UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

----------X

AMAZON.COM, INC. AND AMAZON.COM SERVICES LLC,

Petitioner,

CASE.: IPR2024-00691

-against-

NOKIA TECHNOLOGIES OY,

Patent Owner.

----------X

DEPOSITION OF

DR. IAIN RICHARDSON

NEW YORK, NEW YORK

FEBRUARY 27, 2025

REPORTED BY: KIARA MILLER

FILE NO.: 566945

AMAZON-1017
Amazon.com, Inc. v. Nokia Techs. Oy
IPR2024-00691

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

----------X

AMAZON.COM, INC. AND AMAZON.COM SERVICES LLC,

Petitioner,

CASE.: IPR2024-00691

-against-

NOKIA TECHNOLOGIES OY,

Patent Owner.

----------X

Deposition of DR. IAIN RICHARDSON, taken on behalf of PETITIONER, at 1301 AVENUE OF THE AMERICAS, New York, New York, commencing at 8:05 A.M., FEBRUARY 27, 2025, before Kiara Miller.

A P P E A R A N C E S:

ON BEHALF OF PETITIONER:

PERKINS COIE
11452 El Camino Real, Suite 300
San Diego, CA 92130

EMAIL: TKENDRICK@PERKINSCOIE.COM
JCARTER@PERKINSCOIE.COM

BY: TYLER KENDRICK, ESQ.
JON R. CARTER, ESQ.

ON BEHALF OF PATENT OWNER:

MCKOOL SMITH
300 Crescent Court, Suite 1200
Dallas, TX 75201

EMAIL: SUDICK@MCKOOLSMITH.COM

BY: STEVEN UDICK, ESQ.

A P P E A R A N C E S C O N T I N U E D . . .

ON BEHALF OF PATENT OWNER:

ALSTON BIRD
1201 West Peachtree Street, Suite 4900
Atlanta, GA 30309

EMAIL: NICK.TSUI@ALSTON.COM

BY: NICHOLAS T. TSUI, ESQ.

DR. I A I N R I C H A R D S O N, after having first been duly sworn by a Notary Public of the State of New York, was examined and testified as follows:

EXAMINATION BY

MR. KENDRICK:

Q Good morning, Dr. Richardson.

A Good morning.

Q So we met briefly before your deposition started, but my name is Tyler Kendrick and I am one of the attorneys for Amazon.

Can you please state your name for the record?

A Iain Edward Richardson.

Q Thank you very much.

And you can hear me okay since we're sitting right across each other?

A Sure. Yeah, we're good.

Q The court reporter read your oath beforehand, so you understand you're under oath today?

A Yes.

Q And do you have any documents in front

of you?
A I have two binders in front of me, and I think there is a total of seven documents in those binders.
Q Okay. What are those seven documents?
A So I have a copy of Exhibit 1001, the US patent 805-0321. A copy of Exhibit 1003, declaration of Dr. Havlicek. A copy Exhibit 1005, which is a document ISO/IEC and then there's a number 11172-2. Possibly refer to that as MPEG-1 or MPEG-1 video. I have a copy Exhibit 1006, US patent 691-2351.
Q Just for the record, those are in one of the binders that you have in front of you?
A That's right.
And in the other binder I have three documents, Exhibit 2001, which was my declaration, which I believe is filed with the patent owners preliminary response. I might call that my first declaration.
There's Exhibit 2004, which is a copy of my declaration, which I believe was filed

with the patent owner's response and I might call that my second declaration. And, finally, there's Exhibit 2005, which is a transcript of the deposition of Dr. Havlicek.

Q Thank you.

And are any of the documents that you just covered marked or annotated?

A Not as far as I'm aware. I certainly haven't marked or annotated any of these.

Q Any other documents in front of you?

A There's a laptop in front of me which you provided.

Q Okay.

MR. KENDRICK: Just for the record, so the laptop has clean versions of the papers and exhibits that were filed in this IPR.

Q As you know, the ground rules will be similar to the deposition you had yesterday. If you could -- we'll try to take breaks every hour, if you need a break sooner than an hour, just please let me know, all I ask

is that you finish the pending question and then we can take a break.

During those breaks, if you cannot communicate with your attorneys about the substance of your testimony or this IPR, that would be helpful. Talk about the weather or the rain outside is fine. Is that okay?

A Yes.

Q I will try to ask clear questions, but if you don't understand anything, can you please let me know?

A Okay.

Q And audible answers will be helpful for Kiara, so no head nodding yes, nos instead of uh-uh, uh-huh; is that okay?

A Okay.

Q I will do my best to let you fully provide your answer, excuse me, and if you can also do your best to let me ask my full question before responding that will be helpful for her as well; is that okay?

A Understood.

Q And then your attorney may object from time to time, but unless he instructs you not to answer, I'll ask that you still answer the question; is that okay?
A Understood.
Q Is there any reason why you cannot testify truthfully here today?
A Not that I'm aware of.
Q Any medications that are affecting your memory or impairing your memory?
A No.
Q So as I mentioned before there's a laptop with exhibits and papers from the IPR, my understanding is that those are keyword searchable and if you open them up in PDF, so you can search around for them. If something turns out not to be searchable, just let me know. There's also a Chrome browser opened up to what should be an electronic version of your H.264 book that we purchased, so you made a little bit of money off of this deposition. I don't have a hard copy of that

book, but if there's any questions about that, let me know.

A Okay. And just on that point, it appears to be an electronic copy of the book that I wrote in 2010. I've written two books relating to H.264.

Q What was the other book related to H.264?

A I think the title was H.264 and MPEG-4 Video Compression or possibly the other way around. And I wrote that in 2003.

Q Okay, great, thank you.

And you submitted two declarations in this IPR, correct, I believe you referred to them as your first declaration and your second declaration?

A Yes. I -- yeah, was one -- in this one I think my recollection is I submitted two declarations.

Q And the first declaration you submitted was associated with Nokia's patent owner preliminary response and that's labeled 2001;

is that right?
A That's generally my understanding. I just wrote the declaration.
Q Understood.
And then you wrote a second declaration, which is Exhibit 2004?
A Yes.
Q And your understanding is that was associated and filed with Nokia's patent owner response?
A That's my understanding. Obviously I didn't do the filing.
Q Understood.
And so for shorthand, we'll try to refer to Exhibit 2001 as your first declaration or preliminary declaration; is that okay?
A That's fine, yeah.
Q And then for Exhibit 2004, I'll try to refer to that as your second declaration or response declaration; is that okay?
A Okay, yes.
Q Your second declaration incorporates

your opinions from your first ones; is that correct?

A Yes. And I state that in Paragraph 2 of my second declaration.

Q Great, thank you.

And both your declarations contain your opinions about the validity of what I'll call the '321 Patent.

Do you understand what the '321 Patent is?

A I understand what the '321 Patent is, they -- both declarations contain opinions that I've provided with respect to that matter.

Q Also with respect to Amazon's prior art grounds that it contains in this petition?

A So as part of providing those opinions, I reviewed a number of documents including the petition document in this matter, so if that's what you mean, then I would agree with you.

Q Do you understand that Amazon has

challenged the validity of Claims 8, 9, 10 and 11 of the '321 Patent?
A I don't actually recall without checking whether that's the entirety of the claims that were challenged.
Q If I represent to you that is the entirety of the claims, would you agree with that?
A I can't think of anything that would make me disagree with that.
Q And if I refer to these claims, Claims 8, 9, 10 and 11 as the challenged claims, is that okay with you?
A Okay, yes.
Q Just to orient ourselves, if you want you can open up Exhibit 1001 in your binder on the laptop that was provided to you.
A Okay.
Q If you want to go to Claim 8, please, which is the last Page 13, it starts at the very bottom of Column 23.
A Okay.

Q And you've been an expert in numerous patent litigations; is that correct?
A I've been retained to give technical opinions in a number of patent litigation matters.
Q So if you look at Claim 8, it starts with what's called a preamble, which has a method before decoding a compressed video sequence, the method comprising.
Do you see that?
A I see those words and I recollect that can be described or denoted a preamble.
Q Then the parties here have broken up Claim 8 into different limitations and just to make sure we're on the same page, we'll do some shorthand for this limitation.
So if you look at the bottom of Column 23 it starts with the first limitation, which starts from decoding from the video sequence and indication.
Do you see that?
A Yes.

Q Is it okay if we refer to that as limitation 8A?
A That's fine. I'd ask if any time -- well, if I want to remind myself, I'll check with you. So obviously I don't have markings on this particular exhibit. Though I think I do refer to Claim 8A, element 8A, in one or both of my declarations.
Q Then if you look at the top of Column 24, Line 3, there's a second limitation called, "Starting the decoding of the video sequence."
Do you see that?
A Yes.
Q Is it okay if we refer to that as Limitation 8B?
A With the same proviso and it might be helpful to me if where possible you say the limitation beginning starting decoding, so I'm answering the question from the right perspective.
Q Great, I'll try to do that.

Yeah, if you have questions or need clarification on that, please let me know, I'm happy to provide it.

And a couple more similar questions, so if you look down at Line 7, there's a limitation that starts with decoding identifier values, is it okay if we refer to that as limitation 8C?

A Yes. With the same proviso.

Q Great. Finally, there's a the last limitation of Claim 8, excuse me, that starts with resetting the identifier value. Is it okay if we refer to that as limitation 8D?

A Yes. With the same proviso.

Q Great. What did you do to prepare for today's deposition?

A I reviewed the documents in my binders. I reviewed other exhibits and, yeah, other documents that I recall being, I recall being relevant to this proceeding. I met with counsel, both online and in-person.

Q Okay. So we'll unpack that a little

bit.

What other exhibits outside of the ones in your binders did you review?

A I don't actually remember an exhaustive list.

Q Do you remember part of that list?

A Um. I think I recall reviewing the petition document exhibit. I think I recall reviewing the institution decision.

Honestly, I was probably reviewing a number of exhibits, I can't remember exactly. It's been a number of days and I've been working on three different IPR proceedings.

Q Understood. It's tough to keep all of it straight.

Do you recall if the other exhibits you reviewed, were they all filed in this IPR?

A Yeah, so for the purposes of preparation for today's deposition, that's what I recall is reviewing exhibits, documents. I don't necessarily know the technical difference between them that relate to this IPR.

Q In your earlier answer you said that you recall reviewing other documents that were relevant to this proceeding, do you recall what documents those were?
A So I mentioned that I recall reviewing at some point in the recent past, for example, the institution decision, I don't know whether that's technically an exhibit or not.
Q Okay. Any other documents that are relevant to the proceeding that you recall reviewing?
A I can't think of others, but there may have been others.
Q Okay. And you also said you met with counsel both in video and in-person --
A Yes.
Q Is that right?
How long did you meet with counsel?
A So I met with counsel a number of times on Zoom over the last few months in relation to -- counsel from the same firm, if you'd

like, in relation to three or no, sorry, four IPR proceedings, I can't always necessarily unpack it. I do recall that sometime last week I met with counsel specifically in relation to this IPR on Zoom and I think it was around, for around two, two and a half hours.

Q Okay. And how long did you meet with counsel in-person to prepare for this deposition?

A So I recall this week meeting with counsel for a full working day on Monday, a full working day on Tuesday, and a part of that time was spent preparing for this deposition, part of it was spent preparing for two other depositions that I understand were planned this week, one that happened yesterday and another one that hasn't happened yet.

Q And can you estimate how long y'all spoke about the '321 IPR in particular?

A Not really, no.

Q Okay. And then is it your understanding that the '321 Patent is also involved in a ITC proceeding?

MR. UDICK: Objection to form.

A I recall being aware that there is or was an ITC proceeding that this patent is or was involved in. I don't know the status of that proceeding.

Q Okay. Were you involved in that ITC proceeding?

A I wasn't retained as far as I'm aware to provide any advice or analysis, yeah. My recollection is I provided -- I met with counsel with regard to this patent and this IPR, but I don't recall any work I was asked to do in relation to other proceedings.

Q Okay. In preparing your first declaration, did you -- for this IPR -- did you review any information from the ITC proceeding between Amazon and Nokia concerning the '321 Patent?

A Not that I recall.

Q Okay. When preparing your second declaration for this IPR, did you review any information from the ITC proceeding between Amazon and Nokia concerning the '321 Patent?
A Not that I recall.
Q Same question for your deposition.
When preparing for this deposition, did you review any information from the ITC proceeding between Amazon and Nokia concerning the '321 Patent?
A Not that I recall.
Q If you can turn to Paragraph 17 of Exhibit 2004, which is your second declaration.
A Okay.
Q And in the paragraph you state that your opinions in the declaration are based on the documents you reviewed, the petition, including all materials cited therein, your professional judgment, as well as your education, experience, and knowledge regarding technologies relating to the '321

Patent.

Do you see that?

A Yes.

Q Okay. Are the opinions in your second declaration based on documents that are not specifically referenced or identified in that declaration?

A Yes. To the extent that my judgment, education, experience, and knowledge regarding technologies related to this patent, for example, the video coding field in general. You know, all of that goes back more than 30 years, and I've reviewed many, many documents in that time, written many documents. So I haven't necessarily listed them all in this declaration.

