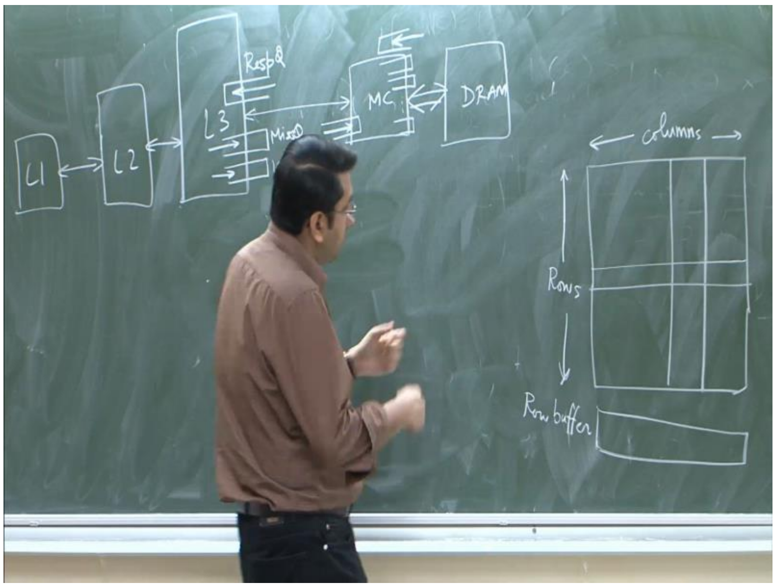
Computer Architecture

Prof. Mainak Chaudhuri
Department of Computer Science and Engineering
Indian Institute of Technology, Kanpur

Lecture - 29
Topics in memory system, DRAM and SRAM Technology

(Refer Slide Time: 00:28)



So, last time we talked about the general system memory architecture that is we have hierarchy of caches inside the processor. So, you state that the can 3 cache controller has at least 3 queues 2 of them are going towards the memory controller 1 is ah miss q another 1 is the write back q. So, this 1 carries the request that missing at the cache and this 1 carries the request this 1 carries the dirty chemical cache box to be return back to main memory.

There is the queue coming in this direction where you get the responses to the misses. So, I will call it a response queue and there is a avoider which schedules these 3 queues according to some protocol and whenever it fix a 1 of these 2 queues. It would pick up the another queue and put it in the memory controller queue all right. And memory controller job is to schedule this queue to pick up request decode them many that you can look at the address and look at the command.

Then depending on the address to figure out the bank number. So, we will today see how how decode the bank id from address and then a bunch of queues 1 1 for each bank of the D RAM that will put corresponding request into the corresponding queue. And the bank controller will schedule this request according to some protocol. And to send the request to the D RAM and eventually the response will come back into this queue. And the response will resend over to the bus into the entry caches response all right. If it is a write back request then of course, there would not be any response the the data will just be return back to the corresponding bank.

So, is this overview clear to everybody this particular objection all right? So, today what I am going to do is assuming this general architecture we are going to open this up and see what goes inside. So, to start with at a very high level the D RAM is organized as rows and columns of bits. So, there are bunch of rows. So, as we mentioned last time the D RAM will have several banks inside all right.

So, we are looking at 1 bank all right we just opening a 1 bank. So, it will have bunch of rows and and bunch of columns now the why the D RAM organization define today that column is not a column of bits it usually multiple bits. So, for example, these 2 linear typical column it contains several bits all right. So, so this is intersection of row and a column all right column is not a single bit remember which is several bits in the column

Now, how do we really read a particular a bank. So, whenever a request goes as I said the memory controller figure out the bank id and put the request in the corresponding bank request queue. The first command that will go from the bank request queue to the D RAM bank is a it is called a row address store or the ras operation. So, what it do it essentially activates the corresponding row with the data column where the data forms. So, again the row number can be decoded from the given address. So, we talk about that today also.

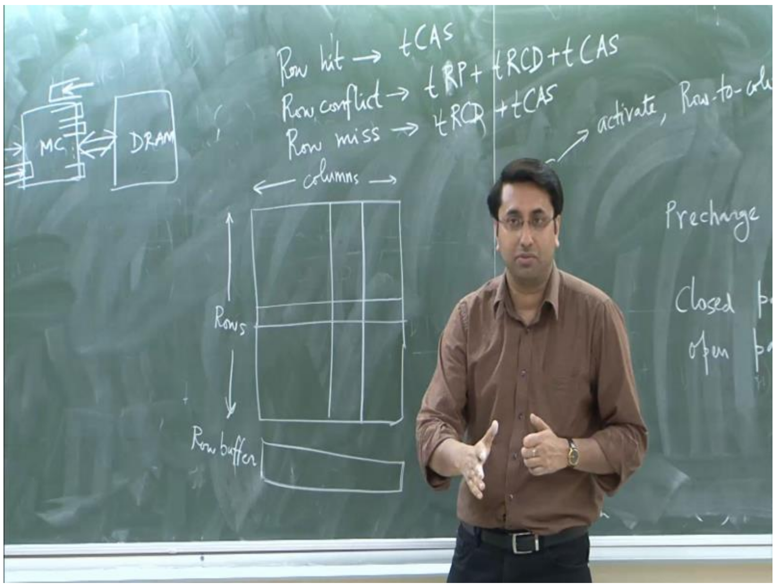
So, essentially what you do is in the when you activate a row you readout this entire row into a row buffer each bank gets a row buffer all right. So, you read this entire row into a row buffer. So, and then the next operation that happens is called a column address row where you send the column address that tells you which column would contain the request data.

