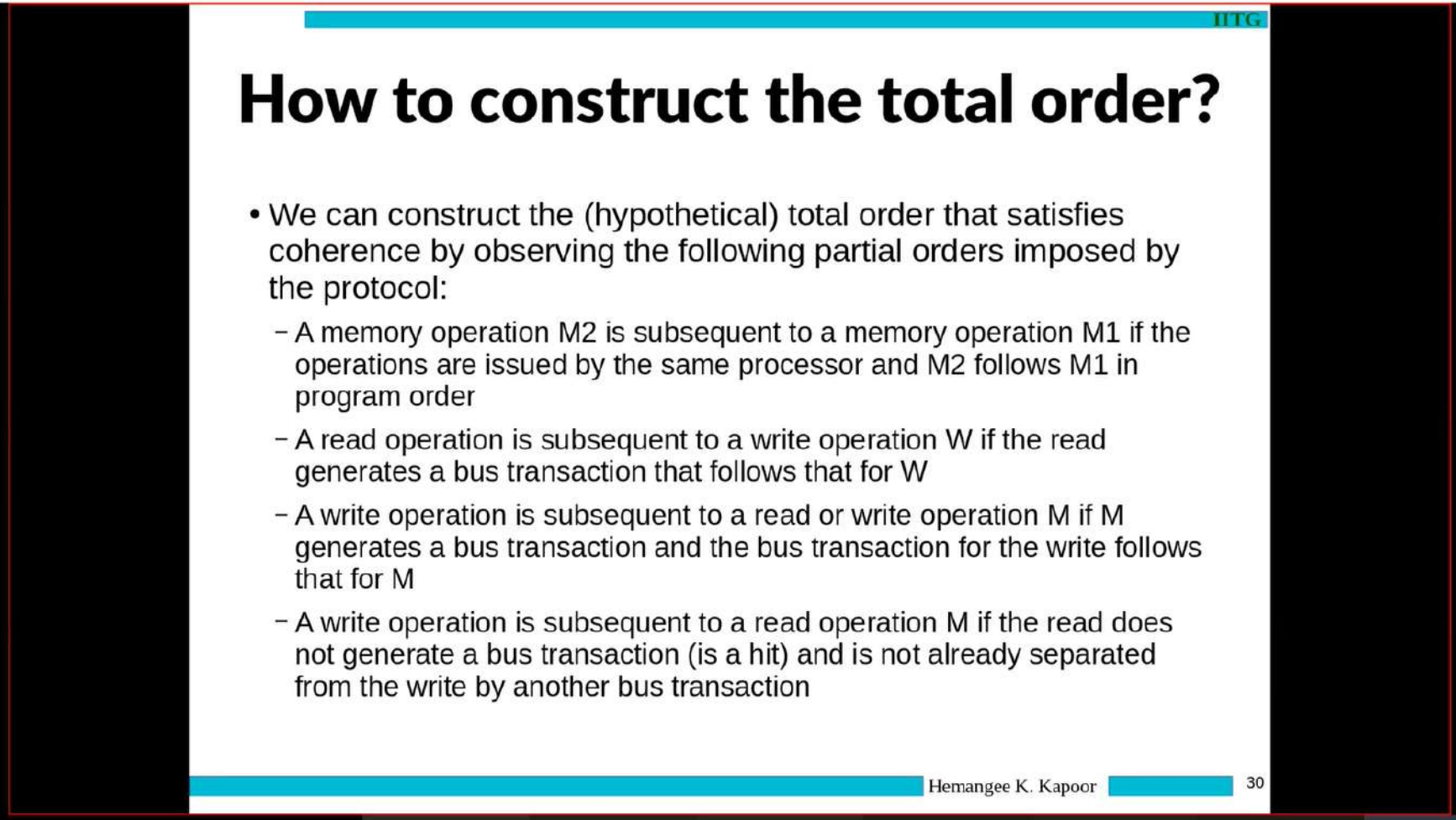
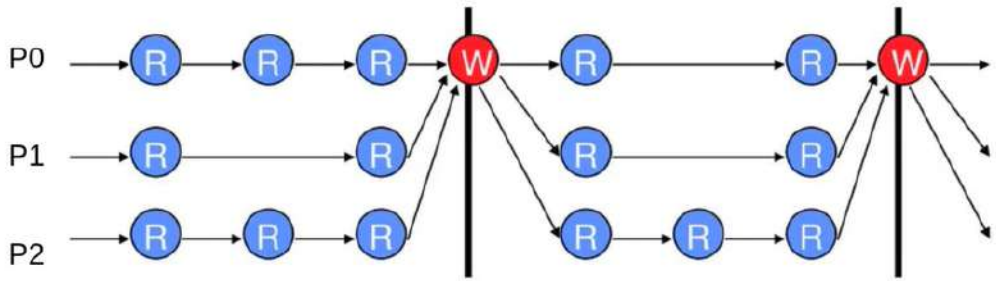


How to construct the total order?

- Any serial order that preserves the resulting partial order is coherent
- The subsequent ordering relationship is transitive
- Several read transaction happening inside the processor which do not appear on the bus are not ordered by the bus
- In fact any interleaving of read operations in these internal segments is a valid serial order, as long as it obeys the program order