Q Are there any documents in particular from your 30 years that you recall relying on for this -- to form the opinions contained in this declaration?

A To the extent that everything -- all of my work in video coding over that period has

informed my current level of knowledge, experience, and judgment. It's really hard to separate anything out.

Q Okay. So outside of the information stated in Paragraph 17 that you say you relied on to form your opinions, is there anything else -- or any -- are there any other documents or information that your opinions are based on but not listed or referenced in Paragraph 17?

A I'm not sure what you're asking me really beyond what I just said. That everything I've worked on in the last 30-plus years has influence, my education, experience, knowledge, and judgment in relation to this field. So, yeah, again, I can't think of anything that stands out, but I certainly don't want to exclude all the things that have influenced my level of knowledge and experience and so on.

Q Understood.

And you don't hold yourself out as an

attorney or legal expert; is that right?
A No. I'm not an attorney or a legal expert, as I say on my -- Paragraph 18 of my second declaration.
Q So the legal standards that are in your declaration were provided by counsel for Nokia, would that be fair?
A They certainly have, I think generally provided me with these legal standards in this section.
Yeah, in the Section 4, Roman numeral IV, legal standards of my second declaration, just flipping through. I think everything in here -- every paragraph starts with "I've been informed by counsel."
Q Okay.
A So, yeah, I've been involved with a number of patent proceedings before, so I've been informed by counsel in other matters on things like obviousness and so on. But, yeah, these paragraphs reflect what I've been informed by counsel of, with regard to this

matter.
Q Got it. Thank you.
And if you can move down to Paragraph 38 of your second declaration.
A Okay.
Q And Paragraph 38 discusses the definition of a person of ordinary skill in the art.
Do you see that?
A Yes.
Q Okay. And if I refer you to that as a POSITA, would you understand what the acronym refers to?
A I think so, yes.
Q Okay. And in Paragraph 38 you state that your analysis applies Amazon's definition of a POSITA; is that right?
A Petitioner is Amazon, as I understand it. So, yeah, this is -- I understand this to be petitioner's definition. And I have applied that definition.
Q Thank you, yeah.

Petitioner is Amazon here. So I may refer to them as Petitioner Amazon; is that okay?
A I prefer petitioner, just to keep it straight for me.
Q Got it. I will do my best to use the term petitioner.
And is it your understanding that petitioner's petition covers three grounds of invalidity of Claims 8 through 11 for the '321 Patent?
A That's broadly my understanding, yes. And my recollection, rather.
Q And for in ground one, petitioner argues that -- well, I guess let's back up here.
When we were talking about your -- or Exhibit 1005, when you were listing off the documents in your binders.
Do you recall that?
A I do, yes.
Q And Exhibit 1005 is the second part of the ISO/IEC 11172 Standard; is that right?

A Yes. Sorry let me just doublecheck.
Q No problem.
A Yes. So I'm looking at the front page of Exhibit 1005, and the standard -- the official designation, as I understand it, is ISO/IEC 11172-2, and part of the title underneath is "Part 2 video," so I think that's consistent with what you said.
Q Great. Thank you.
And Part 2 concerns video; is that right?
A So this standards document, 11172-2 concerns video, yes.
Q Okay.
A And, again, I'm happy to refer to it as the MPEG-1 video and also I might refer to it as MPEG-1 and it's shorthand for this document.
Q That was my next question. So we can refer to this as MPEG-1 and put aside the portions of MPEG-1 for systems audio and compliance testings; is that okay?

A Yes, unless they become relevant, yes.
Q Great. If they become relevant, I'll try to specify that.
A Okay.
Q And looking -- for petitioner, petitioner's ground one argues that MPEG-1 renders Claims 8, 10 and 11 of the '321 Patent obvious; is that right?
A Yeah, I can look up the petition, but that's generally my recollection, yes.
Q And you understand that Amazon is not asserting that MPEG-1 anticipates Claims 8, 10, and 11, right?
A So I'm going to open up the petition document, because I think these questions relate to that.
Q No problem, take your time.
A Okay. So I'm looking at the petition in this matter, Page 10 of the petition, Ground 1, MPEG-1 renders obvious claims 8, 10, and 11. And I notice the petitioner has also stated they're going to use the same

shorthand, MPEG-1, to refer specifically to this document, ISO/IEC 11172-2.

Q Great. Thank you.

And then for Ground 2 of petitioner's petition, it argues that the US Patent No. 6,912,351 renders Claims 8, 10, and 11 obvious.

Is that your understanding?

A So I'm looking at the petition in this matter, Page 38, Ground 2, Kim renders obvious claims 8, 10, and 11. So petitioner talks about a reference, Kim Exhibit 6 -- sorry -- Exhibit 1006. My understanding is that Exhibit 1006 is a patent to the inventor with the surname of Kim. Patent No. 6,912,351. And again, I'm happy if we refer to this as Kim, and I think I've done so in my declarations.

Q Great. That was going to be my next questions, so that's perfect. So we'll refer to that patent, which was submitted as 1006 as Kim.

Then for Ground 3, Amazon argues that MPEG-1 in combination with a patent referred to by the surname of Yagasaki, renders Claim 9 of the '321 Patent obvious; is that right?

A Again, reading from the petition, Page 66 of the petition, that's pretty much what it says. MPEG-1 and Yagasaki render obvious Claim 9. And there's a reference to Yagasaki as Exhibit 1015, and I will just confirm the number of that -- the patent number, rather, just a second.

Q No problem.

A So I just opened up Exhibit 1015 on the laptop that you provided and that is a patent to Yagasaki and others, the patent number is US 5,786,858. And, again, I refer to that as Yagasaki; petitioners refer to that as Yagasaki. I'm happy if we just refer to that as Yagasaki from now on.

Q Okay. And in your second declaration you only argue that Claim 9 of the '321

Patent is not obvious for the same reasons you opined in MPEG-1 doesn't render a Claim 8 obvious.

Do you agree with that?

A So I state in Paragraph 186 of my second declaration, it's my understanding that Claim 9 depending on -- sorry.

Yeah, I state that Claim 9 depends on Claim 8 in the '321 Patent. And the petitioner in the petition, with respect to this ground, relies on its Ground 1, which is -- relates to MPEG-1. And because in any opinion, MPEG-1 fails to teach or suggest all of the limitations of Claim 8, Ground 3 fails.

And I don't recall providing any further opinions than that in my second declaration on this ground.

Q Okay. In your first declaration, do you provide any opinions that Claim 9 is obvious for reasons other than the fact, in your opinion, that MPEG-1 doesn't render Claim 8

obvious?
A So with regard to that question, the Paragraphs 122 and 123 of my first declaration are relevant. And I make a similar statement with respect to Ground 3, Yagasaki and MPEG-1, and I don't recall making any other -- or providing any other opinions on this ground in my first declaration, other than what is provided in those two paragraphs.
Q And you would consider yourself an expert on video and image compression and encoding; is that right?
A Yes. I think that's fair.
Q And I think you mentioned this earlier, you have over 30 years of experience in video and image compression and encoding?
A Yes.
Q What is video and imagine compression encoding?
MR. UDICK: Objection to form.
A So at a high level, video compression

involves converting video data, might be described as raw video data or uncompressed video data, or -- yeah, similar terms -- into a compressed form that is suitable for storage or transmission or other processes; and also converting a compressed form of video data into uncompressed video data, which could also be described as raw video data or displayable video data. That's a high level video, image coding.

To be honest, in the field, the two are sometimes used with overlap. But if we're considering images, then it would be the same process of representing -- converting an image that isn't displayable or uncompressed or raw form into a compressed form suitable for storage, transmission, and so on, and/or converting it from a compressed form into a displayable or raw form.

And, yeah, just to be clear about my answers, I was attempting to explain what video coding and image coding are. The field

of image coding might be described as being wider than that.

Q Thank you for that clarification. I appreciate it.

So just to kind of summarize here. Video coding involves two processes, encoding to convert raw data to an encoded bitstream, and then decoding that bitstream to reproduce that raw data.

Would you agree with that?

MR. UDICK: Objection to form.

A I would say those two processes, depending on how you define the terms, but those two processes are consistent with known processes, which are in the field of video coding.

As I just mentioned, video coding, if you ask me about my experience in video coding, it doesn't merely include those two processes.

Q What other processes would it include?

A So, again, colloquially, if you ask me

what my field is I would say video coding. If you ask me what I've been involved with over the years, it would include those two processes. It would also include transmission and storage of compressed video. It would also include issues relating to the capture of video, the display of video. And some, yeah, some of my work has involved -- relating to video coding has involved processes that are outside of those two, for example, pre-filtering and post-filtering. Human factors, such as subjective quality of uncompressed and compressed video and images and the list probably goes on beyond that.

Q Understood. It's quite a broad field.

I think focusing on two of those processes.

So the encoding process would be performed by what is colloquially known as an encoder.

Would you agree with that?

A A video encoding process or a video

compression process can be performed by a video encoder, which -- yeah.

Q And the shorthand for a video encoder is just encoder.

Would you be okay with that shorthand?

A I prefer to use video encoder whenever possible, unless we're in the middle of a discussion. I guess because encoder could have other meanings in the general field, you know, for example, data communications.

Q And the decoding process could be performed by a video decoder.

Would you agree with that?

A Did you say would or could?

Q Could.

A A video decoding process could be performed by a video decoder.

Q What else could perform a video decoding process?

MR. UDICK: Objection to form.

A For example, one example I can think of could be a transcoder. So Transcoders are

used in this general field or can be used, for example, to convert between compressed formats or to do some other processing on a compressed video signal and a transcoder will often or can carry out video decoding processes and video encoding processes.

Q And a video encoder would carry out prediction transforming in coding processes to produce a compressed bitstream.

Would you agree with that?

A Which video encoder?

Q Just a video encoder in general. Are there specific types of video encoders that are confusing the question?

A So, for example, a motion JPEG encoder would generally not carry out prediction.

Q Is there a type of video encoder that would carry out prediction?

MR. UDICK: Objection to form.

A Yes. I spoke too quickly.

There are known video encoders that carry out prediction processes and I'm

familiar with examples of video encoders that do that.

Q Okay. And then a video decoder essentially refers to the process employed by a video encoder to decode that compressed bitstream. Would you agree with that?

MR. UDICK: Objection to form.

A Which video decoder?

Q What types of video decoders are you aware of?

A I guess I'm aware that there are many implementations of video decoders that conform to standards, for example. There are implementations of video decoders that don't conform to standards. There are implementations, that I'm aware of, that are proprietary. So I don't actually know what they're doing inside. So all of these video decoders that I'm aware of.

Q Looking at a video decoding that conforms to standards, as an example, would that decoder refer to the process employed by

the encoder to decode a compressed bitstream?
A So there are quite a number of published video coding standards, and also published specifications that are arguably not standards.

If I take one, for example, take an MPEG-1, a decoder that conforms to MPEG-1 that starts with a compressed bitstream and decodes it to produce -- or could start with a compressed bitstream and decode it to produce decompressed video frames.

The process in such a decoder, I'm aware of MPEG-1 encoders that would carry out processes that at some level might be considered complimentary to those carried out in the decoder. In MPEG-1 encoders that I'm familiar with or some MPEG-1 decoders that I'm familiar with, the encoding processes are not completely the reverse of the decoder and vice versa.

For example, I'm familiar with software and hardware MPEG-1 encoders that carry out

motion estimation, which is an aspect that MPEG-1 decoder doesn't, according to the standard, have to carry out. So there's a level of complementarity, for instance, in the MPEG-1 encoders that I'm familiar with, there is variable length encoding in MPEG-1 decoding, according to the MPEG-1 standard there is variable length decoding. So there is a complementarity there. As I said, some aspects of the encoder and decoder aren't necessarily complementary.

And also, again, taking MPEG-1 decoding as an example, an MPEG-1 decoder has an inverse quantization or rescaling step which is not a full reversal of the corresponding encoders, forward quantization step, because quantization is a lossy process.

Q What is a lossy process?

A Generally, in this field, for example, taking MPEG-1 as an example, and let's say a software MPEG-1 encoder that I'm familiar with, and a decoding -- an MPEG-1 decoder

that operates according to the MPEG-1 standard, that's the MPEG-1 part two standard we discussed.

The input to the decoder is source video frames -- sorry. The input to the encoder is source video frames. The output of a decoder is decoded video frames. Decoder frames are not necessarily identical to the original video frames. So the process, as a whole, and the example that I'm thinking of can introduce loss.

So you could call it a lossy compression process in the MPEG-1 example and the specific encoder implementation I'm thinking of, the loss is primarily, perhaps entirely introduced by the forward quantization process in the encoder that I'm thinking of.

Q So the loss process wouldn't be caused by the decoder, would you agree?

A In the example I'm thinking of, the decoder -- huh, okay. There's a slight nuance with MPEG-1 because the decoder has a

transform that is not exactly specified. It's specified within the accuracy bounds. If I ignore that part of it, then the decoder that follows the MPEG-1 standard, every decoder that follows that standard should produce the same output to within the transform accuracy bounds I mentioned.