But, essentially the job of the D RAM is that take the column address and take out those bits out of the corresponding row and now within this particular column you may not actually require all the bits. So, there is a column offset also that that is decoded from the address. So, certain bits may be needed from this column offset and finally, those bits go out from this particular bank all right. So, that is how you need the D RAM.

Now, 1 thing to observe here is that the content of the row buffer survives until you get a request to a different row which means, if this bank request scheduler is smart enough you would actually cluster all the request into the same row. Together it will send 1 after another that would save your ras operation you can actually do a ras followed by a sequence of caches satisfying all the request all right. So, that is that is a very common scheduling techniques used in all D RAM. Today where the bank bank request scheduler [paratizes] the request that quote the same row of the bank all right.

Now, what happens when you get a … So, eventually of course, it will happen that you do not have any request if the bank request queue that fall on the same row has the currently open row. So, this what open currently open row all right. So, then what you do of course, you have to you have to you know take the extra latency.

(Refer Slide Time: 07:07)



So, now there are 3 things that happen that call that is called a precharge operation. So, ras was essentially missed it is an activate activate operation and then you wait for certain number of cycles before you can send the CAS. that is called the row to column access delay row to column access delay all right.

So, first thing you do is you. So, we are talking about case where currently open row is x and the bank controller has schedule the request to row y all right. So, clearly your row buffer does not contain the desired data and closes this particular all right. Then you do the same thing you activate row y the new row which reach the row out into the row buffer. And then, you wait for this amount of time row to column access delay and then usually the column access row all right.

So, the first 1 that is a particular request mapping to the current open row it is called a row hit which has the latency equal to t CAS just try to do a column access row all right. Your second scenario where a currently requested row is not open is called a row conflict and it has a latency equal to t precharge. Sometimes it is called TRP that is row precharge plus TRCD that is row to column access delay plus t CAS that is a row conflict.

So, you can now clearly see that row conflict time will be much higher than your row hit time. So, naturally a small scheduling technique would be to want this particular bank request queue and cluster all the request going to the same row id all right same thing 1 after another all right. So, then for the whole cluster you will do this only ones and you will do t CAS for the subsequent, there is no separate time in the concept for activating.

No there is no separate time activate is just a command; however, the row buffer is not stable until these much of time we have to wait for these much of time. Is that clear to everybody now? There is the third thing that is called quietly wondering quietly the row miss and row hit. So, the third thing which is called row miss. So, this slightly different from row conflict.

So, in case of row miss what happens is that there is no open row kind of thing all right. So, essentially what you do is you do not do any precharge all you do is you do an activate and the CAS. Now, we might wonder why should this at all happen ever because I always have to do right which corresponds to a last last executed request at the starting only right.

However, there are certain D RAM controller which follow something called a close page policy which basically says that, if the D RAM controller can inform that this row will not be needed in future we can actually close it earlier. So, that you can hide this particular latency where the next request shows up. So, precharge time is not in the critical path anymore.

Precharge

For for precharge time

Which 1? No.

What I am saying is when you read data into row buffer.

No.

We need not to precharge.

No you activate the row which means you read the row out of. So, so that is called the closed page policy. So, in extreme of closed page policy would be that after every request it closer all right. So, which is of course, not will be very good if you have locality in the access row all right. So, essentially there what will happen is that every request will take this much of time row miss time all right. And the other 1 is called open page policy where you keep the row open until you run into …

So, in that case what will happen is that you will first time you take a row miss, but otherwise you will either have row hit and row conflict and this particular policy is exercise by memory controller. So, the memory controller designer will figure out what to do. So, if the designer can design a good prediction policy for closing the row buffer we will actually use the closed page policy.

So, by the way I am using the word page here because the often this is also called a a page buffer or D RAM page. This particular 1 do not get confused with a waste page there is nothing to do with waste page that is totally different all right. So, is this clear to everybody how I access the D RAM what actually matters when I accessing the D RAM what decisions would actually influence my D RAM latency.

So, these are very simple overview there are many other kind of parameters that you would take care off for example, he asked where there is a time required for activating a row right. So, and there are many other times that I am use, but this is what is a is a first order effect of accessing the D RAM. For example, there are other second order effects like when you write to a D RAM bank that is write to row buffer how much time must laps. Before you can read the row buffer again there are certain constants all right you cannot immediately read that row buffer all right.