Microsoft Teams

Search

Activity

Chat

Teams

Assignments

Calendar

Calls

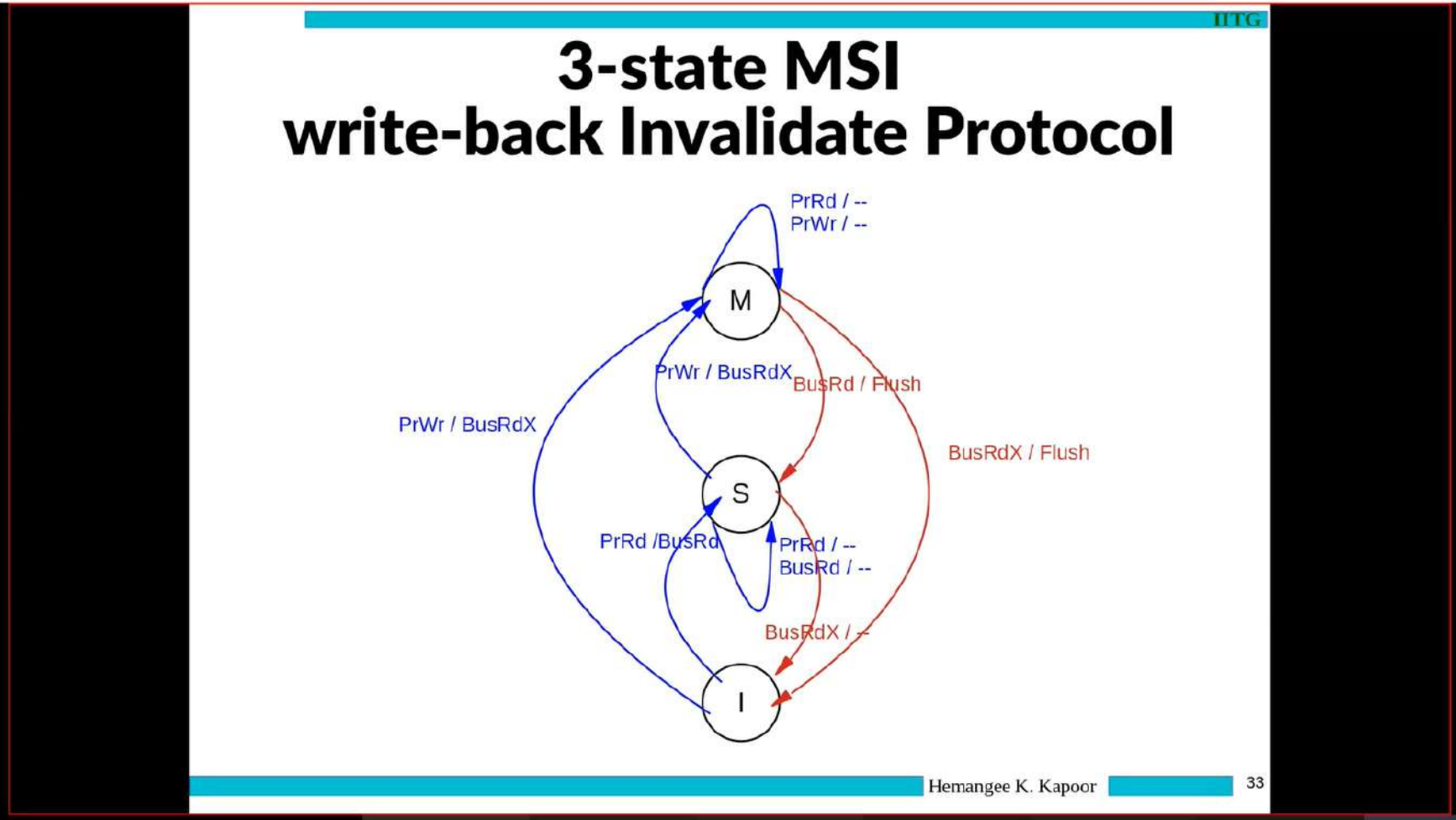
Apps

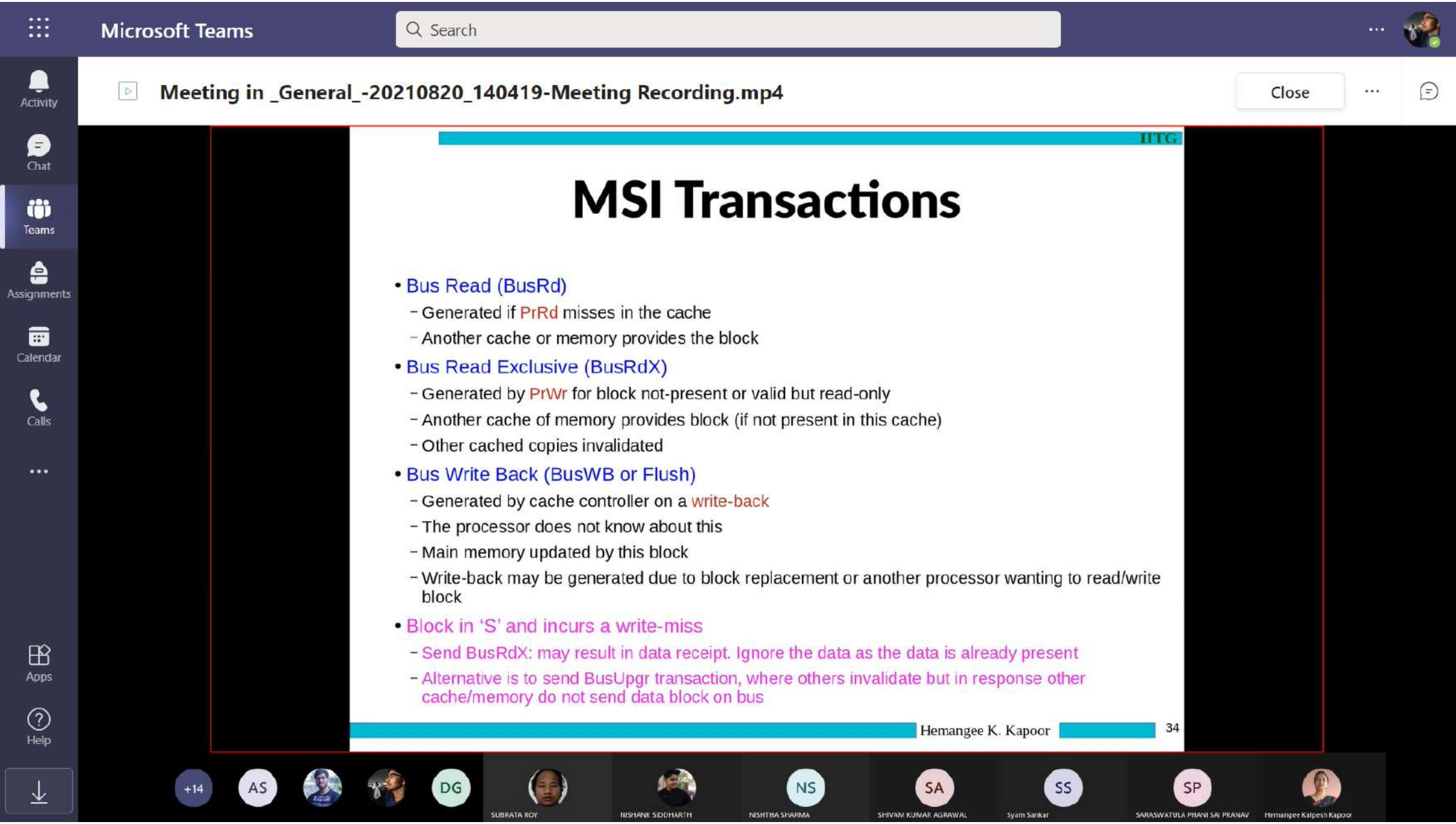
Help

Meeting in _General_-20210820_140419-Meeting Recording.mp4

Close

+14 AS DG SUBRATA ROY NISHANK SIDDHARTH NS NISHTHA SHARMA SA SHIVAM KUMAR AGRAWAL SS Syam Sankar SP SARASWATULA PHANI SAI PRANAV Hemangee Kalpesh Kapoor




Microsoft Teams
Search
Meeting in _General_-20210820_140419-Meeting Recording.mp4
Close
Activity
Chat
Teams
Assignments
Calendar
Calls
Apps
Help
IITG
MSI Transactions
• Bus Read (BusRd)
– Generated if PrRd misses in the cache
– Another cache or memory provides the block
• Bus Read Exclusive (BusRdX)
– Generated by PrWr for block not-present or valid but read-only
– Another cache of memory provides block (if not present in this cache)
– Other cached copies invalidated
• Bus Write Back (BusWB or Flush)
– Generated by cache controller on a write-back
– The processor does not know about this
– Main memory updated by this block
– Write-back may be generated due to block replacement or another processor wanting to read/write block
• Block in 'S' and incurs a write-miss
– Send BusRdX: may result in data receipt. Ignore the data as the data is already present
– Alternative is to send BusUpgr transaction, where others invalidate but in response other cache/memory do not send data block on bus
Hemangee K. Kapoor
34
+14
AS
DG
SUBRATA ROY
NISHANK SIDDHARTH
NS
NISHTHA SHARMA
SA
SHIVAM KUMAR AGRAWAL
SS
Syam Sankar
SP
SARASWATULA PHANI SAI PRANAV
Hemangee Kalpesh Kapoor

Microsoft Teams

Search

Activity

Chat

Teams

Assignments

Calendar

Calls

Apps

Help

Meeting in _General_-20210820_140419-Meeting Recording.mp4

Close

13 August 2021 - Go

20_Aug - Mentimeter

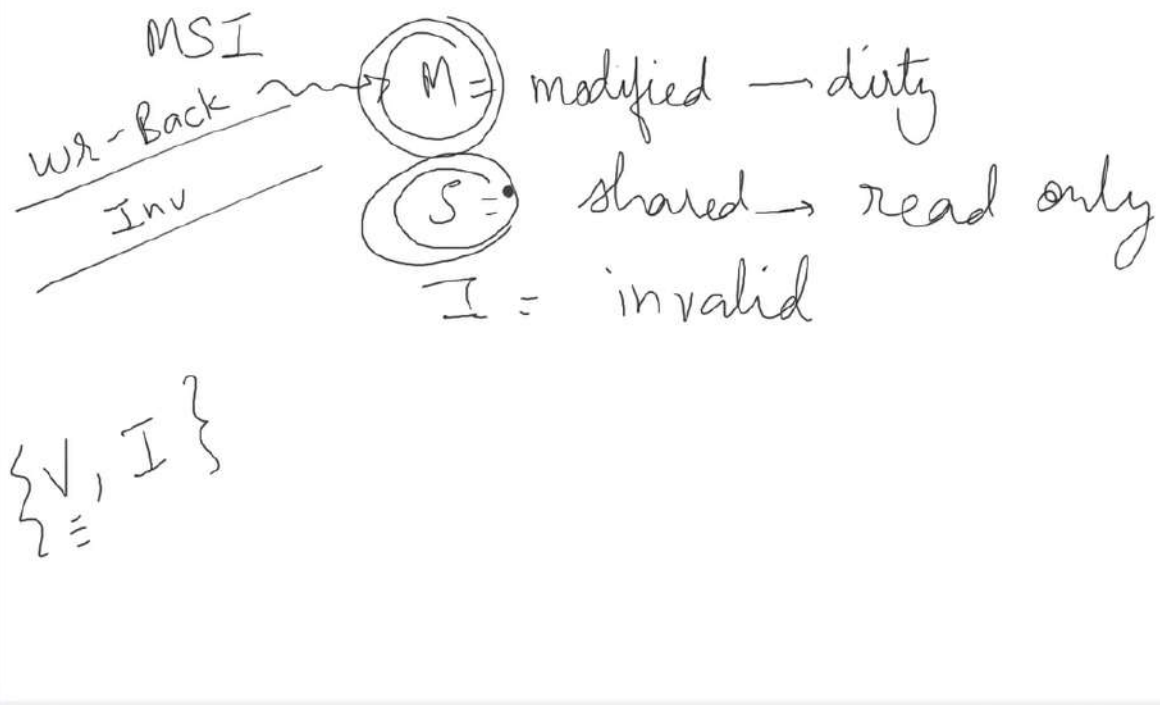
jamboard.google.com/d/17PzgLplDMGMwOzBQkYwX3AP0SrpzSlPyOMgXJjmkxJ8/viewer?f=10

13 August 2021

Share

Set background

Clear frame



85% (0:22) Fri, August 20, 14:14

+14 AS DG SUBRATA ROY NISHANK SIDDHARTH NS NISHTHA SHARMA SA SHIVAM KUMAR AGRAWAL SS Syam Sankar SP SARASWATULA PHANI SAI PRANAV Hemangee Kalpesh Kapoor