And so, I would say, apart from the transform in decoder in that MPEG-1 compatible decoders don't introduce further loss. I think that's correct.

Q So looking back to one of your previous answers we talked about, when an MPEG-1 encoder and decoder may not be complementary. An example you gave where an MPEG-1 encoder and decoder may not be complementary is the inverse quantization process because it's a lossy process.

Do you recall that?

A Broadly, yes. The inverse quantization process or rescaling process in an MPEG-1 decoder is not a lossy process in itself. I

think that would be correct to say.
Q Are there other examples where an MPEG-1 decoder would not be complementary to the encoder?
MR. UDICK: Object to form.
A There certainly could be because the MPEG-1 standard doesn't specify how a bitstream should be encoded. So in one sense that process that produces an MPEG-1 compliant bitstream is, you know, could in theory be created in very a large number of ways.
So it's quite hard to sort of say for MPEG-1, in general, or for an MPEG -- a decoder that complies with the MPEG-1 standard, if I consider an encoding process that produces a bitstream that complies with the MPEG-1 standard, it's hard to sort of put boundaries on what could be inside that encoding process, for the reasons I've discussed.
Q Does the MPEG-1 standard specify how a

bitstream should be decoded?
A So the MPEG-1 standard specifies, among other things, a syntax for compressed video, starting in Section 2.4.2. It specifies the semantics for that syntax is a meaning of the syntax elements or how they should -- yeah, what -- something about what those syntax elements mean and how they should be treated in Section 2.4.3. And then in Section 2.4.4 the standard specifies a video decoding process.

So I think Section 2.4.4, along with the rest of the standard was -- you know, would answer your question, there is a specification of a decoding process.
Q And in your previous answer you said that an MPEG-1 compliant bitstream could be encoded in a large variety of ways.

Would you agree?
A I forget the exact words that I used. What I was trying to explain was that the standard, as I said, specifies the bitstream

and specifies a process for decoding the bitstream -- it specifies the bitstream by specifying the syntax and the semantics. And it specifies a process for decoding that bitstream. It doesn't specify how that bitstream must be created by an encoder.

Q So an encoder would have options during encoding as long as the resulting bitstream conformed to the MPEG-1 standard?

MR. UDICK: Object to form.

A What do you mean by "an encoder" in that question?

Q Well, you said that MPEG-1 does not specify how a bitstream must be created by the encoder. So an encoder that is creating an MPEG-1 bitstream would have options on how to encode and create that bitstream, as long as it conforms to the MPEG-1 standard.

Would you agree?

A So if I think of an encoder that produces an MPEG-1 compliant bitstream, and I'm aware of -- come across a few over the

years, and I can think of a software one, for example, such an encoder -- yeah, has -- can create -- yeah, in theory can create the bitstream any way it likes as long as -- or any way the designer likes, as long as the bitstream complies with the standard and includes being decodable, successfully decodable by the specified video decoding process.

Q And the decoder would process the -- excuse me. Let me start over.

An MPEG-1 decoder would process the MPEG-1 bitstream it receives and decode that bitstream according to the syntax element values in the bitstream.

Would you agree with that?

A So a video decoder that complies with MPEG-1 -- in fact, I think the standard might even state it better than I could paraphrase it.

So the MPEG-1 standard, so it talks about this in several places, as I recall,

page -- stamped Page 15 of Exhibit 1005 says:

Decoding is the inverse of the encoding operation, this is just the words of the standard, it's considerably simpler than encoding. There is no need to perform a motion estimation. There are many fewer options. The decoding process is defined of by this part of ISO/IEC 11172.

And later on stamped Page 18, under 1.1, Scope, it says:

"This part of ISO/IEC 11172 specifies the coded representation of video -- I'm paraphrasing -- and specifies the decoding process."

And I've kind of forgotten the question. Can you ask it again, but I think these are relevant.

Q Yeah. No problem.

So the question was, an MPEG-1 decoder would process the MPEG-1 bitstream it receives according to the syntax elements in that bitstream.

Would you agree with that?
A I think I would put it slightly differently.
A video decoder compliant with MPEG-1 should carry out a decoding process that produces the same result as the decoding process specified in the standard. It should be capable of carrying that out for a coded bitstream that follows the syntax and semantics specified in the standard.
The decoding process essentially has such a bitstream as its input and should produce a decoded output that is the same as that produced by the decoding process specified in the standard.
Q When an MPEG-1 decoder is decoding a bitstream, is it allowed to deviate from the MPEG-1 standard?
MR. UDICK: Object to form.
A So when a decoder that complies with the MPEG-1 standard is decoding a bitstream that complies with the MPEG-1 standard, there

is -- not sure how clearly it's set out, but my understanding is there's a requirement that the output that decoder produces should be the same to within certain tolerances as the output of the decoding process specified in the standard.

And, in fact, yeah, I'd also draw your attention to another part of this standard, which we touched on earlier. Stamped Page 49 of the MPEG-1 exhibit at the top says:

> "Compliance requirements for decoders are contained in ISO/IEC 11172-41."

So, actually, if I wanted to answer this kind of question as accurately as possible, I'd probably want to look at the section of the standard. I'm happy to continue answering questions from my general knowledge as I understand it.

So -- yeah, that's a long answer, but I think it's all -- to be precise, it's all relevant. And so, yeah, the decoder should

produce the same output as the decoding process to within certain tolerances, but I don't recall the standards saying that every step of the decoding process has to be followed exactly as it is set out in the standard, as long as that output is the same.

Q When the decoder -- when an MPEG-1 decoder decodes the bitstream, can it make decisions that are independent of how the encoder encoded that bitstream?

MR. UDICK: Objection to form.

A Okay. I don't understand what you mean.

Q Well, in Paragraph 57 of your second report, it's on Page 20. The second sentence, you state:

"The decoder processes the received MPEG-1 bitstream, decodes the bitstream according to syntax element values in the bitstream."

Do you see that?

A Yes.

Q When the MPEG-1 decoder is decoding the

bitstream, does it have choices about how to decode the bitstream or does it have -- sorry. Let's just end it there.

Does it have choices to how to decode the bitstream?

A I used the phrase an MPEG-1 decoder, not the MPEG-1 decoder.

So an MPEG-1 decoder, I think what I wrote in Paragraph 57 as a summary, as a summary I think it's technically correct. Such a decoder, so when I say "the decoder," I mean an MPEG-1 decoder, a decoder that complies with the MPEG-1 standard decodes a bitstream according to the syntax element values in the bitstream.

How I write that in software or how I design that in hardware is kind of up to the designer, as long as all the requirements of the standard are met.

Q Well, in the first sentence of Paragraph 57, you state:

"The encoder makes decisions and

choses options whilst encoding the bitstream."

Do you see that?

A Yes.

Q Can the decoder also make decisions and chose options whilst decoding the bitstream?

A For example, a software MPEG-1 decoder, I could write a software MPEG-1 decoder, I could make decisions as long as they don't -- I, as the software designer, I could write the software so the software makes decisions such as whether or not to display certain frames. As long as those decisions don't contradict the requirements of the standard that I've already discussed.

So, yeah, sure, there are decisions, for example, that might affect the display to a user, if I'm, just for the sake of argument, writing software to decode and display frames in realtime.

Yeah. So a decoder can make -- decoding implementation can make decisions. My

recollection is that the first sentence of Paragraph 57, I was trying to summarize the fact that an encoder that produces a bitstream compliant with MPEG-1, has certain choices or can make certain choices that influence, for example, the values that certain syntax elements take. A decoder can't make that kind of choice.

One example might be in the sequence header, stamped Page 36 of Exhibit 1005, there's a syntax elements horizontal_size. It's a 12-bit value. If I look at the semantics for that, stamped Page 41 of the same exhibit, it says:

"The horizontal_ size, which is the syntax elements, is the width of the displayable part of the luminance components in peles or pixels."

So that is a syntax element has a value and the encoder is responsible, or an encoder that creates -- if this bitstream was created by an encoder, that encoder would be

responsible for making a choice of what that horizontal size should be.

A decoder doesn't have such a choice. A decoder compliant with this standard would read that syntax element and process the video accordingly.

Q So in your -- part of your answer you said, before you got to the example, you said that an encoder has certain choices or can make certain choices that influence, for example, the values that certain syntax elements take, and you said a decoder can't make that kind of choice, do you recall that?

A Broadly, yes.

Q Okay. If an encoder makes a choice on what value a syntax element takes, can a decoder change the value of that syntax element?

MR. UDICK: Objection to form.

Incomplete hypothetical.

A I don't quite know what you mean. MPEG-1 was directed towards -- I'm reading

from Page 12 of the stamped exhibit, represent and compressed video on various digital storage media such as CDs and it lists some other examples of digital storage media.

So the idea of MPEG-1 at the time was a code -- to provide a code of representation that could be put onto a video CD, which as I recall predecessor of the DVD disc and so once it's on the CD, it's on the CD. And the decoder in that scenario, or an MPEG-1 in that scenario, read that bitstream from the CD, decode it, process it according to the standard.

Q So you talked about the example I think horizontal size for the sequence header on Page 36 of MPEG-1?

A Yes.

Q If the encoder sets a specific horizontal size, could the decoder change what the horizontal size is?

A I'm struggling to see how a decoder

could change a CD, for example. A read only CD, is that what you're asking.

Q Yeah, that's what I'm asking.

If an encoder sets a horizontal size, specific horizontal size to the sequence header, can a decoder change that value when it decodes the bitstream?

A Okay. Short answer is, I could create a nonstandard decoding process that did something weird with that, but an MPEG-1 decoder or video decoder compliant with the MPEG-1 standard, has to, as I recall, decode that syntax element and then use that value. I don't recall anything in the standard that says the decoder can change that value.

MR. KENDRICK: We've been going about an hour, do we want to take our first break?

MR. UDICK: Sure.

(Whereupon, a recess was taken from 09:10 AM until 09:22 AM.)

Q Welcome back, Dr. Richardson.

During the break did you discuss the substance of your testimony with counsel?
A No.
Q Great, thank you.
So before we took our break I had a question about whether a decoder can change the value of a syntax element that is sent by the encoder.
Do you recall that line of questioning?
A More or less, yes.
Q Okay. So if you have an MPEG-1 compliant decoder, can it change the syntax value of -- or change the value of a syntax element set by an MPEG-1 compliant encoder?
MR. UDICK: Objection to form.
A To be honest, I can't imagine a scenario where it could.
Q Okay. And I believe you used the term complementary earlier, so an encoder and a decoder that are, that conform to a video coding standard need to be complementary of each other, do you agree with that?

A So a standard such as MPEG-1 specifies a decoding process, it does not specify an encoding process. So a video encoder that produces a bitstream that conforms to MPEG-1 for example, the only really restriction is if the bitstream has to conform to MPEG-1 be decodable in the manner that is set out in the standard, which can include -- yeah, it's sort of more rigorously defined perhaps in later standards, but it can include -- sorry, where am I going?

The encoder has to produce a bitstream that is decodable according to the standard. Yeah, there is a process for compliance. I've forgotten the page, now. In another Section of the MPEG-1 standard.

Lost it now, so yeah, to cut my long answer, bring it to a close. An encoder or an encoding device that produces an MPEG-1 bitstream, it just has to produce a bitstream that meets the requirements of the standard. So it doesn't necessarily have to, I think I

said this earlier, carry out precisely complementary processes in a sort of structure or way defined in the standard. The only absolute requirement is it produces that bitstream.

Q I want to go back to one other thing I forgot to cover earlier.

So a couple of questions ago I asked, so if you have an MPEG-1 compliant decoder, can it change the value of a syntax element set by the MPEG-1 compliant encoder, and you said to be honest, I can't imagine a scenario where it could; do you recall that?

A Yeah, I didn't want to, to the extent that I accepted your characterization of an MPEG-1 compliant encoder, I didn't want to accept any particular characterization, but otherwise, I recall an answer along those lines.

Q Understood. If a decoder did change the value of a syntax element from the value set by an encoder, what would happen?

MR. UDICK: Objection to form.
Incomplete hypothetical.
A No idea. Yeah, without a specific example.
Q Would the bitstream, would the decoder still be able to decode the bitstream?
A No --
MR. UDICK: Objection to form.
A Sorry, no idea unless I have a specific in mind.
Q Okay. So earlier I believe we discussed your work with H.264 and you mentioned that you wrote two books on H.264; is that right?
A Yeah, so I wrote a book that was published in 2003 that covered the H.264 Standard and focused on that standard and also on the MPEG-4 part two video compression standard, and in 2010 or published in 2010 was my second book relating to H.264 -- my second book relating to H.264, which focused specifically on the H.264 Standard.
Q Okay. And if you look on the laptop

that was provided to you, there's a Chrome web browser open to a digital copy of your second H.264 book published in 2010, does that look to be a copy of that book?

A Yeah, I'm just -- we might need the internet just to -- yeah, so I can't open the front matter which has the, should have the copyright date and so forth.

MR. KENDRICK: Can we take a two-minute break?