So, if you are interested I can send you the link you can read about ah D RAM vendors data chips you can see many more details. If you I do not know if you have ever try to explore d rams, but if you do you find the D RAM latency is normally mentioned as something like 10 10 10 3 numbers that usually right. So, those are these 3 numbers. So, this is what you actually mentioned all right all right.

So, what I do is I will work you through 1 example where ah I take a typical D RAM and actually look at how addresses are decoded into bank rows columns etcetera all right. So, this is 1 thing that I can show you in this course unfortunately other things are all inside the chips I forgot that is why I brought these dim cards. So, if you have seen this thing memory cards. So, this is what they look like.

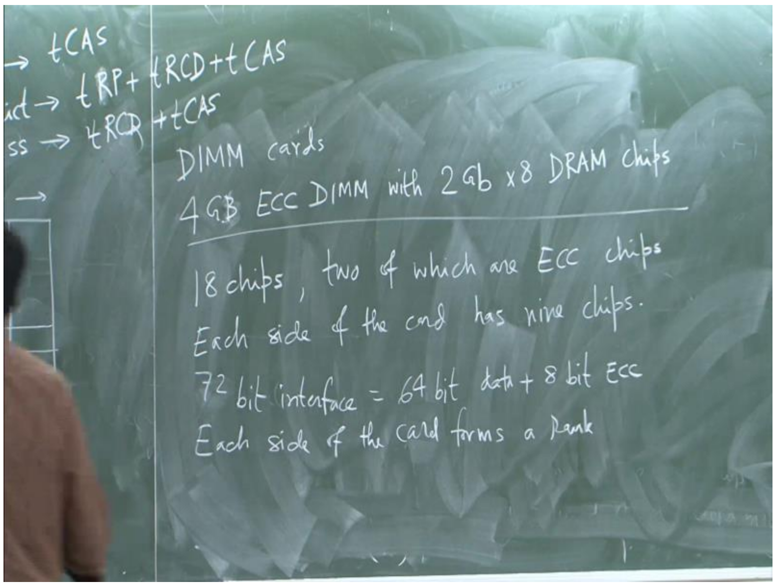
So, you see this black chips right these are D RAM chips and they normally appear in both sides they can appear on the only 1 side also. So, this 1 has this has this has 4 chips on this side over here there are eight chips, but these are DDR 25.5 megabyte memory card. And these are the pin connectors you can see copper wirings in the bottom right and this 1 is the DDR connector this is the these are DDR memory card this is not DDR 2 all right same capacity, but the layout is different.

So, 1 way to figure out whether it is DDR 2 or DDR is that if you align the DDR card with the DDR 2 card. This is I can show you to you find that these 2 notches here are not align actually. So, the DDR 2 notch will appear early the DDR it DDR notch. So, above 1 is the DDR card and lower 1 is DDR 2 card. So, you can see you can read about exact layout of DDR 2 and DDR why they do not align and etcetera.

So, what I am going to do is I am going to talk about 1 such card all right how what exactly it means. So, you can see the 9 chips are on this side nine on this side there are eighteen chips right out of which actually 16 a data chips 2 are error writing chips. So, let us try to see what we actually goes inside this. So, this particular write-up you also post on the course web page. So, you can read about that this is just a summary of 1 particular D RAM from micron.

So, what you do is you look at one. So, this is called dual in-line memory module this 1 both of this dual in-line memory module or DIMM.

(Refer Slide Time: 16:12)



→ tCAS

ıd → tRP + tRCD + tCAS

ss → tRCD + tCAS

→

DIMM cards

4 GB ECC DIMM with 2 Gb x8 DRAM Chips

18 chips, two of which are ECC chips

Each side of the card has nine chips.

72 bit interface = 64 bit data + 8 bit ECC

Each side of the card forms a Rank

These are called DIMM cards they are used to be single in-line memory module or SIMM card long back in ninety's they are less number of pins, but today no 1 would buy the SIMM card. They are not in the market or you get the DIMM card all right and remember that which has nothing to do with whether the chip circuit on 1 side or both sides there is nothing to SIMM or DIMM. So, we are going to look at 1 4 g b e c c DIMM with 2 gigabit x eight D RAM chips. So, I am I am not explain what is quotation bits; so, this 1 stands for e c c. Does anybody know?

Sorry

Error writing code

Error writing code exactly. So, each D RAM chip is 2 gigabit in size all right. So, how many chips do I need to cover a 4 gigabit gigabyte.

16.

16 right this is 4 gigabyte this is 2 gigabits all right. So, normally whenever you see a D RAM chip capacity they are normally mentioned in bits how many bits is all right. So, there are sixteen chips, but this DIMM card would actually have eighteen chips 2 of

which are e c c chips. So, what is the e c c density how many e c c bits for byte of data. Can anybody guess.