Microsoft Teams

Search

Meeting in _General_-20210820_140419-Meeting Recording.mp4

Close

13 August 2021 - Go

20_Aug - Mentimeter

jamboard.google.com/d/17PzgLpIDMGMwOzBQkYwX3AP0SrpzSIPyOMgXIjmkxJ8/viewer?f=10

13 August 2021

11/12

Share

Set background

Clear frame

MSI

Wr-Back

Inv

M = modified → dirty

S = shared → read only

I = invalid

{V, I}

Wr-Miss

inv

BusRd (1)

BusWr

PrWR

PrRd

BusRd X (2)

flush

Wr-Back (3)

87% (0:22) Fri, August 20, 14:18

Activity

Chat

Teams

Assignments

Calendar

Calls

Apps

Help

+27

AS

SUBRATA ROY

DG

DEVANSHI GUPTA

NS

NISHTHA SHARMA

SA

SHIVAM KUMAR AGRAWAL

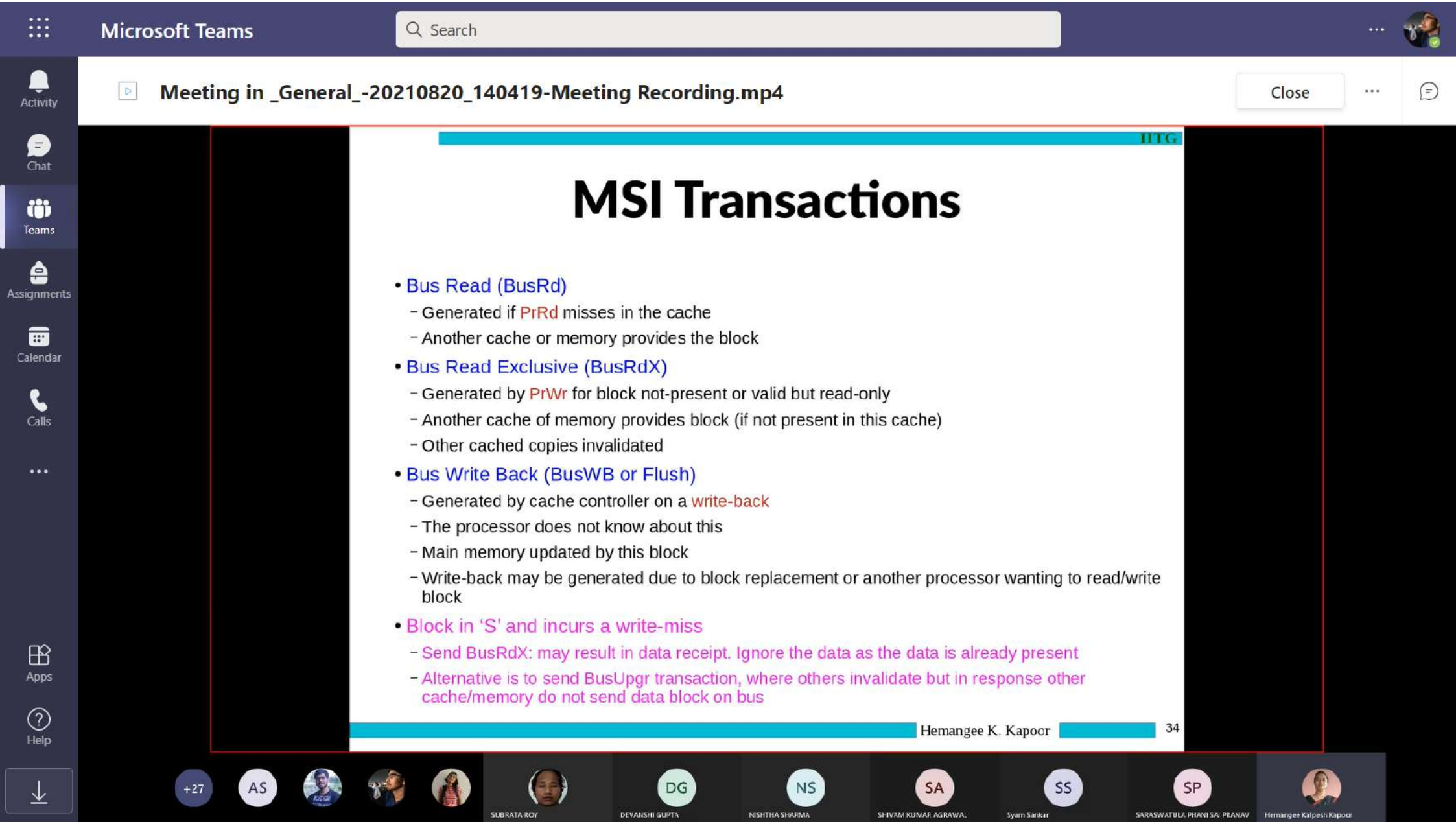
SS

Syam Sankar

SP

SARASWATULA PHANI SAI PRANAV

Hemangee Kalpesh Kapoor


Microsoft Teams
Search
Meeting in _General_-20210820_140419-Meeting Recording.mp4
Close
Activity
Chat
Teams
Assignments
Calendar
Calls
Apps
Help
IITG
MSI Transactions
• Bus Read (BusRd)
– Generated if PrRd misses in the cache
– Another cache or memory provides the block
• Bus Read Exclusive (BusRdX)
– Generated by PrWr for block not-present or valid but read-only
– Another cache of memory provides block (if not present in this cache)
– Other cached copies invalidated
• Bus Write Back (BusWB or Flush)
– Generated by cache controller on a write-back
– The processor does not know about this
– Main memory updated by this block
– Write-back may be generated due to block replacement or another processor wanting to read/write block
• Block in 'S' and incurs a write-miss
– Send BusRdX: may result in data receipt. Ignore the data as the data is already present
– Alternative is to send BusUpgr transaction, where others invalidate but in response other cache/memory do not send data block on bus
Hemangee K. Kapoor
34
+27
AS
SUBRATA ROY
DG
DEVANSHI GUPTA
NS
NISHTHA SHARMA
SA
SHIVAM KUMAR AGRAWAL
SS
Syam Sankar
SP
SARASWATULA PHANI SAI PRANAV
Hemangee Kalpesh Kapoor

Microsoft Teams

Search

Activity
Chat
Teams
Assignments
Calendar
Calls
Apps
Help

Meeting in _General_-20210820_140419-Meeting Recording.mp4

Close

13 August 2021 - Go... | 20_Aug - Mentimeter

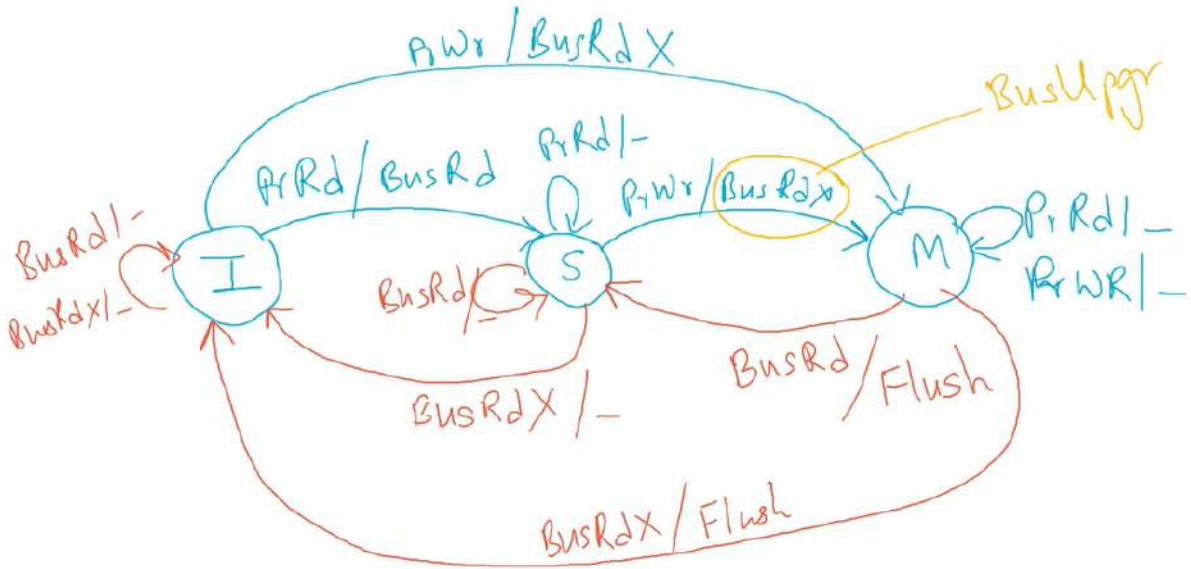
jamboard.google.com/d/17PzgLpiDMGMwOzBQkYwX3AP0SrpzSIPyOMgXIjmkxj8/viewer?f=11

13 August 2021

12 / 12

Share

Set background | Clear frame



93% (0:19) Fri, August 20, 14:31

+26 | DG | YK YOGESH KUMAR | TANVISH | NS NISHTHA SHARMA | SA SHIVAM KUMAR AGRAWAL | SS Syam Sankar | SP SARASWATULA PHANI SAI PRANAV | Hemangee Kalpesh Kapoor