(Whereupon, a recess was taken from 9:28 AM until 9:29 AM.)

A Yeah. So this looks like an electronic copy my 2010 book.

Q Okay. And if you can navigate to Section 3.1 of that book?

A Okay.

Q In the second paragraph of Section 3.1 it starts with the sentence, "Compression involves a complementary pair of systems, a compressor (encoder) and a decompressor (decoder)."

Do you see that?
A Yes.
Q What do you mean by a complementary pair of systems there?
A So it's 14 or 15 years since I wrote this, my recollection is that chapter three, video concept is the title of the chapter and it's meant to be a really high level introduction to video compression. And I can't remember precisely, but I think I was trying to summarize at an extremely high level the fact that the encoder is converting source data into compressed form and the decoder is converting a compressed form back into a representation of the source data or the original video data.
So I think that's the aspect of complementarity I wish to capture there.
Q And you said in your answer, you said the decoder is converting a compressed form back into a representation of source data, what do you mean by a representation of

source data?
A So I'm just reading from that section of the book or pretty much paraphrase or read from that section of the book.
So you're asking me what I meant at the time?
Q Well, with what you meant from your answer. Your answer said converting a compressed form back into a representation?
A I want to be precise in my answer, so I was reading or paraphrasing from the section you asked me to look at. So you asked me what I thought at the time or what I would mean by that if I were to consider it now.
Q Let's start what you thought at the time?
A It was 14 or 15 plus years ago when I was writing this. So I don't recall exactly.
Q What about now, what does that statement mean to you now?
A So the text is the decoder converts the compressed form back into a representation of

the original video data. I think that would be consistent with, for example, with typical standards based, standards compliant decoders that I was familiar with at the time and unfamiliar with now, which start with -- which have an input of a compressed form, i.e., I think there I'm sort of trying to summarize compressed video bitstream.

And have an output of decoded video frames and are representation of the original video data, I think that's consistent with the fact that many but not all, practical, yeah, many but not all video encoders and decoders I'm familiar with, the system is lossy as we discussed earlier. So a typical MPEG-1 decoder, the output is not necessarily going to be identical to the original video frame, it's going to be a representation of the original video frames, or hopefully from the user's perspective something that looks adequate quality, but is not necessarily identical to the original data.

Q When the decoder converts the compressed bitstream back into a representation of the original video data, does it change the bitstream?

MR. UDICK: Objection to form.

Incomplete hypothetical.

A So this is a very general statement. So if I consider a video decoder compliant with the MPEG-1 standard, it reads the bitstream, it decodes the syntax elements in the bitstream to obtain values, it applies a process that at least produces the same result as the video coding/decoding process specified in the standard, and it creates or decodes output video frames, hopefully displayable video frames.

Q When a decoder or video decoder decodes syntax elements to obtain values, does it change the value of the syntax element.

MR. UDICK: Objection to form.

Incomplete hypothetical.

A I can't think of an instance in MPEG-1

where that happens.
Q What about H.264, would a H.264 decoder when it decodes syntax elements to obtain values, would it change those values?
MR. UDICK: So objection to scope and form.
A I think that's a different question, could you ask it again.
Q The last question or you answered you said an instance of an MPEG-1 decoder, looking at a decoder outside MPEG-1 and taking, for example, a decoder that conforms to the H.264 Standard.
MR. UDICK: Objection to the scope as H.264 is not relevant to the proceeding and then to form.
Q When an H.264 decoder decodes the syntax elements to obtain values, would it change those values?
MR. UDICK: Same objection.
A Which values?
Q The values of the syntax element it's

decoding?

MR. UDICK: Same objection.

A Okay. So I don't have the H.264 Standard in front of me, I've obviously worked with it for a number of years, so from memory the H.264 Standard specifies a syntax and semantics for a compressed bitstream. The syntax includes certain defined syntax elements. Each of those syntax elements, the standard, specifies how to decode that syntax element to generate a value and -- so that's the process that an H.264 compliant decoder should follow.

Q And when an H.264 compliant decoder decodes the syntax element to generate a value, will it change that value?

MR. UDICK: Objection to scope and form.

A Okay. Which syntax element?

Q Do you need a specific syntax element to answer that question?

A I don't know because there are a lot of

syntax elements in the standard itself as you're probably aware. Covered 600 plus pages and I don't have all of it in my hand.

Q Understood. Can you think of a syntax element where the decoder can change the value once it decodes it?

MR. UDICK: Objection to scope and form.

A So I can roughly remember some of the names of the syntax elements in the H.264 Standards. I can't even remember -- I wouldn't like to speculate on the precise names of these. And then standards specifies exactly what should be done within each syntax element and it's different syntax element by syntax element, so it would be easier to answer with a specific syntax element in mind and standard in front of me. I'm happy to do that.

Q In your second declaration do you recall discussing H.264?

A So in Paragraph 10 of my second

declaration I mentioned that I'm the author of two widely cited books from the H.264 MPEG-4 industry standards for video compression. I mean, I can do a search, I don't recall whether H.264 if I mentioned it again in that document.

Q No problem. Do you want to turn to Paragraph 106 on Page 41?

A Okay. Oh, yes.

Q So in Paragraph 106, you see you discuss an H.264 syntax element referred to as a NAL unit, N-A-L, do you see that?

A Yeah, it's not a syntax element.

Q But it defines a syntax element?

A A NAL unit is, as I said at the top of Paragraph 106, a syntax structure.

Q And that syntax structure includes a NAL unit type; is that right?

A So as I state in Paragraph 106, the NAL unit syntax, referring to H.264, includes a parameter called MAL, that's M-A-L, _unit_type and my recollection is that is a

NAL_unit_type in H.264 is a syntax element.
Q And would an encoder that's compliant with the H.264 Standards set a value for the NAL_unit_type syntax elements?
MR. UDICK: Objection to form.
A Which encoder?
Q Do you need a specific encoder to know when it sets the NAL units type syntax element?
A Okay. So maybe stepping back. What do you mean by a video encoder that complies with the H.264 Standard? Do you want me to consider a video encoder that produces a bitstream that is compliant with the H.264 Standard or something else?
Q Yes.
A Okay. And what was that the question?
Q For a video encoder that produces a compliant -- sorry let me start that over.
For a video encoder that produces a bitstream that is compliant with the H.264 Standard, will it set the value of the syntax

element NAL unit type?
A So I don't recall whether the syntax element NAL unit -- NAL_unit_type is a requirement of every H.264 compliant bitstream or not. And I can work that out from the standard document.
And the reason that I say that is because some syntax elements can be set either in the bitstream or some parameters can be set as syntax element values for outside of the bitstream, according to the standard. Others are specified as being part of the bitstream syntax, and then there's also profiles to consider.
Q Okay. If the NAL unit type is set or if the NAL unit type is part of an H.264 compliant bitstream, would the encoder set the value of that NAL unit type?
MR. UDICK: Objection to form.
A Which encoder?
Q An encoder that produces a bitstream compliant with H.264?

A So an encoder, video encoder that produces a bitstream that's compliant with H.264, could set the value of -- and if it included the NAL_unit_type syntax element in that bitstream, such an encoder could signal a value or select an encoded value for that syntax element.

Q Assuming an encoder selects and encodes a value for the syntax element NAL unit type, that syntax element would be decoded from the bitstream by a decoder.

Would you agree?

MR. UDICK: Objection to scope and form.

A So if a decoder that complies and conforms with the H.264 Standard decodes a compliant bitstream, if that bitstream has syntax element NAL_unit_type in it, and if the decoding process specifies in the standard requires the decoder to actually decode and process that syntax element, then the decoder will decode that syntax element.

Q And when the decoder decodes this NAL unit type syntax element, will it change the value that was set by the encoder?

MR. UDICK: Objection to form and scope.

A So I don't have the H.264 Standard in front of me. I don't recall a scenario, as I sit here, where that would happen, or could happen, but I don't have the Standard in front of me.

Q But you are very familiar with the H.264 Standard; would you agree?

MR. UDICK: Objection to form.

A So I've worked with and worked on the H.264 Standard and written about the H.264 Standard for quite a number of years. I've written two books about it or that cover it, so I'm familiar with the standard. It is a very large document, so I don't have the entire thing in my memory.

Q Understood. I'd be impressed if you did.

If a decoder did decode the NAL unit type syntax element from the bitstream and then change that value, what would happen?

MR. UDICK: Objection to form and scope and incomplete hypothetical.

A Yeah, I don't know what the scenario is where it's changing the value and what the reason for that is and how that relates to processing the decoder does after that.

Q Okay. But you can't imagine -- you testified that you don't recall a scenario where the decoder would change the value of the NAL unit type syntax element that decodes from the bitstream?

MR. UDICK: Objection to form.

Misstates prior testimony.

A Yeah, I don't remember exactly what I said. I can't think of a -- you know, I've worked with a number of H.264 compliant decoders. I don't remember an instance when a decoder decoded such a value or was programmed to decode such a value and then

changed the value of the syntax element -- that syntax element. I just don't recall.

Q Okay. If you can turn to Section 4.3 of your H.264 book. It'll be on the browser. The second sentence, fist paragraph under Section 4.3 states:

> "An H.264 video decoder carries out the complementary process of decoding inverse transform and reconstruction to produce a decoded video space."

Do you see that.

A I do --

MR. UDICK: Objection to the form and scope.

A Sorry. I do see those word.

Q What do you mean by that sentence?

MR. UDICK: Same objection.

A Yeah, similar answer to before. So I wrote this probably around 15 years ago. Are you asking me what I meant by it then or what I think it means now?

Q What do you think it means now?
A So I don't actually recall all the writing processes 15 years ago. So this is a part of a book on H.264. Yeah, I'm going to preface my answer here.

The book is not the standard and I think I probably say that at least once in the book. The book is not the standard. It's meant to be a way into the standard, I think I even wrote that in the introduction. So it's a way to the help readers understand the standard.

So here I'm trying to summarize at a really high level some aspects of what a typical or a known or a possible H.264 encoder might do. And I think what I was talking about there was an encoder that produces a bitstream that complies with H.264. And I'm talking about a -- what I had in mind, perhaps, is a typical decode -- H.264 decoder, so a decoder that conformed to the H.264 Standard.

And when I used the word "complementary processes" or the phrase "complementary processes" there, I'm not entirely sure what I meant at that time, but it was possibly the fact that there are processes in a video -- a typical video or a known video decoder or a video decoder that, perhaps -- I'm reading this -- but might come across that are complementary. I think maybe the word "complementary" I used to voice necessarily implying that every process is an exact inverse.

So, for example, looking at my Figures 4.4 and 4.5 in that 2010 book. Figure 4.4, I've got a final stage of the encoding shown here as the entropy encoder and the first stage of decoding shown in the next figure is entropy decoding. And I think I was, perhaps, intending to explain at a high level these two complement each other, so that the -- and, actually, an output of the decoder is, I think in H.264 the same as

input to the entropy encoder.

And then you've got the encoder, going back one, you've got transform quantize. In the decoder you've got inverse transform and inverse quantize or rescale. And those are complementary to each other. They're not exact inverses of each other, for reasons that I discuss in the book. But I think I used the word "complementary" to reflect that, you know, one could look at how I describe it in the book later on.

You can look at transform and coding and transform of the quantization in Chapter 7. In the same portion of the chapter, look at inverse transform and rescaling because those are in the terms that I've used here, describing those as complementary.

Q If the decoder was not complementary of an encoder, could the source bitstream -- or the -- excuse me -- could the source frames be reproduced by the decoder?

MR. UDICK: Objection to form.

A So a decoder that conforms to H.264 doesn't actually need to mirror anything about the encoder, according to the standard. The standard just says it has to be capable of decoding a coded bitstream as specified in the standard to produce output decoded video frames using a process that is either the sames as or produces the same results as the process defined in the standard.

So the standard, as I recall, again, I don't have it in front of me, the standard doesn't say anything about the decoder having to use complementary processes to the encoder in that scenario.

Q In your opinion, is the '321 Patent related to improving random access coding?

A I think that's pretty much what I say at Paragraph 41 of my second declaration, or one of the things I said.

Q And you say that it enables a decoder to identify where to start a decoding process such as when a user randomly selects a frame

to start viewing video.

Do you see that?

A Yes.

Q Okay. And in Paragraph 41 you give an example of when this could occur, which is when a user is trying to move forward or backward in a streaming file.

A Sorry, where do I say that?

Q In Paragraph 41. You say, "for example when joining a stream partway through or when rewinding or fast forwarding."

The last clause of the first sentence of the Paragraph 41.

A Okay. Could you give me the question again, because I think you maybe slightly mischaracterized what I said.

Q Yeah, I'm sorry. And if I did, I did not mean to.

So you state that the '321 Patent relates to an improvement in video coding that enables a decoder to identify where the start of decoding process, such as when they

randomly select a frame to start viewing video. And an example of when they would -- a user would randomly start viewing video, you give two here; would be when they joined a stream partway through, or when they rewind or fast forward.

Do you agree?

A Those are the examples I gave, yes.

Q What is your definition of video streaming?

A Sorry. Definition of what?