1 bit per byte

1

1 bit per byte

1 bit per byte exactly. So, usually the simplest e c c protocol is that you store the exon of all the eight bits with the single bit all right. So, why exon?

Sir, if priority changes.

Right, exactly if 1 bit flips then I should be look actually all right. So, we have eighteen chips 2 of which are e c c chips. So, 16 are data chips all right. So, each side of the card has nine chips there will be some small control chips also on the on the control card. But, we are ignoring them all right and this x 8 here means that each D RAM chip it provide eight bits of output. Whenever you send a particular command to the D RAM chip can either write or read eight bits all right. So, usually these DIMMs have a 64 bit interface sixty 4 bit interface in the memory controller all right. So, how many chips I need to fill up 64 bits each chip can provide eight bits right.

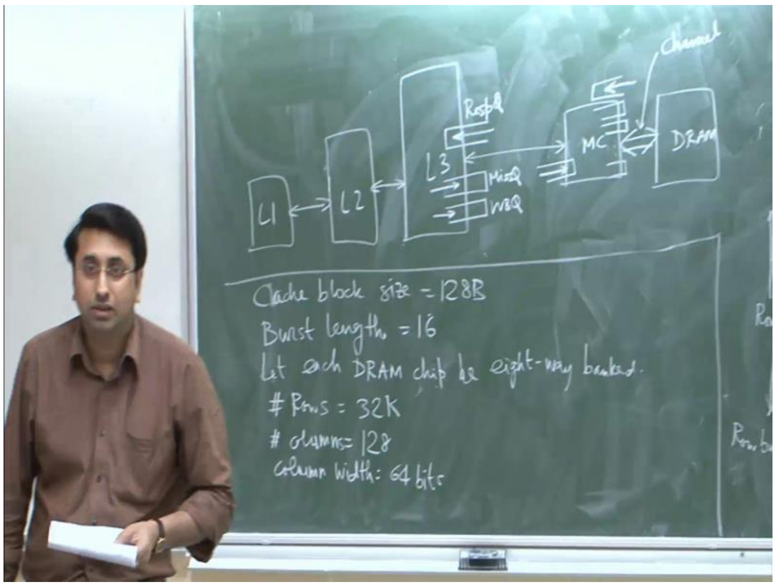
8 bits

8 bits right. So, whenever a request goes normally I will activate eight D RAM chips out of 16 of course, 1 e c c chip will also be activate all right. And then what will happen is that these 8 chips will give me total of 64 bits of data. So, let us see how will actually does that each side of the chip forms a rank each side of a card.

So, in this case, so this is just an accident that once a cards forms a rank in general rank means that a subset of D RAM chips that participate in generating a data packet all right. So, in this particular case it happens that we need eight D RAM chips to generate a data packet which is on 1 side of the card all right. Other you can you can easily come up with a with a different you know D RAM architecture where you can find that only half of 1 side will be 1 actually that is possible.

So, this 1 is actually seventy 2 bit interface which is 64 bit data plus eight bit e c c all right. So, we have 2 rams here just find out the ram now I will write here.

(Refer Slide Time: 21:42)



So, now the memory controller when it receive the request from this side you receive a request for cache block instead of waiting for that.

Now, let us assume that we have a cache block size 128 bytes now we get sixty 4 bit data in 1 request from the D RAM all right. So, how many how many how many bus for connections I need here 16 right and we got sixteen such transaction to fill up 1 cache block. So, this is often called the burst length of the d ram. So, burst length is configured when you moved the memory controller in this case is going to be 16 burst lengths.

Now, definitely want we really want is that the first request there is a first eight byte request for the cache block may have a row conflict or row miss depending on the protocol. But, that is remaining 15 transactions should be row hits that is what I will expect and that is how we must we must put our row and column I d’s in the axis. So, that a cache block may have 15 row hits and may be at least I mean at most 1 more bits of complex 1 16 could be row miss also.

So, let us see and this 1 is called a channel that we discussed last time and you can have multichannel memory controllers as well and that is also possible. So, in this case what will happen is that you connect 1 DIMM to 1 channel and another DIMM to another channel all right. So, there will be 2 separate DIMMs connected to 2 channels and that will give you better aggregate timing.

Let us assume that each D RAM chip be eight way banked literally all right. So, and each banks looks like this it has bunch of rows bunch of columns all right. So, that means, when I want to access a piece of data what I need is for this particular 1 I need to tell the row I d. So, first of all I need to choose my bank id within the bank I will tell the row id I will tell the column id. And also I will tell the column offset that is from where this 8 bit should come within the column because 1 column may not be 8 bit it may be longer all right.

So, for this particular 2 g b 2 gigabit D RAM chip let us assume that we have number of rows equal to 32 k which is 2 to the power of 15 number of columns equal to 128 and column width is 64 bits. So, that unity defines my bank organization or the bank itself. So, how big is the row buffer?

128 and 64

How is that in bytes?