Microsoft Teams

Search

Activity

Chat

Teams

Assignments

Calendar

Calls

Apps

Help

Meeting in _General_-20210820_140419-Meeting Recording.mp4

Close

13 August 2021 - Go... | 20_Aug - Mentimeter

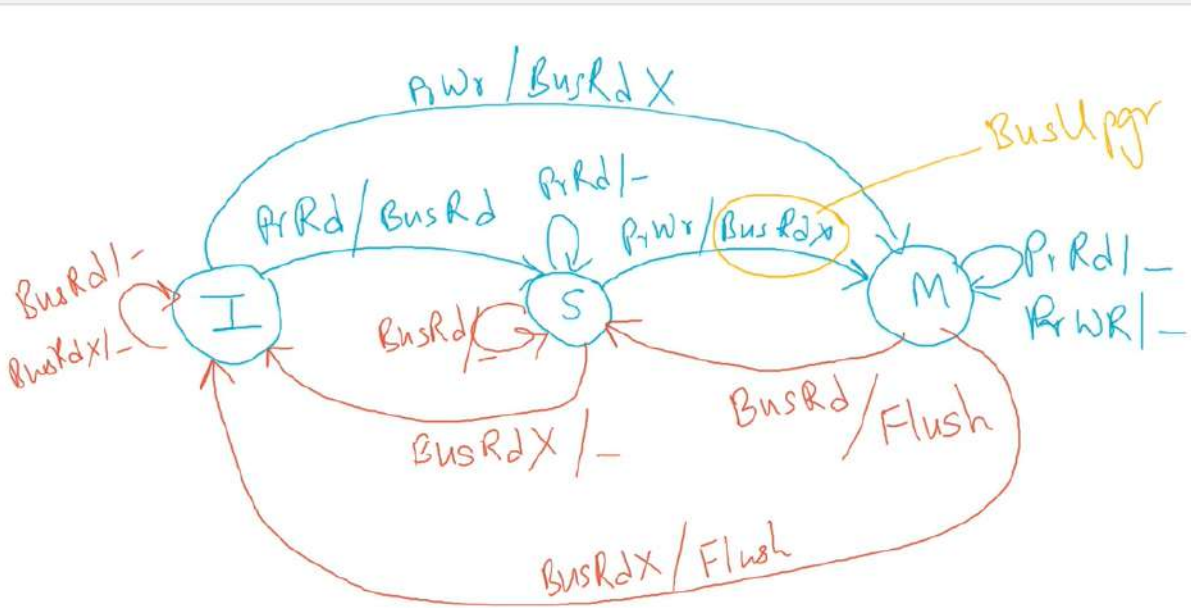
jamboard.google.com/d/17PzgLpIDMGMwOzBQkYwX3AP0SrpzSIPyOMgXIjmkxJ8/viewer?f=11

13 August 2021

12 / 12

Share

Set background | Clear frame

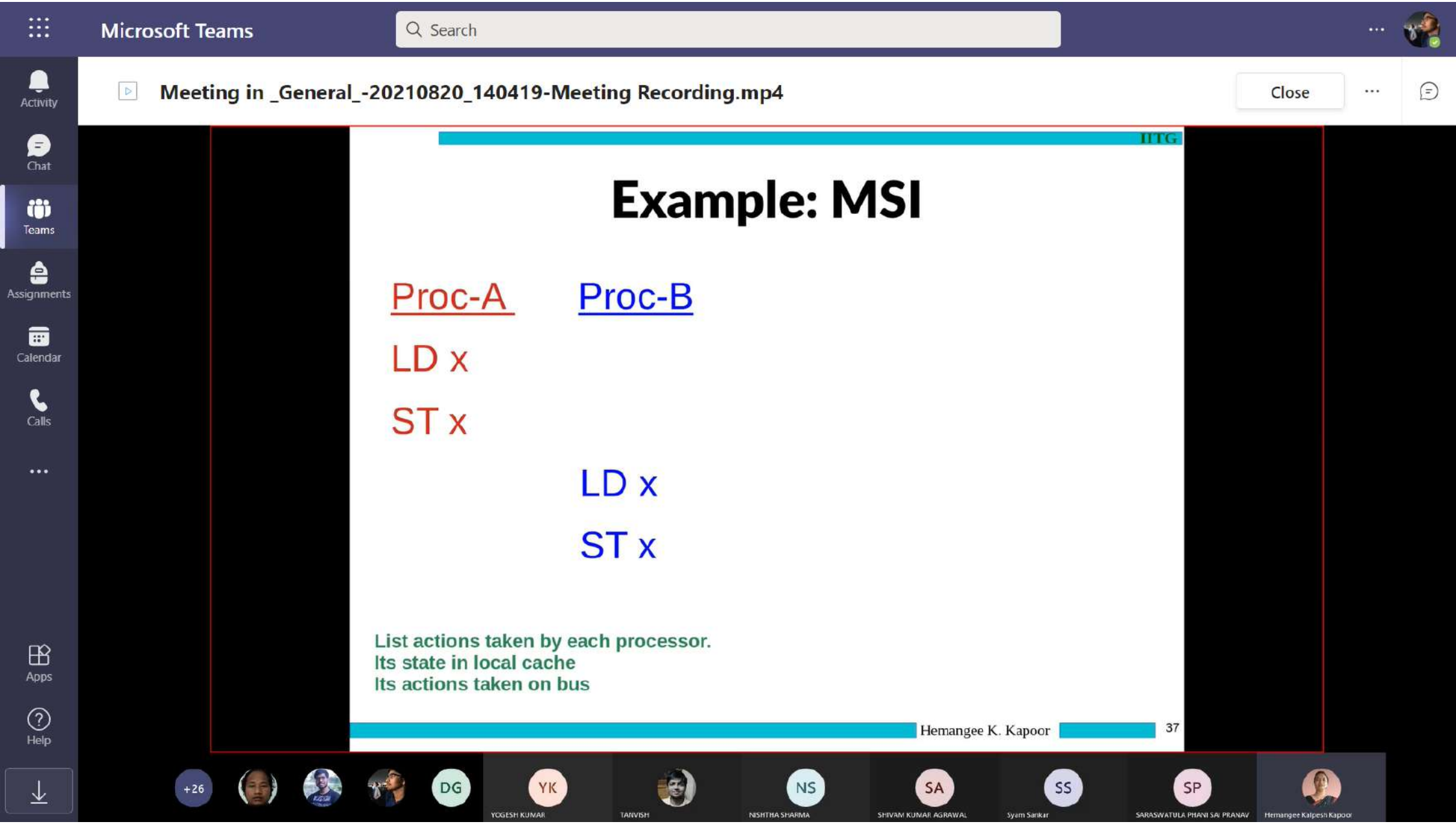


93% (0:19) Fri, August 20, 14:31

+26 | DG | YK YOGESH KUMAR | TANVISH | NS NISHTHA SHARMA | SA SHIVAM KUMAR AGRAWAL | SS Syam Sankar | SP SARASWATULA PHANI SAI PRANAV | Hemangee Kalpesh Kapoor

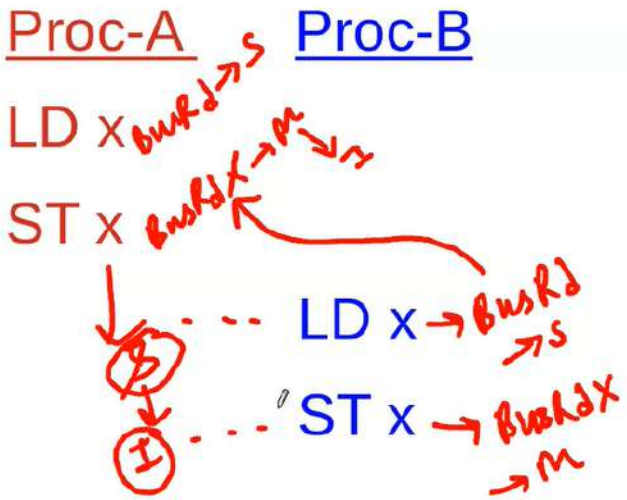
MSI state transition table

State	This Processor		Other Processor	
	LOAD	STORE	LOAD	STORE
I Invalid	Miss → S	Miss → M	-	-
S Shared	Hit	Upgrade-miss → M	-	→ I
M Modified	Hit	Hit	Send Data → S	Send Data → I


Microsoft Teams
Search
Meeting in _General_-20210820_140419-Meeting Recording.mp4
Close
Activity
Chat
Teams
Assignments
Calendar
Calls
Apps
Help
IITG
Example: MSI
Proc-A
Proc-B
LD x
ST x
LD x
ST x
List actions taken by each processor.
Its state in local cache
Its actions taken on bus
Hemangee K. Kapoor
37
+26
DG
YK
YOGESH KUMAR
TANVISH
NS
NISHTHA SHARMA
SA
SHIVAM KUMAR AGRAWAL
SS
Syam Sankar
SP
SARASWATULA PHANI SAI PRANAV
Hemangee Kalpesh Kapoor