Q Video streaming.

A I don't recall giving a definition of video streaming. I think it's a term that would be known to a person of ordinary skill in the art, but I don't recall defining it here.

Q What would -- sitting here today, what would your definition of video streaming be?

A So I think a POSITA at the time of invention -- yeah, known services by then, and I think I can think of a few names. And

certainly nowadays I think we're all familiar with video streaming. So an example might be a service provider such Netflix or a provider like that, who provides video streams, makes them available across the internet. And the other side, if you'd like, is a user under certain conditions can access that service and can view videos on a device that was provided by that service provider.

And the underlying technology includes the internet. It includes, typically, encoding videos into formats that can, then, be provided through the internet or streamed, I think is like a colloquial word for that. And it includes receiving such streams a user device. Decoding a video, displaying a video to the user.

So I think that -- yeah, it's not a definition but that's the kind of thing that a person of the ordinary skill might understand by "video streaming."

Q In your opinion, would -- is there any

limitation of how many frames per second are required for something to constitute video streaming?

MR. UDICK: Objection to form.

A I don't recall placing a definition on this. I do recall maybe discussing that with respect to the Kim reference. I don't remember.

Q It would be Paragraph 79, if you --

A Thanks.

Q No problem.

A Yeah, so as I say in Paragraph 79, it's my understanding that Kim is concerned with intermittently recorded image data, i.e., single-image frames spaced at intermittent time intervals. And I think Dr. Havlicek agrees with me that this is not typical, 30 frame a second consumer video. And that's not necessarily a limit or a bound. But that's what I think -- what Dr. Havlicek was referring to there was the user experience, perhaps the user experience video streaming

using a service such as YouTube or Netflix where moving video is provided to user.

And that would typically be a number of frames per second, 30 is like a well-known example.

Q Again, when you use the term "video streaming," for example, in Paragraph 79 of your second report or declaration, do you place any bound or limits on the number of frames per second that are required?

A So I think one of my starting points for that paragraph was the '321 Patent itself, that's Exhibit 1001. And the inventor or inventors, is it inventors? Not just one inventor.

The inventor describes the field of the invention as being particularly in connection with streaming. And then there's a sort of a definition or an overview of streaming in the following paragraph beginning at Column 1 -- sorry -- yeah, Column 1, Line 19 of the '321 Patent.

The term streaming refers to simultaneous sending and playback of data, typically multimedia data in which the recipient may begin data playback already before all the data has to be transmitted has been received. So I think that's consistent with the high level overview of streaming that I just gave you. And in my view, and I think also in Dr. Havlicek's view, that is not consistent with reproducing the intermittently recorded image data on something like an optical disc.

Q And I don't know if that quite answered my question, but when you use the term "video streaming," for example, in Paragraph 79 of your second declaration, are you placing any bounds or limits on the number of frames per second that are required?

A I think I'm using the term "video streaming" there in a similar or consistent way with the way the inventor of the '321 Patent uses it.

So simultaneously sending and playing back data, such as multimedia data, across -- typically across a network. So the inventor doesn't, I see it put bounds on that, I was giving examples, Column 1 of the '321 Patent, Line 57, talking about video files in this kind of scenario, typically 15 to 30 frames per second, nowadays it would be often higher than that, to create an impression of a moving image.

So I think Dr. Havlicek's -- sorry, Havlicek's definition of examples that were consistent with what I understand about video streaming, or the way I tend to use the phrase video streaming, and I think consistent with the way I used it in Paragraph 79.

Q So would you put a lower bound on 15 frames per second for video streaming, given the disclosure of the '321 Patent?

MR. UDICK: Object to the form.

A I don't see the disclosure of the '321

Patent putting a lower bound on it, but it does say that the inventor does write that these types of video files create an impression of, I assume when decoded and displayed, create an impression of a moving image. So I think that is -- certainly gives the reader some context for the invention of the '321 Patent.

Q Okay. But there's no lower bound that you would put on the video streaming outside of the examples given in the '321 Patent; is that right?

A Well, I think what I'm doing in Paragraph 79 is comparing a system such as the one described by the inventor of the patent and, you know, this gives some context to the patent, and a system such as Kim, that intermittently records and then displays individual images. I forget exactly how Kim describes it, but that's my summary of Kim.

So in my mind, the system of Kim isn't consistent with what I think a POSITA would

understand to be video streaming, or this understanding in the patent, the '321 Patent, that when these -- such video files are decoded, they create an impression of a moving image.

So Kim's method doesn't, as I understand it, create an impression of a moving image. It allows the reproduction of an intermittently recorded frame.

Q So your definition of video streaming would be the minimal number of frames per second required to create the impression of a moving image?

MR. UDICK: Objection to form.

A I haven't given that definition in this matter, that I recall.

Q I'm just -- sitting here today, would your definition of video streaming be the minimal number of frames required per second to create the impression of a moving image?

A I don't recall being asked to give a definition of video streaming in this matter.

And I hesitate to do it on the fly without thinking about it in the context of the issues involved in this matter and documents and so on.

But I stand by what I say in my Paragraph 79, and what the '321 Patent describes and what the Kim patent, the '351 Patent describes. It's my opinion that they're different. I think Dr. Havlicek agrees with me. That's the point that I'm making.

Kim describes a system or method for recording and reproducing intermittently stored or intermittently captured image frames. And I don't recall Kim anywhere talking about the streaming, and I don't think a POSITA would consider that to be what's generally known to be video streaming. That's the point I'm making in Paragraph 79 of my second declaration.

Q If you can turn to Claim 8 of the '321 Patent, which would be on the last page,

starting in the bottom of Column 23.
A Okay.
Q Is there anything in Claim 8 that relates to the frame rate required?
A So my understanding is that -- I'm not a patent lawyer, but I've read a few patents in my time. And my understanding is that when you read the claim of a patent, you read it in the context of the whole patent.
So I don't see the words "frame rate" in Claim 8 of the '321 Patent, but that doesn't mean that it's not in the context of video coding or in the context of grouping the multimedia files, particularly video files, and particularly in connection with streaming, which is described as the field of invention of the '321 Patent.
Q I believe you testified earlier that you've been retained as an expert in quite a few patent litigations; is that right?
A Over the years, yes.
Q And during your time as an expert in

patent litigation you've worked on over the years, do you understand that it's improper to import limitations from a specification into a claim?

MR. UDICK: Objection. Calls for a legal conclusion.

A I don't know what that even means.

Q Or do you have any patents in -- I'm sorry. Let me ask a different question.

Are you a named inventor on any patents?

A I am, yes.

Q How many? If you can ballpark.

A I think it's 11 -- sorry, 11 patents or patent applications. I think I've listed the ones that I recall in the last page of my CV, which is at the end of Exhibit 2001.

Q So you have a general understanding of how the patent -- that patent has different portions or specification the claims, maybe a background section abstract.

Would you agree?

A I recall seeing such sections in patents

that I've been involved with over the years.
Q Do you recall that the specification could discuss certain preferred embodiments of the invention described in the patent?
MR. UDICK: I'll object. And I'll just caution the witness that in the process of filling out the patent applications, if any of your answers may rely on counsel communication at that time, do not disclose that, but, otherwise, you can answer.
A Okay. I recall reviewing patent specifications and seeing descriptions of the embodiments. And possibly the term "preferred embodiments". I'm not sure.
Q If you look at Column 4, Line 13, of the '321 Patent. Do you sign the term "preferred embodiments"?
A Okay. Yes.
Q So a patent specification will discuss preferred embodiments. Do you agree?
MR. UDICK: Objection to form.

A I don't know. Sorry for jumping in, but I don't know.
Q Okay.
A This specification discusses preferred embodiments.
Q If you look at -- turn back to Claim 8.
Does it -- did the claim limitations include any requirement of a minimal frame rate?
A I don't see minimal frame, those words in the limitations of Claim 8 of the '321 Patent.
Q Do you see any limitations that would require a minimal frame rate, even if it doesn't use those exact words?
A Okay. I think I'm now in -- outside my domain of expertise, which is a technical one.
So my understanding is that one reads a claim in the context of the patent, but beyond that, I don't have -- I don't think I have enough knowledge to understand or to

answer your question.
Q Okay. Limiting it this -- my next question, then, do you see the words "video streaming" anywhere in Claim 8?
A So I don't see the words "video streaming" in -- in the text of Claim 8.
Q It would be outside your expertise to opine on whether Claim 8 requires video streaming, unless it uses those exact words?
A I didn't --
MR. UDICK: Objection to form.
A I didn't say that, you know, certainly it's within my expertise to provide opinions relating to video coding and to video streaming, because I've got a lot of experience in both of those technical fields.
In terms of the exact requirements of patent law, other than the ones that were explained to me by counsel, I'm not a patent lawyer.
Q So, based on your expertise in video coding and video streaming, does Claim 8

require a decoder that decodes video streaming?

MR. UDICK: Objection to form.

A Claim 8 is what it is. It's a method for decoding a compressed video sequence. The method comprised a number of elements and this decoding from the video sequence and so forth. We talked through them earlier.

It seems quite clear to me -- yeah. I'm not quite sure what you're asking me. Either something I've given an opinion on or that's a question related to technically what is required here.

Q Well, I guess you say in Paragraph 79 that Kim does not relate to video streaming. And I just want to know whether Claim 8 requires video streaming?

A I don't state that Claim 8 requires video streaming. And I'm reluctant to give -- I haven't already given an opinion on that. I'm reluctant to give it on the fly today.

Q Okay. No problem. Thank you.

Turning back to our discussion a little while ago, it's your opinion that the '321 Patent relates to an improvement in random access in video coding -- sorry. Let me ask a different question.

In your opinion, does the '321 Patent improve random access in video coding?

MR. UDICK: Objection to form.

A Yeah. In my opinion, the '321 Patent is directed to improving random access in video coding.

Q How does it improve random access in video coding?

A Okay. You've changed the words in my answer.

Q Sorry. How does the '321 Patent improve random access in video coding?

A Okay. I said it was directed to improving random access. So the '321 Patent is just a document. So I'm not quite sure what your question is.

Q How does it teach improvements to random access in video coding?
A Okay. So, for example, as I explained in Paragraph 42 of my second declaration, and I'm quoting from the '321 Patent there, I won't read out all of it, but it identifies a problem -- an example of a problem in image numbering when the browsing of a video file is resumed from a random point after previous browsing. And a problem that can affect a decoder, and my recollection is that is an example of a problem that the '321 Patent teaches or is directed to avoiding or -- yeah. Avoiding or solving, such as I summarize in Paragraph 43 of my second declaration.
Q Okay. If you look back at Paragraph 41, the second sentence of Paragraph 41, in that sentence, you state:
"This is known as random access in video coding which the '321 Patent is directed to improving."

Do you see that?
A Yes.
Q And that sentence is followed by the three citations to the '321 Patent.
Do you agree?
A Yes.
Q If you can turn to those citations, starting with the first one, which is the '321 Patent, at Column 3, Lines 52 to 56.
Do you agree?
A Okay. Yes.
Q And that citation appears in the background section, the background of the invention section of '321 Patent.
Would you agree?
A Yes.
Q What does the '321 Patent state at that section, Column 3, Lines 52 to 56?
A Do you want me to read it out or --
Q Sure. That'd be great.
A Line 52, there's a partial sentence, I'll start at the full sentence at the end of

that line.

"Since it is possible to also select at least one image preceding an I Image, it begins a group of pictures, GOP as the reference image. The group of pictures employing reference picture selection cannot necessarily be decoded independently."

Q Is that a description of how the '321 Patent is directed to improving random access in video coding?

MR. UDICK: Objection to form.

A My recollection is that obviously it's a sentence in the context of quite a long section entitled background of invention. My recollection is that this is an example of aspects of a problem the inventors identified with random access prior to the invention.

Q And if you look back at Paragraph 41 of your declaration, your second declaration, the next cite you give is Column 3, Line 62 to Column 4, Line 4.

Do you see that?
A Yes.
Q And to save Kiara's fingers, I won't make you read that on the record, but if you could read that to yourself and let me know when you're done.
A Okay.
Q Would you agree that section also describes a problem the inventors identify with random access prior to the invention?
A Generally, yes, you could argue, or I think it would be reasonable to say that this could be discussing multiple problems, such as one, two, three different scenarios in which, as I recall, inventors identify problems can occur.
Q If you turn back to Paragraph 41, the last citation for the '321 Patent, that you give four -- or Column 4, Lines 48 through 60.
Do you see that?
A Yes.

Q And if you can turn to that in the '321 Patent, and you can read it to yourself and let me know when you're done.
A Okay.
Q In that section, there's a paragraph in Column 4 of the '321 Patent, and it starts with:

> "An advantage of the procedure or the invention is that it enables to start browsing a video sequence from a random point, i.e., it provides the decoder with information about the first picture of an independently decodable sequence."

Do you see that?
A Um-hmm. Yes. Sorry, yes.
Q When it says it provides the decoder information about the first picture of an independently decoded sequence, what is the "it" there? What is the "it" referring to, I guess would be a better question?
A I think reading this in isolation, but I

think that "it," the article it, refers to the procedure of the invention. I mean, the paragraph is talking about an advantage of the procedure of the invention, but I think -- I think the article, it would seem to reflect back to the procedure of the invention.