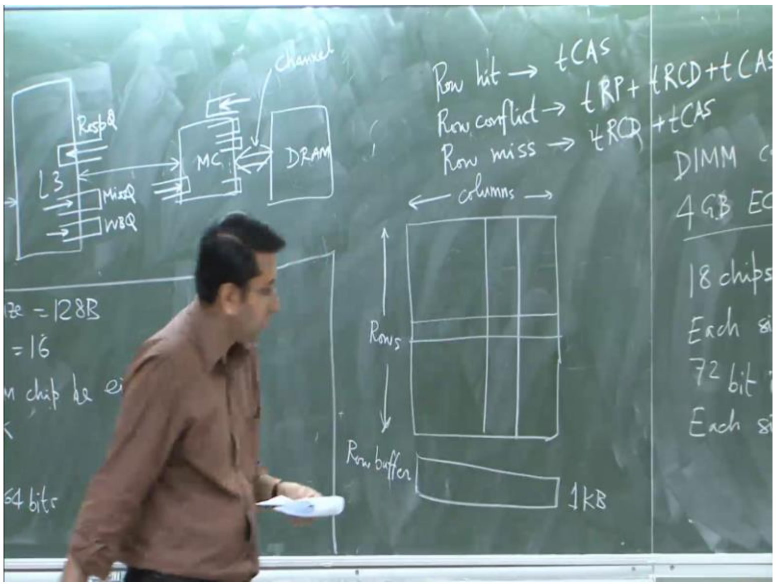
1 0 to 4 0

1 0 right

1 0

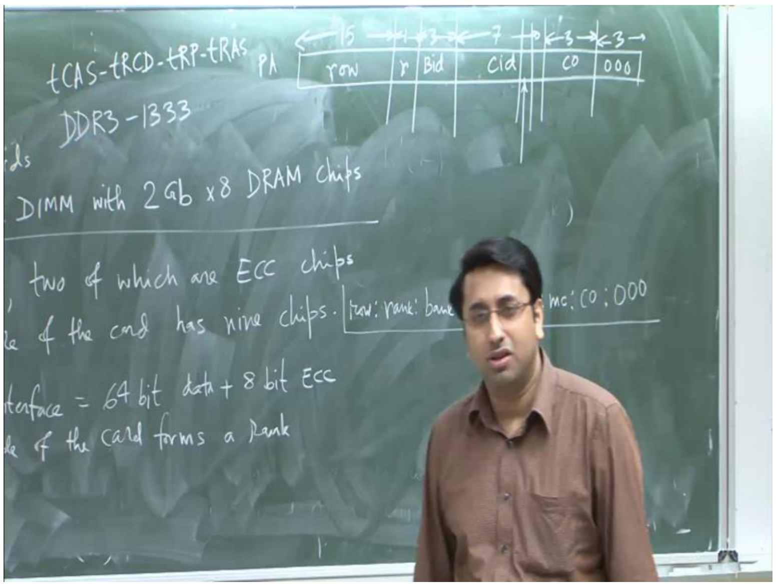
Yes.

(Refer Slide Time: 25:52)



So, we have 1 kilo byte row buffer per bank ok all right. So, let us see how the address is important 1.

(Refer Slide Time: 26:07)



tCAS-tRCD-tRP-tRAS
DDR3-1333
PA
15
3
7
3
3
row
r
Bid
cid
co
000
DIMM with 2Gb x8 DRAM chips
two of which are ECC chips
of the card has nine chips.
row: rank: bank
mc: co: 000
terface = 64 bit data + 8 bit ECC
of the card forms a Rank

So, this is my physical address right now whenever memory controller sends a request to the D RAM it allies it to 8 pipe bounder. Why is that?

Because, the interface is of sixty 4 bit data we always talks it also 64 bit times. So, it allies it to 8 byte 4 bit which means a last 3 bits are 0. So, any physical address that the D RAM chips will have last 3 bit is equal to 0. So, which means it can be ignored all right.

So, after that comes the column offset which tells me within a column which bits should be these eight chips. So, how many bits do I need for column offset column width is 64 bits.

3 bits.

Why 3 bits

Because 8 bit address 3 bits 8 byte will address.

Exactly. So, I have I have eight possibilities updating eight bits output out from the 64 bit column all right. So, this is 3 bits column offset then comes the column id which is seven bits and then the bank the bank id which is 3 bits then comes the rank which is 1 bit. And we need 15 bits for the row how well it does come from 32 bits right and that is what we would expect we access the 4 gigabyte DIMM we need 32 bits of all right.

So, when the when the memory controller gets the particular physical address its job is to do this it partition of the address into this different fields, removes the last 3 bits and tells the D RAM about the following things the column offset the column id. And the row id and it sends these 3 things in the corresponding rank in the corresponding rank. And the request get broadcast to all the chips in that rank that would actually provide the data transfer.