IITG

Example: MSI



- LD by Proc-B

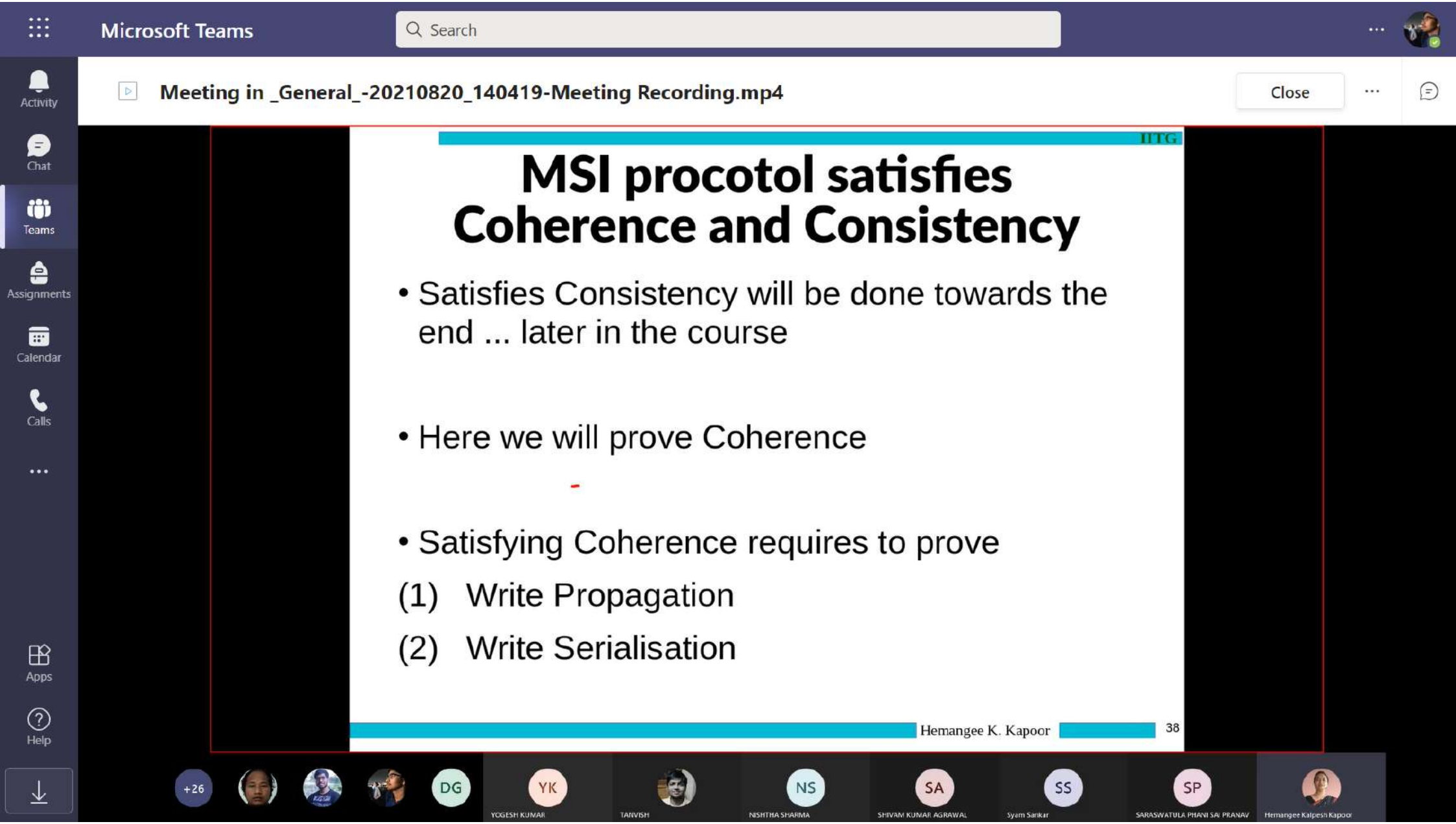
  generates BusRd,

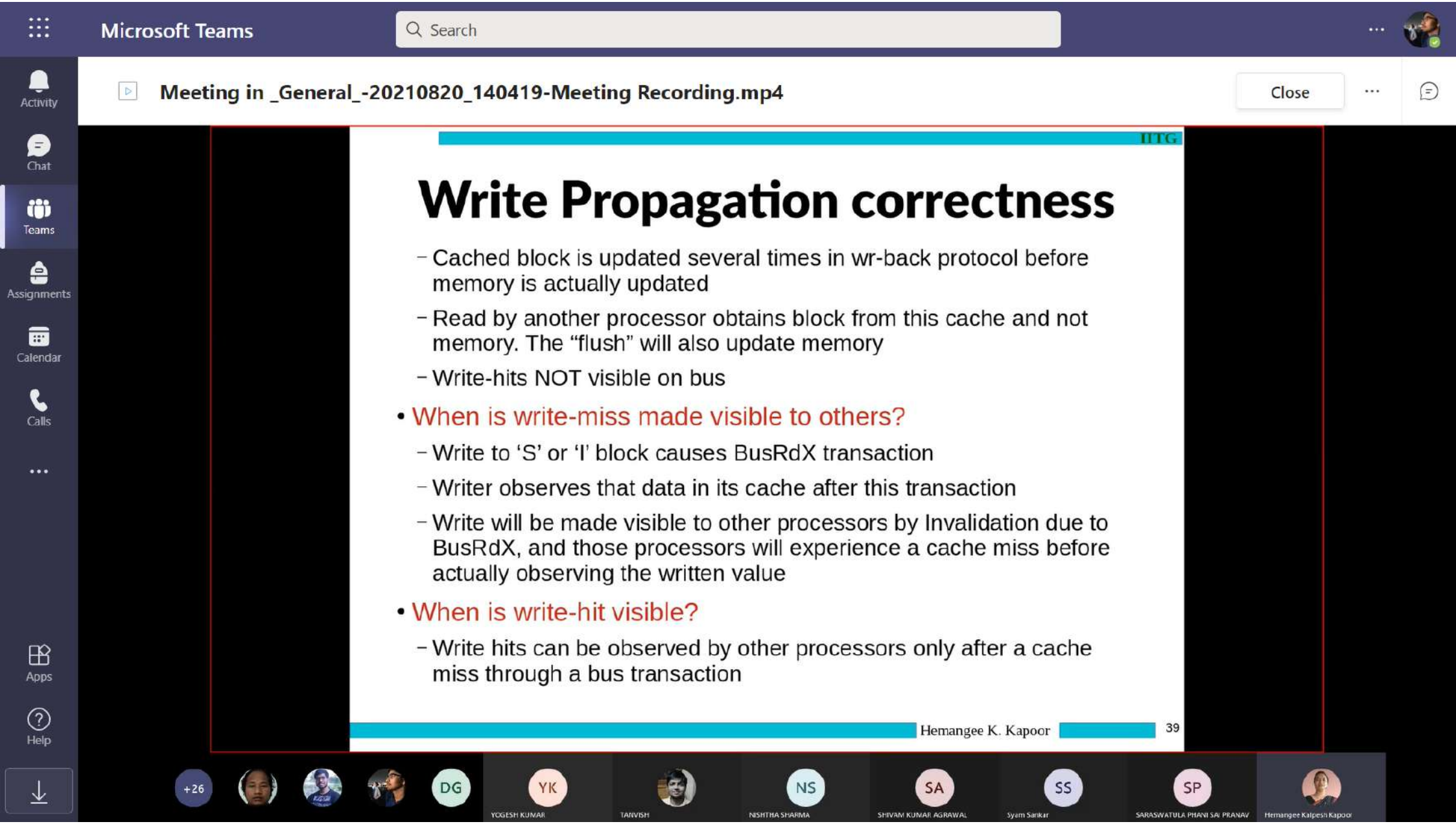
  Proc-A sends copy of A and becomes 'S'
- ST by Proc-B