So, yeah, it would be reasonable to say the procedure of the invention provides the decoder with the information about the first picture of an independently decodable sequence.

Q Would the encoder provide information about the first picture of an independently decodable sequence to the decoder?

MR. UDICK: Objection to form.

A In what context?

Q In context of the '321 Patent?

A Sorry, it's a big patent, so can you give me some context.

Q Well, you talked about the procedure of the '321 Patent. In that procedure would it

be an encoder that provides the decoder with information about the first picture of an independently decodable sequence?

MR. UDICK: Objection to form.

A So this is just a brief -- in the section entitled brief description of the invention and the sentence you asked me to look at talks about the procedure of the invention, but the invention described throughout the patent, generally that's my, again, not a patent lawyer, but generally need to read the whole thing in the context. So we can go to embodiment, so we can go to claims, but it's a little bit difficult for me to give an answer just based on that one sentence.

Q Okay. You can look back at your declaration, please. In looking down in Paragraph 42. In Paragraph 42, you discuss a problem where discontinuity and image numbering occurs, do you see that?

A Yes. More or less, yes.

Q And part of the problem is that the decoder can't interpret this discontinuity as an unintentional loss of image frames.

Do you see that?

A That's what the inventors say.

Q And in Paragraph 43, you go on to discuss the '321 Patent and where it describes that the discontinuity problem can be avoided by the decoder to finding an initiation image in a group of pictures and numbering the initiation image to 0.

Do you see that?

A I don't think you read that correctly.

Q Okay.

What corrections do you have with that?

A I think you said can be avoided by the decoder or something like that.

Q Okay.

A Or, yeah, could you read it again, I want to be, I know this is important, I want to be absolutely precise.

Q Sure. In Paragraph 43 you state the

patent and I assume you mean the '321 Patent there. Explains that "This bracket problem can be avoided in the decoder by defining an initiation image in an independently decodable group of pictures, GOP, that is activated at random points of the video file and number of initiation image set to 0".

Do you see that?

A Yes.

Q It continues "Consequently, when activated at a random point, the decoder preferably sets the identifier as the first image, preferably an I frame, of the independently decodable sub sequence."

Do you see that?

A Yeah, I think you missed out a word frame, but the decoder preferably sets the identifier of the first image frame preferably an I frame of the independently decodable sub sequence.

Q My apologies about that. Thank you for catching it.

When the decoder as being described here -- or sorry, when the video sequence is activated at a random point, the decoder is setting the identifier of the image frame; is that correct?

MR. UDICK: Objection to form.

A Say that again, please.

Q When the video sequence is activated at a random point, the decoder is setting the identifier value of the first image frame, do you agree?

MR. UDICK: Objection to form.

A What do you mean by the video sequence is activated at a random point.

Q So it talks about, Paragraph 43 quoting the patent -- defining -- "discusses defining an initiation image, an independently decodable group of pictures that is activated at a random point of the video file."

Do you see that?

A Yes.

Q Then it says, "When activated at a

random point the decoder preferably sets the identifier of the first image frame."

Do you see that?

A Yes.

Q So it's -- you're saying -- the patent is stating here that the decoder is setting the identifier value of the first image frame; is that right?

A It's pretty much what the patent says. In Column 11 Line 38, it says, "The decoder preferably sets the identifier of the first image frame." And then I'll skip a bit. "At 0" paraphrasing it says "Decoder preferably sets the identifier at 0."

Q What does the term set the identifier mean?

MR. UDICK: Objection to form.

A I think it means what it means. I think it's clear in the context of the patent that it's, yeah, setting an identifier, sorry, setting an identifier to 0 is setting the value of an identifier to 0 in general and I

think that's consistent with what's being described here. This particular sentences says sets the identifier at 0, but my recollection is that within the context of the patent that's consistent with setting something to 0.

Q Is the identifier value part of the bitstream that the decoder is decoding?

A Where does it say that?

Q I'm asking the question. I'm just asking the question if it does?

A I don't see this passage in Column 11 in the '321 Patent saying this identifier is part of the bitstream.

Q Is another term for bitstream a compressed video sequence?

A In the context that we're discussing today, I think that would be reasonable.

Q Is the identifier value referenced in the '321 Patent part of that compressed video sequence?

MR. UDICK: Objection to form.

A So identifier and identifier value, I think are used multiple times in the patent, so where do you want me to look.
Q You can look at the Claim 8, that would be helpful?
A Okay. What's the question.
Q Is the identifier value referenced in the '321 Patent part of the compressed video sequence?
A I see there's an element, claim element, I think we called that element 8C earlier, decoding identifiers, sorry, say it again, decoding identifier values for image frames according to a numbering scheme. I don't see in that element of the claim a requirement that such identifier values are part of a video bitstream or whatever you asked.
Q If you look at the preamble of Claim 8, does it recite a decoder decoding a compressed video or a compressed video sequence?
A It recites a method for decoding a

compressed video sequence.
Q Would you understand that the identifier value recited in limitation 8C that you just read is part of that compressed video sequence?
A What do you mean by part of that compressed video sequence.
Q Is the identifier value included in the compressed video sequence?
A Do you mean as a specific syntax element or do you mean something else.
Q Specific syntax element.
A I don't see such a requirement that the identifier value has to be included as a specific -- say an individual syntax element.
Q If it doesn't have to be included as a specific syntax element, then what does the term decoding identifier values mean?
A Just what it says, I think it's -- I don't think it's ambiguous. I mean, for example, this is a method for decoding a compressed video sequence, so in this context

a person of the ordinary skill reading this claim I think would understand that the method involves decoding video frames. It doesn't mean that each video frame is an individual single syntax element in the bitstream.

Q Would decoding those video frames include decoding identifier values for those video frames?

A That's kind of what that element, roughly what the element says.

Q If the element says it's decoding identifier values, what is it decoding them from?

A It's overall just -- again, consider video frames. If an MPEG-1 decoder is decoding video frames it is decoding video frames essentially from the bitstream. But each video frame in that example is decoded by following the processes or the process of the MPEG-1 standard and that process, as I recall, decodes a whole number of syntax

elements and processes them in a certain way to result in decoding of the video frames. So the information required to decode the video frame is in that example, a series of syntax elements in the video bitstream, but in that example, yeah, that's a completely consistent example a POSITA would be familiar with, where a bitstream is a decoded, the bitstream comprises syntax elements, the result of that decoding is that video frames are decoded. That's kind of the language in the context that we're talking about here.

Q Looking back at Claim 8 the method recited, you don't believe the identifier values are part of the compressed video sequence that is being decoded?

MR. UDICK: Objection to form.

A I did not say that and I don't recall stating that opinion.

Q So earlier I asked if you would understand the identifier value cited in limitation 8C to be part of the compressed

video sequence. You asked for clarification about whether I meant as a specific syntax element or whether I meant something else. I clarified that I meant a specific syntax element and you stated, I don't see such a requirement. This identifier value has to be included as a specific, say an individual syntax element.

Do you agree with what you testified to earlier?

MR. UDICK: Objection to form.

A I said what I said in response your question. To be honest, I'm not going -- I don't want to try and reanswer the same question.

Q Understood.

A I'll just say as with every question, I'm attempting to answer to the best of my ability based on what's in front of me.

Q Understood, and I appreciate that.

A I wonder is this a good time for a break.

MR. KENDRICK: Yeah. That's fine.

(Whereupon, a recess was taken from 10:35 AM until 10:45 AM.)

Q Welcome back, Dr. Richardson.

During our break, did you discuss the substance of your testimony with counsel?

A No.

Q Thank you.

If you look back at Claim 8 and limitation 8D, it says, "Resetting the identifier value for the indicated first image frame of the independent sequence."

Do you see that?

A Yes.

Q Is the decoder resetting that identifier value?

A To Claim 8 we talked about the preamble part method for decoding, so I guess reading this as a technical person I would see it as -- I'm not making any statements about the preamble or any legal statements, but it would seem to be consistent with, you know,

something that's executing a method for decoding, so that could be a decoder. Sorry, when I say it could be consistent with, you know, the stacks, or yeah, the items, the things that are recited in the elements of the claims seem to be consistent to me with what a decoder or decoding process or something implementing a method for decoding might do.

Q And that -- those -- and the identifier value is decoded from the compressed video sequence in limitation 8C?

A Which identified value?

Q Well, you see resetting, it's resetting the identifier value.

Do you see that?

A So element 8D is resetting the identifier value for the indicated first image frame of the independent sequence.

Q Would the identifier value for the indicated first image frame of the independent sequence be decoded from the

compressed video stream in limitation 8C?
A Limitation 8C says what it says, decoding identifier values for image frames according to a numbering scheme.
Q Okay. Would the identifier values for image frames include the identifier value for the indicated first image frame?
A I don't recall being asked to form an opinion on that. I'm not sure one bit -- yeah, I haven't actually thought about whether you can satisfy the claim without that being true, I don't know.
Q So you aren't able to answer -- you don't know whether identifier values as referred to in limitation 8C would include the identifier value for the first image frame?
A Would or could?
Q Would.
A I haven't formed an opinion on that and I'm kind of reluctant to do so on the fly. I mean, I could take my time and think about

it, but I don't recall thinking about that or trying to.

Q Sitting here right now, you don't have an opinion as to whether the identifier values for images frames includes the identifier value for the indicated first image frame?

MR. UDICK: Objection to form.

A What do you mean by the identifier values for image frames.

Q I'm reading from limitation 8C, which says, decoding identifier values for image frames. So sitting here right now, you don't have an opinion as to whether identifier values for image frames as recited in limitation 8C includes the identifier value for the indicated first image frame that's recited in limitation 8D; is that right?

MR. UDICK: Objection to form.

A I don't see those words in the claim and I don't recall being asked to provide an opinion on that question, and I don't recall

offering an opinion on that question.
Q So you don't see the term identifier values for image frames in limitation 8C?
A What do you mean I don't see the term.
Q You said you don't recall seeing those words in the claim?
A Yeah, you asked me -- yeah, those words I think I meant whatever form to your question was you talked about one including the other.
Q Okay. So limitation 8C states "Identifier values for image frames."
Do you see that?
A That's part of the limitation.
Q But you see that term, it's included in the limitation?
A Okay. How is a term different from a limitation.
Q Well, a limitation has -- this limitation says, decoding identifier values for image frames according to a numbering scheme. A term of that limitation is

identifier values for image frames. Do you see that term?
A Okay. I see that phrase, and I don't recall whether that's a -- whether a term is term of art for in patent --
Q No --
A I see the phrase within that -- I know that's a limitation. I know that much. I see that phrase identifier values for image frames.
Q Would that identifier values for image frames include the identifier value for the indicated first image frame?
A Well, they're two separate elements, so my general understanding is that claim has a number of elements, you can have a number of elements. To satisfy the claim, you have to satisfy all the elements. So are you asking me -- yeah.
You seem to be asking me whether it's required in the claim, or whether to satisfy the claim the identifier value for the

indicated first in each frame and the last element has to be one of the decoder identifier values in the previous element. Is that what you're asking me, or something else?

Q Yes.

A Okay. I don't see the claim stating that. I haven't thought through whether it's -- in the context of the patent whether it's a logical or necessary requirement that the identifier value in the final limitation, Limitation D, is an identifier value that has been decoded in the previous element. I'm not sure.

Q So you don't have an opinion on that?

MR. UDICK: Objection to form.

A I think I already said I haven't been asked, as I recall, to give an opinion on that, and I don't recall expressing an opinion on that in either of the declarations. And I'm reluctant to kind form an opinion -- if I haven't expressed it

before, I'm reluctant to form an opinion on that.

I think you're asking me how to satisfy the claim, and I've been asked to give opinions on the validity of the claims.

Q Well, I'm just asking -- I'm asking this question.

So the claim recites that a method for decoding a compressed video sequence, the claim later recites decoding identifier values for image frames.

My question is: Whether decoding the identifier value for image frames would include the identifier value for the first image frame?

A I just don't have an opinion as to whether that's required by the claim or not.

Q Okay. Do you --

A I would have to think about that and consider the claim, again, in the context of the patent.

Q I apologize for interrupting.

A Yeah, that's okay.
Q Do you have an opinion on whether the identifier values that are decoded in Limitation 8C are part of the compressed video sequence?
A So Claim 8 is a method for decoding a compressed video sequence, the method comprised, in reading from -- I think it's the preamble. And then one of the elements is decoding identifier values for image frame. So I think it's consistent that with a method for decoding a compressed video sequence that the identifier values are decoded as part of decoding the compressed video sequence.
Sorry. I've forgotten the exact question that you asked me.
Q I think you answered it, thank you.
A Okay.
Q And as part of that decoding of identifier values -- sorry, strike that.
And as part of decoding those identifier

values, the decoder would decode the value of that syntax element; is that right?
A What do you mean by "that syntax element"?
Q Do you agree that identifier values, as used in claim 8, are -- is a syntax element?
A I don't see where the claim says that.
Q Okay. When the identifier value for the first indicated image frame is reset, is it reset to a value that's different than the value that was decoded in the limitation 8C?
MR. UDICK: Objection to form.
Incomplete hypothetical.
A I think we've already discussed that I haven't -- I don't have an opinion on whether that identifier value was one of the identifier values that was decoded. It just says resetting the identifier value.
Q Looking in Claim 8, do you see the terms "random access" anywhere in Claim 8?
A I don't see the words "random access" in the text of Claim 8 of the '321 Patent.