So, 8 chips would now what concurrently and within this chips each of this chips will actually operate on this particular bank and would access this particular row. This particular column and with the column this particular offset and each chip will provide you eight bits will finally, you get a 64 bits of data and eight bits of parameter. And then, you communicate that over the channel to the memory controller. And then the burst length is set to 16 which means the D RAM will now every cycle which keep on providing all such character until the burst length is exhausted at which point the request completes. In case of write of course, there they do not be any communication memory controller the write will happen 1 after another which also wait points is it clear.

Sir each transfer is called a transaction.

It is usually called a burst.

Over there

Yeah that that is the no which comes

Each transfer within a burst

8 byte

Yes

Yeah, that is a burst you make 16 burst to get a cache block any question on this clear now there are couple of things that you have to resolve if I have a multichannel memory controller where should I put my channel bits in the address. So, in other words given address I must be able to figure out on which channel to send this particular request. And each channel is a beam connector. So, where should I put my channel bits? What makes more sense?

On this side

Here

Yeah

Why?

What is the reason?

You have ignored the last 3 bit.

That is that is because the address are align

Yeah.

So, imagine that I have a dual channel memory controller right. So, what is the purpose of having dual channel? So, that I can make both the channels I can replies both the channels concurrently right what miss mean to that. So, what will what will enable both the channels to use concurrently how should I put the channels what is the grain of a request.

When you schedule the request order both channels are reused.

That is right yes.

Actually the first bit is previous 1

First bit

First bit of this single address

Why do you say so?

I want 2 independent request to go to 2 channels right. So, that they can work independently what is a grain of a request that memory controller access.

Byte

Sorry

Bytes cache blocks.

Cache blocks right. I said 1 cache block to this channel the next 1 to that channel right. So, adjust on to it the cache blocks with channels. So, I will put it right after the see in this case I have 120 byte cache blocks. So, I put the 7 bit here this would be my channel the column id will gets spilt unfortunately in this case right. So, this is 1 one the most useful wire put in the channels so that, you can alternate between cache blocks.

Sir why are the row bits towards the …

Yeah I will get that yes I will coming to that why this is coming this way yeah just 1 more question before we get that what if I have multiple in multichannel memory controllers. So, then I have to select the memory controller as well where should I put them what makes more sense.

Yeah. Does it make sense to switch alternate cache blocks to memory controller?

No, because we have multiple in multichannel.

Fine. So, within which memory controller I will alternate between channels right that that make sense why we put the row id on the most significant side. Can anybody explain why think about these latencies, does it maximize the amount of contiguous data in a row if I put the row id on the most significant side?

It does right.

Yes.

So, which means if I sequential access I would have

Most and then row hits except the first axis all right ok

When you introduce not you state.

Yes I do write the locality, but if you still look inside a channel within a channel still continues yes over and I am still writing the locality, but within the channel itself use.

Sir would not you would not the channel id in the least significant after the three. So, the channel needs to.

No there is a there is a technical problem in between that then essentially your switching bits across channels which cannot be done. So, in the column offset cannot be spilt across a channel that is not possible actually.

Column offset.

No essentially what I am saying is that why wouldn't split 1 request across the channels that is what you are saying right.

Yeah

Yeah. So, I am I am saying that you know why would you not have given independent request to channels. So, they can proceed independently because what may now the problem of what you are proposing is that now you have to wait for both the channels to complete before responding to the processor. But, with 2 independent channels given to 2 cache blocks they may be responded out of order actually as soon as 1 completes cycle respond to the bus I think that is about it.

Yeah. So, the latency parameters I have just mentioned.

So, they have specified like t CAS t r c d t r p. So, that is the 3 numbers you often find a fourth number which is t ras the time taken for a ras operation. So, that is how the D RAM shifts normally specify the latency parameters otherwise we normally talk about d rams either like. So, DIMMs are normally talked about like this the D RAM chips would be like this also along with this there would be a DDR specification in a frequency parameter.

So, they look like this DDR 3 13 33. So, what this means is that this is the third generation double data rate D RAM and the effective frequency is thirteen 33 megahertz my effective frequency what I mean is that. So, double data rate D RAM can transfer data on both the clock edges which means in this case if it is a DDR D RAM it would transfer 64 bits on both the clock edge. So, which means to have 16 burst we would actually require.

8 cycles

8 cycles the question is what is the frequency of this cycles what is the cycle time this 1 is not the not the actual frequency; half of this is actual frequency this is an effective frequency. At this frequency we require 16 cycles all right half of the frequency will require eight cycles all right. So, keep that in mind. So, that is pretty much about it that you need to know if you want to really make an educated decision about buying a DIMM card I do not think you need know anything else so that, the marketing if you cannot.