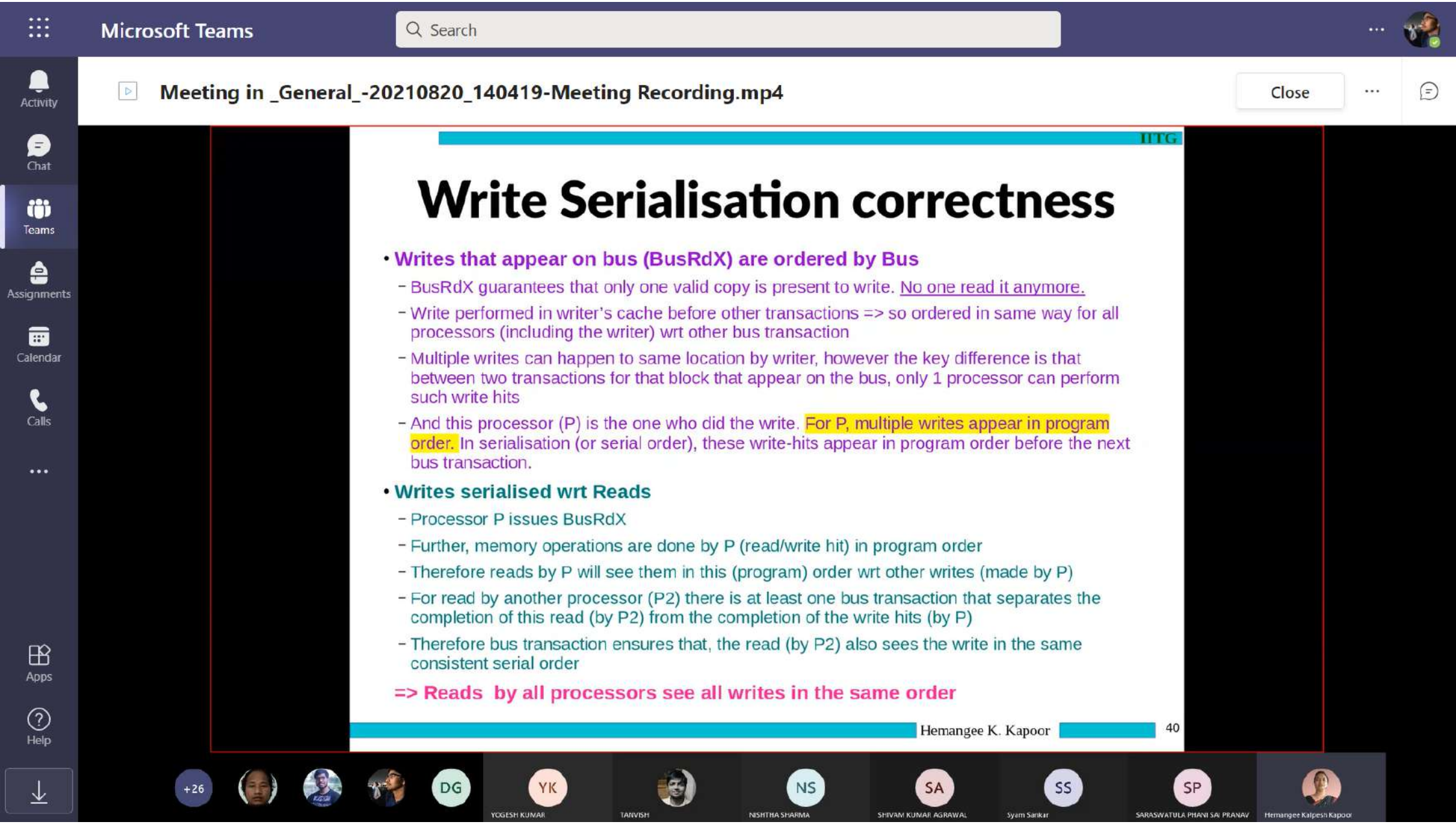
  generates BusRdX,

  Proc-A transitions to 'I'

List actions taken by each processor.
Its state in local cache
Its actions taken on bus


Microsoft Teams
Search
Meeting in _General_-20210820_140419-Meeting Recording.mp4
Close
Activity
Chat
Teams
Assignments
Calendar
Calls
Apps
Help
IITG
MSI procotol satisfies Coherence and Consistency
• Satisfies Consistency will be done towards the end ... later in the course
• Here we will prove Coherence
• Satisfying Coherence requires to prove
(1) Write Propagation
(2) Write Serialisation
Hemangee K. Kapoor
38
+26
DG
YK
YOGESH KUMAR
TANVISH
NS
NISHTHA SHARMA
SA
SHIVAM KUMAR AGRAWAL
SS
Syam Sankar
SP
SARASWATULA PHANI SAI PRANAV
Hemangee Kalpesh Kapoor


Microsoft Teams
Search
Meeting in _General_-20210820_140419-Meeting Recording.mp4
Close
Activity
Chat
Teams
Assignments
Calendar
Calls
Apps
Help
IITG
Write Propagation correctness
– Cached block is updated several times in wr-back protocol before memory is actually updated
– Read by another processor obtains block from this cache and not memory. The "flush" will also update memory
– Write-hits NOT visible on bus
• When is write-miss made visible to others?
– Write to 'S' or 'I' block causes BusRdX transaction
– Writer observes that data in its cache after this transaction
– Write will be made visible to other processors by Invalidation due to BusRdX, and those processors will experience a cache miss before actually observing the written value
• When is write-hit visible?
– Write hits can be observed by other processors only after a cache miss through a bus transaction
Hemangee K. Kapoor
39
+26
DG
YK
YOGESH KUMAR
TANVISH
NS
NISHTHA SHARMA
SA
SHIVAM KUMAR AGRAWAL
SS
Syam Sankar
SP
SARASWATULA PHANI SAI PRANAV
Hemangee Kalpesh Kapoor


Microsoft Teams
Search
Meeting in _General_-20210820_140419-Meeting Recording.mp4
Close
Activity
Chat
Teams
Assignments
Calendar
Calls
Apps
Help
IITG
Write Serialisation correctness
• Writes that appear on bus (BusRdX) are ordered by Bus
– BusRdX guarantees that only one valid copy is present to write. No one read it anymore.
– Write performed in writer's cache before other transactions => so ordered in same way for all processors (including the writer) wrt other bus transaction
– Multiple writes can happen to same location by writer, however the key difference is that between two transactions for that block that appear on the bus, only 1 processor can perform such write hits
– And this processor (P) is the one who did the write. For P, multiple writes appear in program order. In serialisation (or serial order), these write-hits appear in program order before the next bus transaction.
• Writes serialised wrt Reads
– Processor P issues BusRdX
– Further, memory operations are done by P (read/write hit) in program order
– Therefore reads by P will see them in this (program) order wrt other writes (made by P)
– For read by another processor (P2) there is at least one bus transaction that separates the completion of this read (by P2) from the completion of the write hits (by P)
– Therefore bus transaction ensures that, the read (by P2) also sees the write in the same consistent serial order
=> Reads by all processors see all writes in the same order
Hemangee K. Kapoor
40
+26
DG
YK
YOGESH KUMAR
TANVISH
NS
NISHTHA SHARMA
SA
SHIVAM KUMAR AGRAWAL
SS
Syam Sankar
SP
SARASWATULA PHANI SAI PRANAV
Hemangee Kalpesh Kapoor