Q If you can turn back to your declaration, and starting at the bottom of Paragraph 45, which spans Pages 14 and 15.
A Okay.
Q And so you have an example of image frames with a car driving through those image frames.
Do you see that?
A Yeah, it's like a really -- it's a graphic, if you like. They're not intended to be realistic image frames.
Q Yeah, understood.
And in paragraph -- below Paragraph 46, you show that part of -- you're showing a Picture 0 that contain video content for part of Picture 1.
Do you see that?
A Sorry. Could you ask that again, please?
Q Yeah. If you look at the image below Paragraph 46, it shows that Picture 0 contains video content that is necessary for

Picture 1.

Do you see that?

A Oh, yeah. That's kind of broadly what I'm trying to illustrate there.

Q Okay. And you refer to this concept as temporal prediction; is that right, in Paragraph 48?

A Generally, yes.

Q Okay. And then in Paragraph 48 you state that decoding errors can occur when a reference picture, which in your example is Picture 0, is missing; is that right?

A Yes.

Q What is a reference picture?

A So I think I was using this as a very high-level description this part -- explanation, this part of my declaration. I think here I'm using it in kind of a broad manner, such as a picture or a coded frame that other coded pictures of frames use or rely on in order to construct predictions.

Q In Paragraph 48, you state decoding

errors occur when the reference picture, parentheses, Picture 0 is missing, and the prediction pictures, parentheses, Pictures 1 through 5 cannot reference that video.

Do you see that?

A Yes.

Q Okay. What decoding errors occur?

MR. UDICK: Objection to form.

A So if a portion of Picture 1 in the diagram, at my -- below my Paragraph 48, if a portion of Picture 1 in that diagram -- if in order to decode that portion the decoder has to make use of picture data, such as pixel data, from Picture 0 to construct a prediction, then decode that portion. If Picture 0 is missing, then a decoder is unlikely to be able to successfully decode information in that area that was predicted from Picture 0.

Q And then in the image below Paragraph 48, you have an arrow pointing from the top box of Picture 1 to Picture 0; is

that right?
A Yes.
Q Okay. And is that showing a dependency?
A It's illustrating dependency, yes, at a very, very high level.
Q Can Picture 1, which depends on picture 0 be decoded if Picture 0 is missing?
MR. UDICK: Objection to form.
A So a picture can be decoded. It won't be -- in this example, it will be a different picture from the picture that would have been decoded if Picture 0 was present and was an available decoder. In this kind of scenario, which does happen, it's up to the decoder implementation on how to proceed. But if the decoder implementation attempts to create an output frame, to create a decoded frame, it might be able to create something. But it's -- in this scenario, unlikely to be the same reproduction that the decoder had received from Frame 0 and decoder Frame 0 would be able to create.

And from an output perspective, the user might see a missing area or they might see an area that has some obvious facial distortions, for example, depending on how the decoder handles that.

Q What would happen if Picture 1 was displayed before Picture 0 was decoded?

MR. UDICK: Objection to form.

Incomplete hypothetical.

A So in a way, that's kind of what I'm illustrating here. So Picture 0 was not successfully decoded and there are a number of reason why that could happen, a packet loss or bit errors that caused the decoder to simply fail to decode that picture.

If the decoder choses to attempt to decode Picture 1 without Picture 0 being available, it could choose, and this is a design choice, not something specified in a standard, such as MPEG-1. The decoder could choose to attempt to decode Picture 1 and to display something to the user. So, for

example, here, I've shown displaying a partial picture. And this is illustrating a scenario where part of Picture 1 was put into Picture 0, part was not. So this is the kind of thing that would happen.

Q So Picture 1 would not be able to be fully displayed because it would be missing a part of the picture that depends on Picture 0?

A I'm not sure I would say "fully displayed." With this type of error, and this type of scenario, if a decoder attempted to decode Picture 1, it might be able to decode something, some form of a picture. In this example I've tried to illustrate that the decoder has not been able to successfully decode all of the portions of Picture 1. So in a way, it has provided in the example a representation of Picture 1, but it's a representation in this example with a pretty obvious error region.

Q And your declaration continues on in

Paragraph 50, where you expand on the challenge of random access. And you state that, "Dependencies present challenges with random access because the decoder does not know from which image frames it can begin decoding."

Do you see that?

A Did you say Paragraph 50?

Q Yes. Last sentence in Paragraph 50.

A Okay, yes.

I see that.

Q Okay. And then, if you can turn back to the example you give above Paragraph 50, with the cars. Picture 5, in that example, depends on Picture 0.

Do you agree?

A I think I'm trying to illustrate there that Picture 5 is -- yeah, it essentially follows on from Paragraph 49, I'm attempting to illustrate that Picture 5 is predicted from Picture 0. So part or all of the data in Picture 5 is in the bit, in terms of the

way the bitstream is constructed, relies on Picture 0 to form predictions, or a prediction or predictions.

And so, yeah, Picture 5 is, in this illustration, predicted from Picture 0.

Q Okay. So Picture 0 would have to be decoded before Picture 5 can be displayed; is that right?

MR. UDICK: Objection to form.

A So for -- let's say error free display, and, you know, by error free, it's sort of a term I coined. It's kind of a vague term, but let's compare it to a situation where all of the pictures were available and decoded to, let's say, a standard such as MPEG-1. If Picture 5 depends on Picture 0, then Picture 5 cannot be completely decoded in an error free manner without the data corresponding to Picture 0. I'm not sure if that's what you asked or not.

Q My question was more of whether Picture 0 would have to be decoded before

Picture 5 can be displayed?

MR. UDICK: Object to form.

A Okay. So in a decoder that conforms to MPEG-1, for example, MPEG-1 specifies a certain decoding order and certain display orders, which I think I talk about in my declaration.

So to satisfy those orders, certain things have to be decoded in certain order before certain pictures can be displayed. It doesn't prohibit a decoder from attempting to do something else. For example, if there's an error, a decoder can attempt to handle the error, but then it's -- one could say it's working beyond what the standard might specify.

Q So Picture 5 here could not be decoded without errors if Picture 1 was unavailable?

MR. UDICK: Objection. Form.

Incomplete hypothetical.

A Picture 0?

Q Sorry. Let me rephrase that question.

So if Picture 0 is not previously decoded in this example under Paragraph 49, a decoder trying to decode Picture 5 will not be able to do so without errors; is that right?

MR. UDICK: Same objections.

A So in this example, if Picture 5 is predicted from some of the -- if some of the data in Picture 5 or some of the blocks, macroblocks, however it's constructed, and imagining a standard such as MPEG-1 and a decoder conforming to that standard, if Picture 5, it blocks -- the data within Picture 5 is predicted from Picture 0, then to construct those predictions in the way that they intended -- yeah, the way that the standard says, the decoder has to decoded Picture 0.

Again, in the -- thinking of the MPEG-1 context where prediction is something that's done from decoded previously received and stored reference frames.

Q Okay. Looking at the -- so there's an -- another example of image frames below Paragraph 51.

Do you see that?

A Yep, yes.

Q And those images frames, there's an Image Frame 0 through 5 as part of that example?

A Yes.

Q Do those correspond to the Image Frames 0 through 5 below Paragraph 48, for example?

A How do you mean "correspond"?

Q Like -- so you see an image. There's an Image Frame 0 in the example below Paragraph 51 and an Image Frame 0 in the example above Paragraph 48.

Do you see that?

A Yes.

Q Okay. They both look to be the same image; is that right?

A Um, I think that's my recollection.

Q Okay. Same thing for Images 1, 2, 3, 4

and five, do you agree?
A I think that's what I was intending to illustrate. Again, it's just an illustration, very high level -- could be, like, a schematic illustration.
Q Understood.
Looking back at the example below Paragraph 51. Image -- or Picture 3 is a P-frame or would be a P-frame, right?
A Why do you say that?
Q Because if you look up at paragraph -- where the -- your dependency arrows around Paragraph 48, it shows Picture 3 depending from Picture 0.
A That's a different diagram.
Q So you -- they're not the same illustration?
A Clearly, they're not the same illustration. I thought you were asking me about whether the diagram above Paragraph 50 and the diagram above Paragraph 52 were illustrating the same basic sequence. And I

think my recollection was they were. And now you're asking me -- yeah, what are you asking me?

Q Looking at the example below Paragraph 51, you have a red arrow to image three or Picture 3.

Do you see that?

A Yes.

Q Is Picture 3 an I-frame or a P-frame?

A It's just a diagram. I don't think I intended to illustrate a particular frame type.

Q What about Picture 5, would it be an I-frame or a P-frame?

A I think there I'm illustrating a frame that has some prediction dependency on Picture 0 in this illustration. I don't think I was intending to exclude the possibility that five could have other prediction dependencies.

Q What about picture four, would it be an I-frame or a P-frame?

A I don't think it makes much of a difference for the discussion of Picture 5, which is what I'm discussing. In Paragraph 51, I'm discussing Picture 3, Picture 0 and Picture 5.

Q Well, in Paragraph 51, you state that the user selects Picture 3 as a random access point.

Do you see that?

A Yes.

Q You say:

"The decoder starts decoding at Picture 3. An error occurs when the decoder gets to Picture 5 because Picture 0 is not previously decoded."

Do you see that?

A Yes.

Q So I'm asking, in order for the decoder to start decoding at Picture 3, Picture 3 cannot have any dependencies; is that right?

A Why do you say that?

Q Because if it had dependencies, then how

would it be decoding if the decoder started there?

MR. UDICK: Objection to form.

A If I implement a decoder according to the MPEG-1 standard, if I get a partial bitstream with no errors, nothing missing, and I start decoding at the beginning, then my decoder should be able to decode according to the MPEG-1 standard.

If I get a bitstream with an error or something missing, or if I make my -- I, as the designer, the decoder again attempt to start decoding whatever frame maybe a user choses, I can write software that reads the bitstream and attempts to start decoding.

I might not be able to get a displayable video frame at all or I might get video frames that have errors and we actually see this, if you're watching a streaming service and you get an error, usually the decoder attempts to keep going and you might see some disruption in the picture.

Yeah. I don't think I'm necessarily -- I don't think my simple illustrative example depends on what's in Picture 3, what type of coded data is in Picture 3. And it doesn't exclude the fact that Picture 3 may or may not be decodable without visual disturbances or errors.

Q Okay. So you are unsure if the error -- if you start the decoding at Picture 3, in the example below Paragraph 51, whether an error would occur before Picture 5?

MR. UDICK: Objection to form.

A I'm not unsure. This is not a coded video bitstream. It's just an illustrated diagram.

Q Well, it's illustrating a problem with random access as described in the '321 Patent, right?

A It's illustrative of a potential problem with random access.

Q How does it illustrate the challenges of random access?

A Paragraph 51. If the decoder starts decoding at Picture 3, an error occurs when decoding Picture 5 because Picture 0 was not previously decoded and so is not available as a decoder for predicting Picture 5.

So that's a problem. It's just one paragraph after a whole lot of paragraphs. And then I talk about a solution in the '321 Patent. Incorrect decoding of a picture is a problem from the user perspective.

Q And the image below Paragraph 51, you're representing that Picture 5 cannot be displayed because Picture 1 was not previously decoded, right?

MR. UDICK: Objection to form.

A I'm not saying anything about display. I'm just saying something about an error that occurs during decoding. I say what I say.

Q If you go to Paragraph 54, there's some additional images under there.

Do you see those?

A Yes.

Q Are those the same images that are below Paragraph 51?
A No.
Q How are they different?
A There's different color coding. There's different labels. There's different arrows. Different in a whole number of ways. There's no X.
Q But the images themselves with the little car driving through, are those different?
A Yes. One of them has got an X through it, in Paragraph 51. Which I think is pretty clear from the description.
Q Okay. But the images under Paragraph 54 are still showing that Picture 5 is dependent on Picture 0.

Would you agree?
A There's still an indication in the first of the diagrams below Paragraph 54 and my second declaration, that Picture 5 is dependent on Picture 0 in this illustration.

Q There's also an indication that Picture 5 is dependent on Picture 0 in the second illustration of Paragraph 52, correct?
A Yes.
Q And there's a red arrow in the first illustration, which you say -- which is identifying a user can select Picture 3 in group two as the random access point for the decoder to begin the video decoding sequence. Do you agree?
A That's illustrated over a user selecting a particular random access point.
Q Okay. And, then, you state in Paragraph 54 that decoding begins at the first image of one of the independent sequences either Picture 0 in group one or Picture 0 in group three, right?
A Yes.
Q Would the decoder described in the '321 Patent begin decoding either one of these pictures?
MR. UDICK: Objection to form.