So, it is required anything else here and that is about it. So, this is about d ram. So, we are talking about the main memory right and you might notice that. So, 1 obvious question that 1 could ask is why do you separate the RAS and CAS operations. Why do you send the row edges and column edges together all right? That would definitely save a lot of time because I could actually combine many commands together and probably give out your data much earlier. Because here, what I do is a I first activate the row then I you know read the row out and then I activate my column all right. So, instead if I had the row edges and column edges I could probably read out only this much and could give you the data much.

Any idea why the why is the separation between basis RAS and CAS what could be the reason for doing this? So, normally these the row address and the column address. So, as I said you know when I mentioned the I said that the memory controller sends the row id column id and then column offset to the D RAM bank controller. They actually do not excel together first you send the row id and then you actually communicate the column id and column offset on the same address bus with multiplexing both DIMMs 1 columns. Why that is what is the reason why? So, under what circumstances you normally think about multiplexing?

When you want to know the selection for

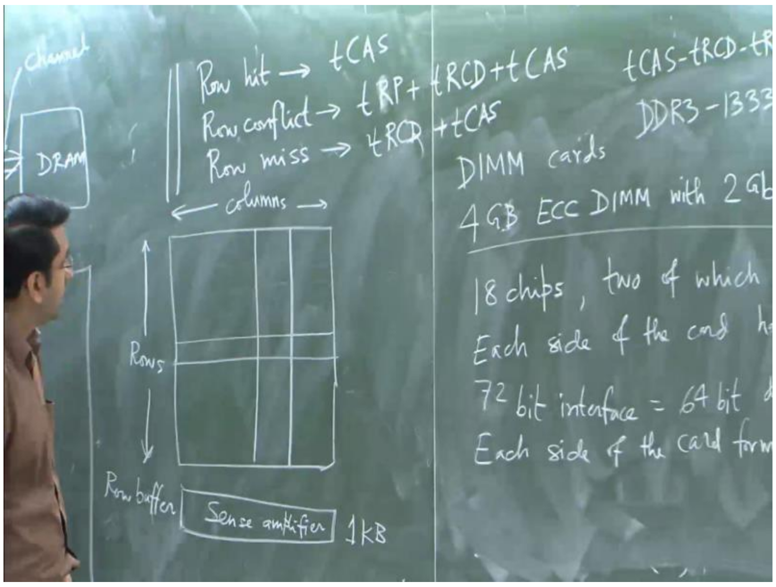
When you want to know the selection

No, here I am talking not talking about multiplexor we are talking about multiplexing 2 things on a single physical resource that is address bus right. You first send the row address and then you send the column address why do you why do you need to do these things.

Yeah, exactly when you are short of wax right that I do not do not have you know where is to communicate the row and column edges in parallel. So, the point is that the D RAM chips have pin limited it have very much pin limited and the reason is that of course, obvious question why? And why cannot we this number of pins what hertz increasing number of bits the reason is that that hertz a density actually in D RAM. The density is extremely important in how many bits you can pack in bit area that is the factor when you design the D RAM.

So, when you have more pins you normally have more periculum circuitry and that hertz the density of the D RAM. So, here we are we never mentioned the area actually, but that is the very important factor how much area you need to pack this 2 gigabit you know of data all right. So, that is why normally the D RAM chips circuit limited and which is why multiplex and why column axis on the same addresses bus on a factor.

(Refer Slide Time: 41:12)



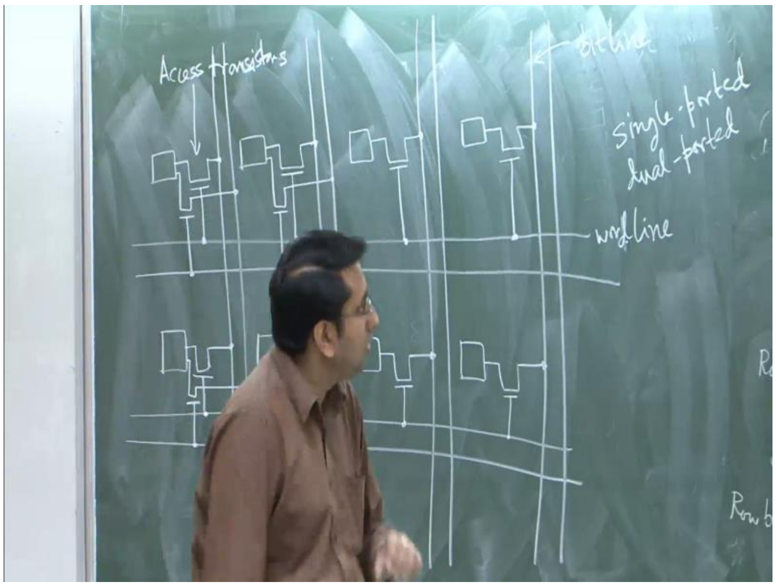
This 1 is called sense amplifier as a technical name if you want to read up on that I can give you some references. So, a narrowing of sense amplifiers actually hold the data gets the row buffer and then it involves a lot of electrical engineering how to design good senses in sense amplifier refresh data is I already talked about. So, periodically the memory controller which scheduling refreshes cycle is on the other hand.

