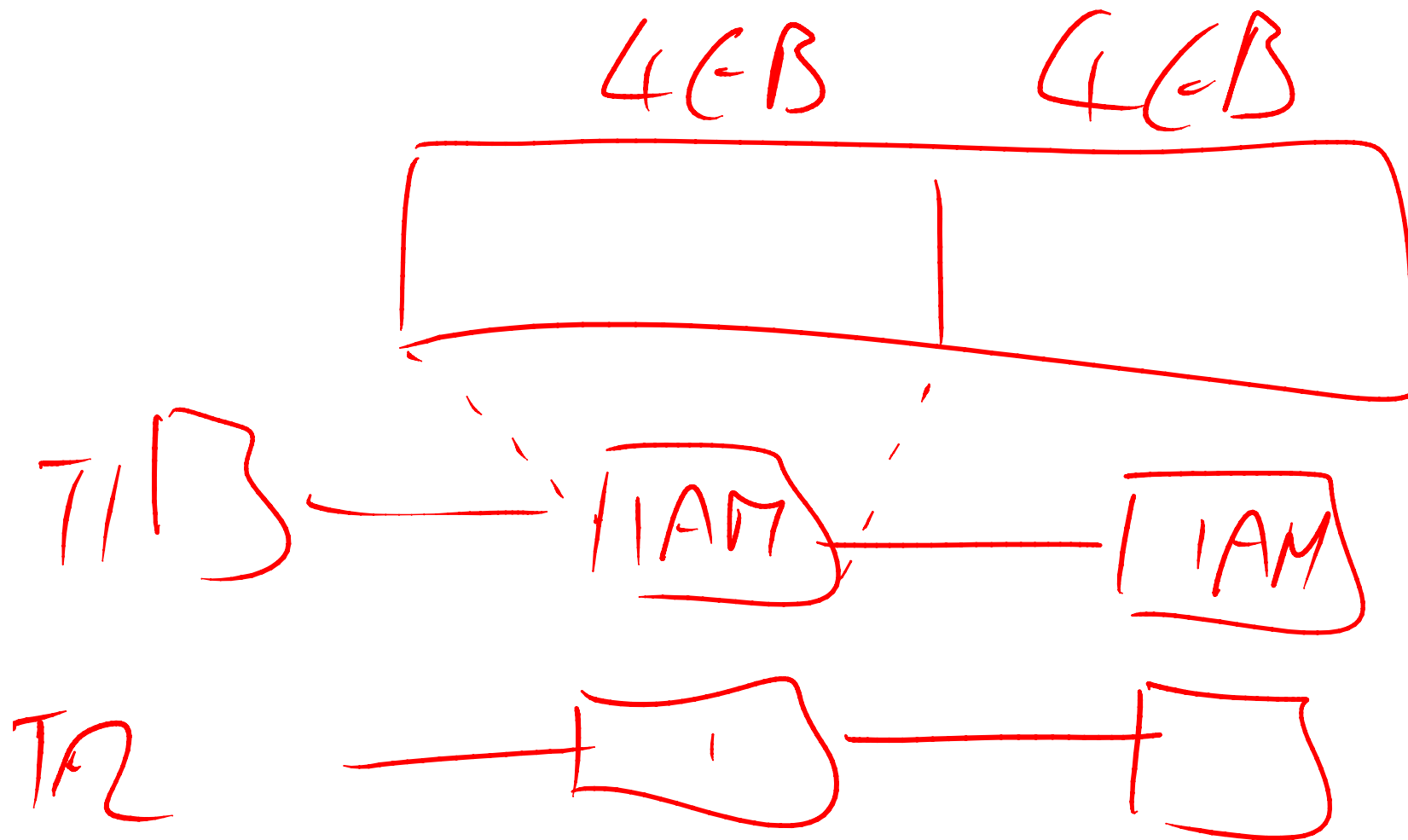


M9 Explaining the data-recovery demo. Pages are deallocated in the source database, then data read from them into the context of the snapshot, to then be inserted into new pages in the source database – overwriting the deallocated pages and causing them to be pushed into the snapshot. A mind-bender!

M10S11 Explaining
how torn-page
detection works with
alternating bit
patterns every 512
bytes







M10 Explaining one of the allocation checks – XORing IAM pages for a GAM interval together to make sure that no two tables have the same data extent allocated.