Microsoft Teams

Search

Activity

Chat

Teams

Assignments

Calendar

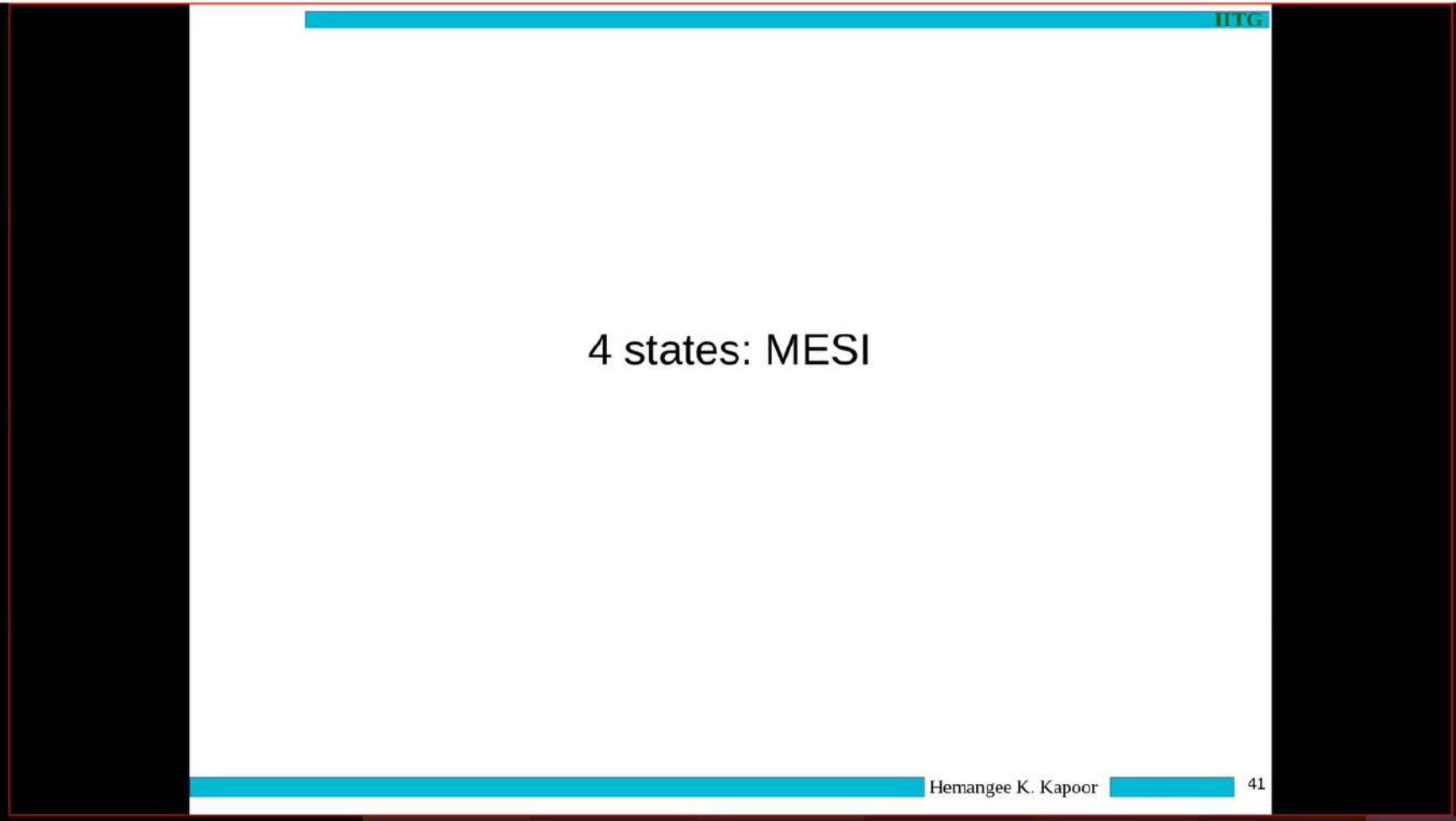
Calls

Apps

Help

Meeting in _General_-20210820_140419-Meeting Recording.mp4

Close



+26

DG

YK YOGESH KUMAR

TANVISH

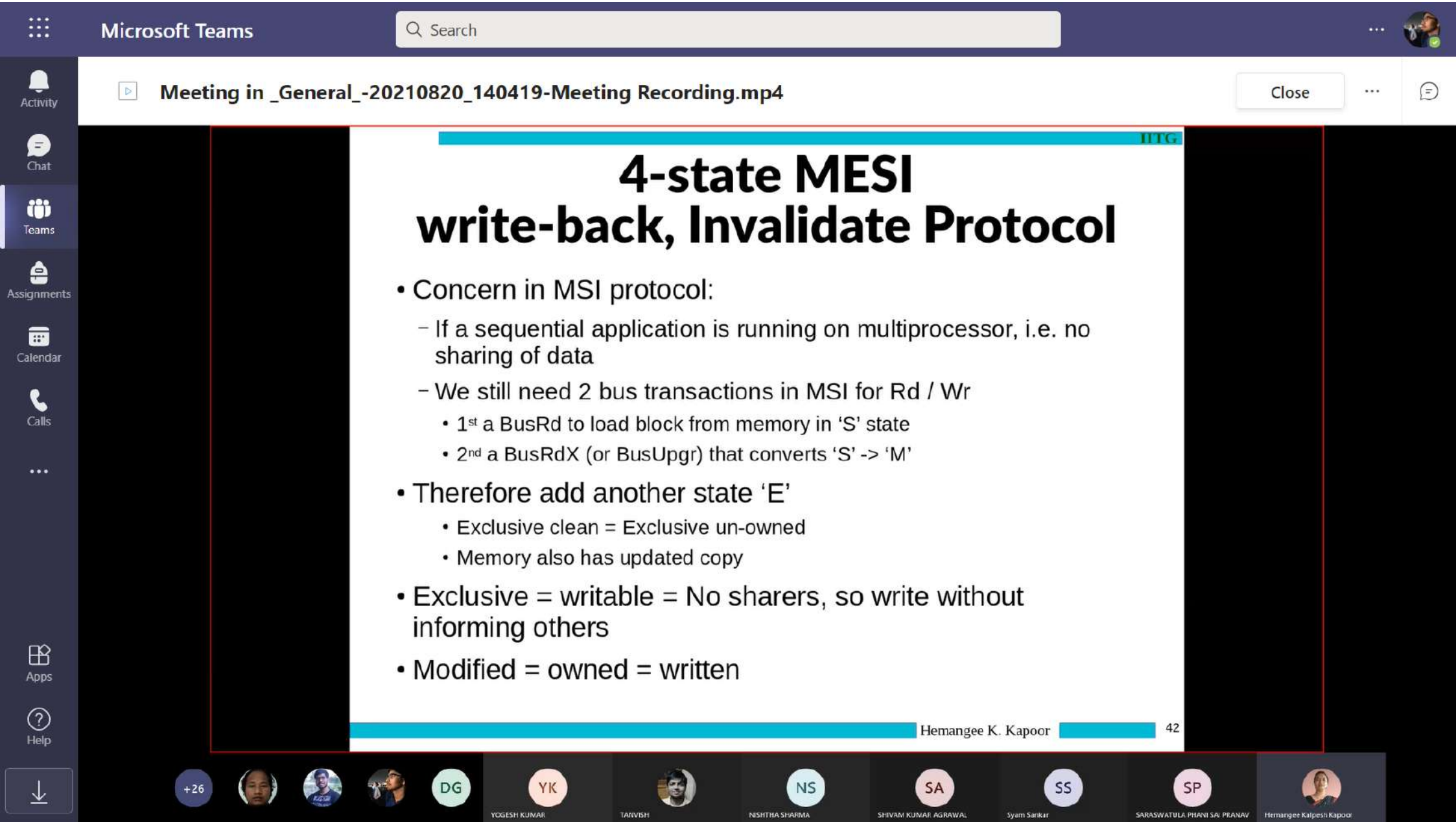
NS NISHTHA SHARMA

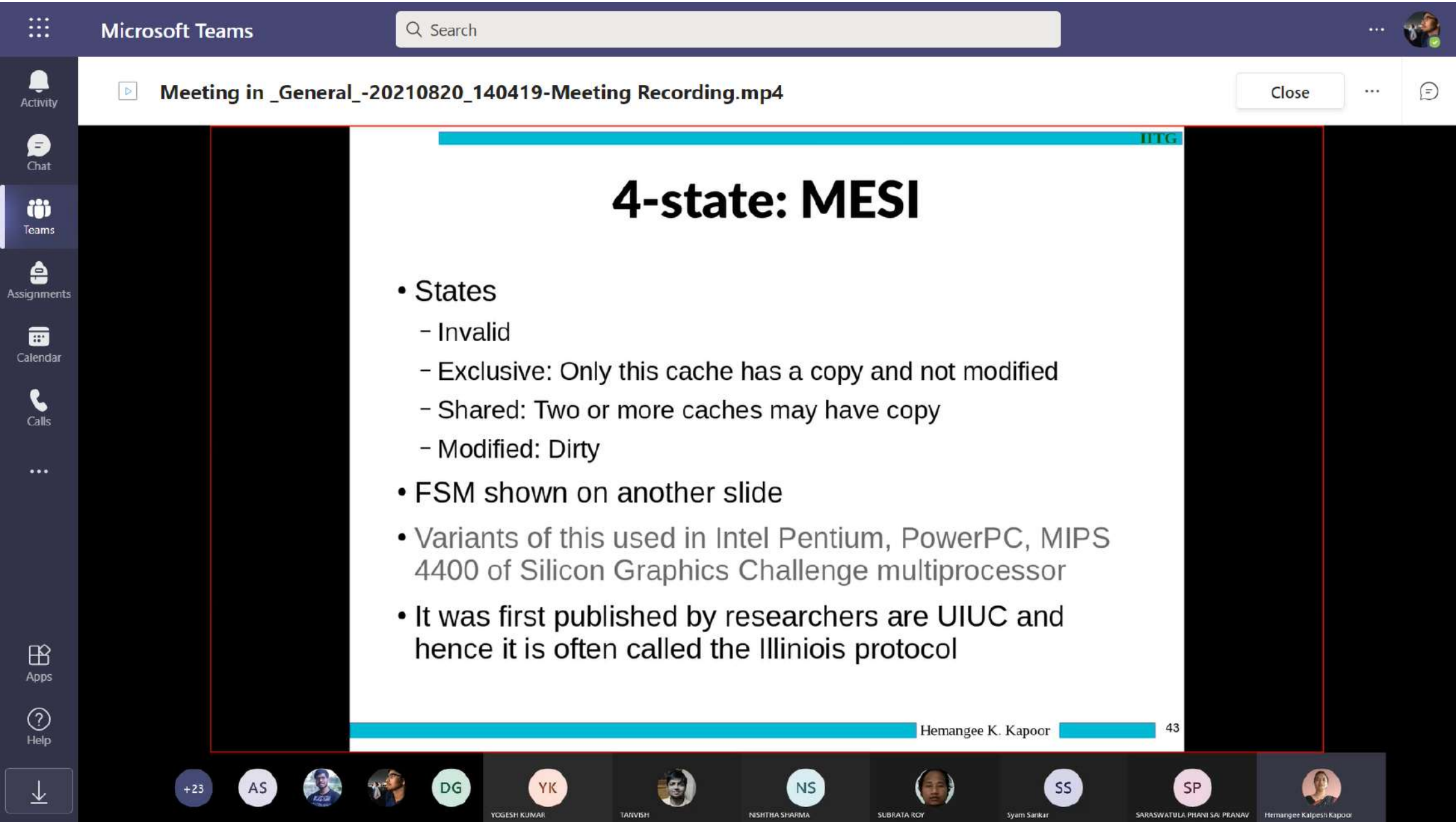
SA SHIVAM KUMAR AGRAWAL

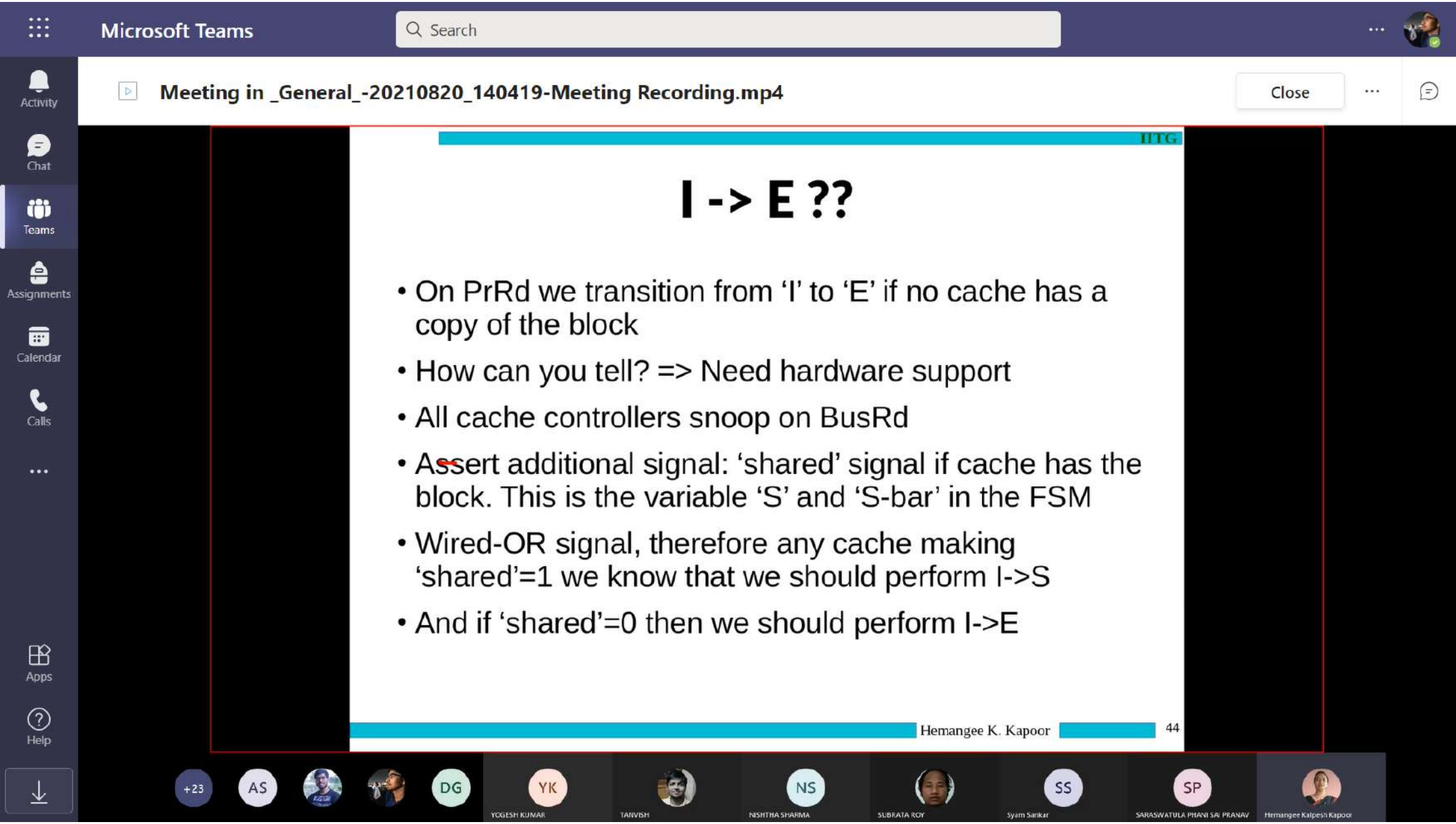
SS Syam Sankar

SP SARASWATULA PHANI SAI PRANAV

Hemangee Kalpesh Kapoor

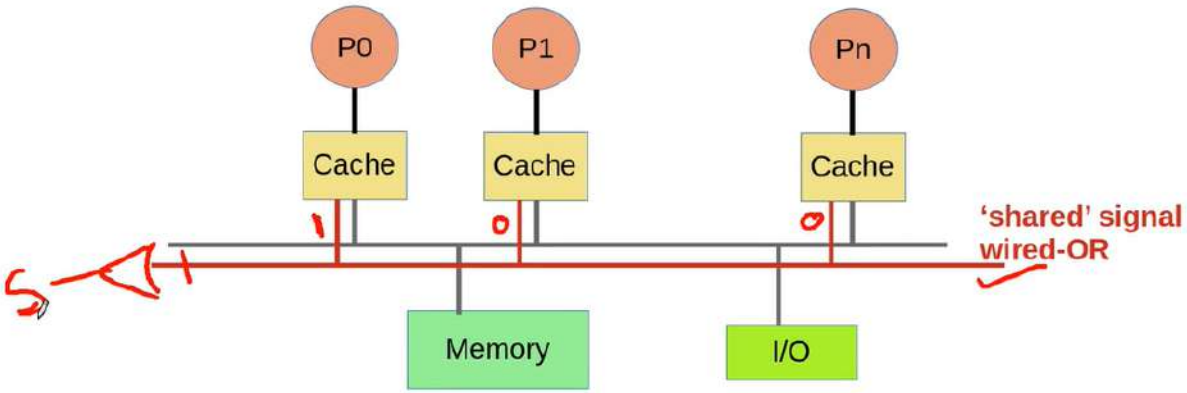

Microsoft Teams
Search
Meeting in _General_-20210820_140419-Meeting Recording.mp4
Close
Activity
Chat
Teams
Assignments
Calendar
Calls
Apps
Help
IITG
4-state MESI
write-back, Invalidate Protocol
• Concern in MSI protocol:
– If a sequential application is running on multiprocessor, i.e. no sharing of data
– We still need 2 bus transactions in MSI for Rd / Wr
• 1st a BusRd to load block from memory in 'S' state
• 2nd a BusRdX (or BusUpgr) that converts 'S' -> 'M'
• Therefore add another state 'E'
• Exclusive clean = Exclusive un-owned
• Memory also has updated copy
• Exclusive = writable = No sharers, so write without informing others
• Modified = owned = written
Hemangee K. Kapoor
42
+26
DG
YK
YOGESH KUMAR
TANVISH
NS
NISHTHA SHARMA
SA
SHIVAM KUMAR AGRAWAL
SS
Syam Sankar
SP
SARASWATULA PHANI SAI PRANAV
Hemangee Kalpesh Kapoor


Microsoft Teams
Search
Meeting in _General_-20210820_140419-Meeting Recording.mp4
Close
Activity
Chat
Teams
Assignments
Calendar
Calls
Apps
Help
IITG
4-state: MESI
• States
– Invalid
– Exclusive: Only this cache has a copy and not modified
– Shared: Two or more caches may have copy
– Modified: Dirty
• FSM shown on another slide
• Variants of this used in Intel Pentium, PowerPC, MIPS 4400 of Silicon Graphics Challenge multiprocessor