So, in a refresh cycle what you normally do is. So, the first of the entire first thing that happens is that the bank which is continuing refresh cannot be access by the processor at the time. So, that is that is the performance it acts on different cycle.

So, during refreshing what you do is you first close this 1 whichever is open in this bank and then readout each row at a time 1 after readout the row. Write it back that is all that is all the refresh does nothing much just read 1 at a time. So, that refreshes the content of the row. So, that is the refresh cycle, but remember that here you have this 32 k row you have to refresh all of them yeah 1 after another that takes a lot of time which is why D RAM refresh is really an expensive operation. And it is normally not done very frequently every D RAM data sheet comes with the nominal time that the D RAM can operate without refresh.

So, the memory controller designer has to schedule a refresh operation within that nominal time. So, that the data is not cross all right. So, 1 more small thing that I want to mention I want to open this up a little bit more.

(Refer Slide Time: 43:05)



Showing 1 more example, so I will show you 2 rows of the D RAM first show you how it works. So, these are more bit rows that I am showing that often 2 rows of D RAM this is the D RAM cell which stores the bit all right. And this is the switch all right which closes whenever I give a high voltage here on this particular wire all right. Whenever I give a high voltage here this closes and what about the value of this here will appear on this particular wire all right.

So, here I am showing 2 different rows this is called a word line and this 1 often called a called a word this is called a bit line. Now, suppose I want to access this particular row. So, what do I do I apply the high voltage on this word line and then I readout this 4 bits 4 four bit lines. So, I get the value in the in that particular row as simple as write.

Now, of course, a lot of engineering goes inside this how to design this particular bit all right how to maintain this and there are many D RAM architectures like for example, there are D RAM cells of only 1 transistor. Therefore, the beating a single capacitor actually implementing using a 1 transistor.

Now, suppose I want to change this these 2 row D RAM. So, that I can access both the rows simultaneously how may be how may be this currently it is not possible where. In fact, we appreciate that if I enable both the word rows the 2 rows will get jumbled double row I will not get any good data. So, how do I enable this when I want to access both the rows what changes when implement.

Sir, in that below 4 besides these 4

Say again

Shift that below 4 cells to beside.

No, that id does not change those rows. So, since I cannot access both the rows here this is called a single ported ram this is single ported ram I can only access 1 of the rows. And you can see that I can add more rows just in the same way. Now, how do I make it more compact?

Sir I access 2 bits.

In 1 cycle we have to access.

Number this is how to access of course, otherwise I can access 1 way another right

Negative

No nothing of this

More bit lines and bridges

Just more bit lines that is enough

No, sir I am switches to connect to both bit line

Why word lines?

Yes.

Yeah. So, I have 1 more word line per row I have 1 more bit line per column and I have 1 more switch etcetera all right now it is clear right I can now enable this row right etcetera and so on. So, I can not only access the same row twice I can access these 2 rows also simultaneously. So, these are dual ported ram what am I pay for making the dual ported.

Wires sir

Wires

Anything else anything comes of latency I tell you that it will be slower than single point why?

Slow.

1 if tell me exactly why it is slower. So, compared to the single ported ram is it true that the word line make increases yeah the length of the word lines spending 1 row does it increases.

Yes sir

It does right because there happen to some gap between the width lines right. So, this width increases does the length of the bit line increase it does right.

Yes sir

So, the time to charge is proportional to the length of the line is r c delay. So, as you add more codes is going to slower and slower ram structures and this is the reason why all right I think I am going to stop here. So, 1 more things these are called access transistors this things and often taken together this whole block here it is called an access stack. So, you need a figure access stack if you have more codes that is not the implication is all right. So, that is pretty much it about D RAM.

So, next time I will take a look at the s ram a little bit at the static random access memory which is used for caches very similar. Just a few extra points that need to be test upon and then we will move on to something else, but that is up to the data of course.

Sir, what is the dynamic here?

This particular implement is not this particular bit now 1 is implemented using 1 capacitor all right. Just 1 capacitor which needs several in static ram this bit is normally implemented as CAS in DIMM orders which advantage of this thing electronic. Of course, if you go cup would be water that acts as a ranch right which holds the bit. So, there is no question of weakly node in this.

So, sir we increase the length of word line

Yes

Number of burst length of bit line.

Yeah

Latency

You will increase.

Yes increase I mean

It slows now

There are more I mean row sizes more and we get the number of rows we increase number of rows actually.

We increase width of each row.

Yeah

Then we decrease number of rows.

So, you make the make the ram wider that is what you are saying it depends how wide the rams are because see.

Precharge latency is to precharge.

Bit line, but these all rows are count the time charge the word line this until that stabilizes the access on these 2 cannot be activated; the switch cannot act properly right. So, this all both the things on the critical path, yeah it may be switch there you may reach switch point where the total latency becomes smaller yes almost smaller.