• It was first published by researchers are UIUC and hence it is often called the Illiniois protocol
Hemangee K. Kapoor
43
+23
AS
DG
YK
YOGESH KUMAR
TANVISH
NS
NISHTHA SHARMA
SUBRATA ROY
SS
Syam Sankar
SP
SARASWATULA PHANI SAI PRANAV
Hemangee Kalpesh Kapoor


Microsoft Teams
Search
Meeting in _General_-20210820_140419-Meeting Recording.mp4
Close
Activity
Chat
Teams
Assignments
Calendar
Calls
Apps
Help
IITG
I -> E ??
• On PrRd we transition from 'I' to 'E' if no cache has a copy of the block
• How can you tell? => Need hardware support
• All cache controllers snoop on BusRd
• Assert additional signal: 'shared' signal if cache has the block. This is the variable 'S' and 'S-bar' in the FSM
• Wired-OR signal, therefore any cache making 'shared'=1 we know that we should perform I->S
• And if 'shared'=0 then we should perform I->E
Hemangee K. Kapoor
44
+23
AS
DG
YK
YOGESH KUMAR
TANVISH
NS
NISHTHA SHARMA
SUBRATA ROY
SS
Syam Sankar
SP
SARASWATULA PHANI SAI PRANAV
Hemangee Kalpesh Kapoor

IITG

I -> E ??



- BusRd(**S**) means shared line is asserted on BusRd transaction
- BusRd(**S-bar**) means 'S' is un-asserted
- **Flush'** : if cache-to-cache transfers: only one cache flushes data, giving it to requestor
- Satisfying Coherence: Arguments same as MSI
- **MOESI** protocol: Owned state = Exclusive but memory not valid
  - M->S : But one cache has updated copy
  - Memory not updated. Ownership transferred. Cache in state 'O' provides the data
- It is possible to have a block in state 'S' even when there are not other sharers
  - Why? Because, others removed their copies and deletion does not trigger transaction. This cache may not know it is the only one left with a copy