Outside the scope.
A Which decoder described in the '321 Patent?
Q How about the method or claim claimed in Claim 8, would that method start decoding at either Picture 0 in group one or Picture 0 in group three?
MR. UDICK: Objection to form. Scope.
A How can the method start decoding?
Q Well, it's the method for decoding, right?
A Claim 8 recites a method for decoding.
Q How about Claim 10, which recites a video decoder, or would that video decoder be able to start at either Picture 0 in group one or Picture 0 in group three?
MR. UDICK: Objection to form. Scope.
A I don't know.
Q You don't have an opinion on that?
A I don't know what that video decoder is, apart from what the claim recites.
Q Is there a citation of Paragraph 54 of

the '321 Patent?
A No.
Q If decoding -- so you see in Paragraph 54, you describe decoding beginning at the first image frame of either Picture 0 in group one, or Picture 0 in group three, right?
A In my illustration for Paragraph 54, yes.
Q Okay. If the decoding begins at -- the user select Picture 3 to begin decoding, how does the decoder work backwards to start Picture 0 of independent group one?
MR. UDICK: Objection to form.
A It depends on the decoder.
Q Well, you describe a process for decoding in Paragraph 54, how would the decoder that you're referring to there work backwards?
A Okay. At the risk of stating the blindingly obvious, Paragraph 54 is a single sentence, a video decoder -- I've implemented

video decoders, they are typically thousands or tens of thousands of lines of software code or hundreds of thousands lines of hardware description code.

Figure 54 is not a video decoder, it's an illustration of a process. So if you're asking me how, in general, video decoders that I'm aware of can work out where to start decoding, there are multiple methods that they can do that.

Q What are one of those methods?

A So, for example, if I'm thinking about an MPEG-1 video decoder designed to read from a bitstream that is recorded on a CD, presumably as part of that system there's a mechanism for scanning backwards and forwards at the decoder, scanning the optical disc backwards and forwards.

MPEG-1, for example, specifies things called start codes, and I think I talk about this in my report, so I would hypothesize that the device scan backwards and forwards,

rebit, or maybe even start from the beginning and find a start code that's at the nearest start point before this point, or find a start point after this point. And I'm just sort of hypothesizing here, but I can imagine setting up a system to do that.

Q Starting more on MPEG-1. So in your second declaration, your discussion of MPEG-1 is kind of in two parts. There's a background section, or what I'll refer to as background section starting on Page 19.

Do you see that?

A Yes.

Q And that goes to Page 29. Would you agree?

A So on Pages 19 to 29, I discuss the MPEG-1 standard.

Q And you haven't all referred to as an argument section, starting on Page 36 of the MPEG-1. Do you agree?

A I'd rather not call it an argument section. You can call it something else.

Q Do you have a preference for what you would call it?
A Well, starting on Page 36, Paragraph 97 of my declaration -- well, actually starting on Page 36, at the title, I begin discussing the grounds and the petition.
Q So refer to that section as the grounds section? Is that okay?
A If you wish.
Q Okay. Looking back at your background section on MPEG-1. You agree that MPEG-1 doesn't specify an encoding process, but instead specifies syntax and semantics of the bitstream of a single processer decoder, right?
A I don't recall saying those exact words. I think that's broadly consistent with what I say there.
Q Okay. And you reference that the encoder can make decisions and chose options when encoding a bitstream, right?
A Yeah. I think that's -- you're reading

from my Paragraph 57 here, I recall I was talking about a video encoder that produces bitstream that complies with the MPEG-1 standard.

Q In that same paragraph, you state that the decoder processes the bitstream that it receives according to syntax specified in MPEG-1?

A It's not exactly what I say. I don't think it's inconsistent with what I say.

Q I'll take it as not inconsistent. That works with me. I'm a C student.

So you're familiar with the temporal reference in MPEG-1, correct?

A I'm familiar with the temporal_reference syntax element. Which has a value -- from which a value can be decoded. As I recall, that value can be temporal_reference.

Q You answered -- I think you have my next question, which is, the temporal_reference is a syntax element, correct?

A Temporal_reference is a syntax element

specified in MPEG-1.
Q The temporal_reference of the syntax element is included in the picture header. Do you agree?
I was going to say 38.
A Yeah. So stamped Page 38 of Exhibit 1005, which is the MPEG-1 video standard, there's a section of the specification type of picture layer, and within that picture layer, immediately after the picture start code or picture_ start_ code is the temporal_ reference syntax element.
Q So you would agree that the temporal_reference syntax element is in the picture layer?
A It's specified in the picture layer of the MPEG-1 syntax.
Q When an encoder is encoding an MPEG-1 compliant bitstream, it will encode a temporal_reference value for an image frame and that will be coded in a bitstream, right?

A A video encoder that produces a bitstream that conforms to MPEG-1, if that encoder encodes a picture layer as part of that bitstream, my understanding and recollection is that the syntax of the picture layer that's encoded should follow or be consistent with the syntax described in the section we discussed on Page 38 of this exhibit. And that syntax includes a temporal_reference syntax element.

Q And the bitstream will include a value for that temporal_reference syntax element, right?

A Which bitstream?

Q The bitstream that's being -- or that is produced from the MPEG-1 -- the MPEG-1 compliant bitstream that's produced from the encoder?

A Every bitstream or just a bitstream?

Q A bitstream.

A So if an encoder produces an MPEG-1 compliant bitstream, and if it's part of

that, it includes the picture layer, syntax. And, sorry, what was the question?

Q The bitstream will include a value for that temporal_reference syntax element?

A So if what I just said is true, and an encoder produces an MPEG-1 compliant bitstream that includes the picture layer syntax, that picture layer syntax, according to the standard, should include the temporal_reference syntax element, and that syntax element will have a value.

Q And an MPEG-1 compliant decoder will receive the value of that temporal_reference syntax element when it receives the bitstream, right?

A In what situation?

Q So the situation -- so we talk about the fact that the value will be encoded into a bitstream if a MPEG-1 compliant decoder receives that bitstream, will the temporal_reference value be included in that bitstream?

A If an MPEG-1 compliant decoder receives and decodes an MPEG-1 compliant bitstream that includes the picture layer syntax as part of that it will according to the standard, or should, according to the standard decode the temporal_reference syntax element and that decoding as I recall should include decoding the value of that temporal_reference syntax element.

Actually, it's an interesting question as to whether a decoder must decode that element.

So the temporal_reference syntax element is certainly part of the bitstream, the MPEG-1 bitstream, if the MPEG-1 bitstream includes a picture layer syntax.

Q And the picture layer contains all the coded information for one picture, right?

A Not necessarily.

Q MPEG-1 states on Page 87 that the picture layer contains all the coded information for one picture, right? Page 87

Section D.5.5.3. You can see that?
A Yes.
That's what it says, yeah, my confusion if you'd like, was that to successfully decode a coded picture I think an MPEG-1 decoder needs to or may need to also decode information from the GOP layer and the sequence header.
Q And the picture layer includes the temporal_reference syntax element, right?
A We've discussed that, yes.
Q And say the term picture here is used in that sentence from MPEG-1 on Page 87, would image frame be a synonym for picture here?
A So MPEG-1 defines picture on Page 25 of this Exhibit, it's quite a long definition and it says, picture --
Q Sorry --
A Sorry, this is relevant. I'm going to give you my answer.
Q Okay.
A And it says, picture, and I'll

paraphrase it just to keep it short, but picture is defined as a source coded or reconstructed image data. So that's how the standard defines it.

Q Great. Thanks. I just want to save Kiara's fingers if we could.

A Yeah, but I think it's relevant to answer your question.

Q Understood. So if a picture has a temporal_reference of 0, will the decoder decode that temporal_reference value and understand that it's associated with the first image frame and display order of a GOP?

MR. UDICK: Objection to form.

A Not necessarily.

Q When would temporal_reference value of 0 not indicate the first image frame display order of a GOP?

A Well, if you look at Page 87 of the MPEG-1 standard Figure D.20 shows an example of a group of pictures.

There's one pretty much in the middle of

that figure that has a temporal_reference of 0 and it's not the first picture in the display order of a group of pictures.

Q If you assume that a group of pictures has less than 200 or 1,023 pictures, would a temporal_reference value indicate the first image frame and display order of this group of pictures?

A Indicate to what or who.

Q Indicate to the decoder that the image frame or that picture is the first in display order of the group of pictures?

A Okay. How does the decoder make that assumption?

Q Well, temporal_reference value of 0. Assume -- let's for the sake of argument assume that a group of pictures has less than 1,023 image frames.

A Okay.

Q A temporal_reference of 0 -- a temporal_reference indicates the display order of those image frames for pictures,

right?
A A temporal_reference, I'm reading from Page 87, is a ten bit number which can be used to define the order in which the picture must be displayed.
Q So a decoder that decodes a temporal_reference value of 0 would understand that the picture associated with that temporal_reference value is the first picture in display order, right?
A I'm struggling to see the scenario in which a decoder would understand that.
Q Why?
A I think you're asking me a decoder to assume that a GOP is of a certain size.
Q Yes.
A How does a decoder know that?
Q I'm just saying -- well, can the encoder and decoder agree that a GOP can be of a certain size?
A Okay. What are the encoder and decoder now in your example?

Q Just an encoder that produces a MPEG-1 complaint bitstream and a decoder that produces an MPEG-1 compliant -- or decodes an MPEG-1 compliant bitstream?
A So a decoder that decodes an MPEG-1 compliant bitstream has to follow the decoding process of the standard. I don't recall in the standard anywhere that it says that the decoder in a certain circumstance can assume a GOP of less than a certain size. Maybe I'm missing something, but I don't recall seeing that.
Q And can you turn back to your second declaration, Paragraph 55?
A Okay.
Q You state that "Encoders can produce compliant bitstreams that are decodable by MPEG-1s decoding process."
Do you see that?
A Yes.
Q You continue, "That an MPEG-1 decoder must produce identifying results to the

defined encoding process."

Do you see that?

A Yes.

Q Why does an encoder need to produce identical results to the defined encoding process?

A Because the standard either says so or implies that by the way it's presented, is my recollection.

And sorry, I'll just add to that my recollection is that the compliance requirements for decoders mentioned on Page 49 of the MPEG-1 exhibit states that the compliance requirements for decoders are contained in ISO/IEC 11172-4, that document says exactly what a decoder should do, and my recollection is that document includes an instruction to the effect that a decoder must or should produce identical results to the defined decoding process and that includes the accuracy bounds of the D.C.T that I think I mentioned earlier.

Q Okay. If you look at Paragraph 57 of your second declaration.

You state that the decoder receives a coded bitstream and parses the temporal_reference syntax element and decodes it according to the standard.

Do you see that?

A Yes.

Q What does the decoder do with the temporal_reference value when it decodes it?

MR. UDICK: Objection to form.

A What do you mean by what does the decoder do with it?

Q Does it store this value somewhere in memory? Does it just decode the value and then ignore it?

A Okay. What decoder are you asking me. Are you asking me to imagine --

Q A decoder that decodes an MPEG-1 compliant bitstream?

A Okay. So a decoder that decodes an MPEG-1 compliant bitstream has to meet

certain requirements we've discussed. I don't recall the requirements that we've discussed in this, in the MPEG-1 part two document, or the MPEG-1 part four document. I don't recall anywhere it says when or how a decoded value should be stored. I mean, I can talk about decoders if I'm familiar with implementations that I'm familiar with.

Q So you don't recall where it says how or when a decoded value has to be stored. Would you agree that a decoded does have to be -- strike that.

You don't recall where it says -- MPEG-1 says how or when a decoded value has to be stored? Do you agree that a decoded value does have to be stored then?

MR. UDICK: Objection to form.

A I can certainly imagine, I can recall working on MPEG-1 software decoders, working with MPEG-1 hardware decoders. I can certainly imagine a situation where a decoder could or would parse a syntax element such as

temporal_reference decoded according to the standard, which in this case Page 44 of MPEG-2 involves, I think simply decoding a 10 bit unsigned integer from that 10 bit value. I can imagine in a software MPEG 1 decoder, for example, a decoder then placing that value in a variable.

Q If you look at Page 16 of MPEG-1.

There's a paragraph above Section 0.5 it starts, "All the macroblocks."

Do you see that?

A Yes.

Q The second sentence states that:

"If the picture is an I-picture that's being or a P-picture in the reference picture for subsequent pictures it is stored."

Sorry, let me rephrase that.

The second sentence says that.

"If the picture is an I-picture or P-picture that is a reference picture for subsequent pictures it is

stored, and and will replace the oldest stored reference picture."

Do you see that?

A Yes.

Q So an MPEG-1 is it stating here that if an I-picture or P-picture is a reference picture that it's going to be stored?

A In this overview architecture, so it's stating what it's stating, but I will draw your attention to Page 15, the start of this Section 0.4, Decoding, and it states that the description that follows and that includes the words you just directed me to, the description that follows is a very brief overview of one possible way of decoding a bitstream. Other decoders with different architectures are possible. So one way I think a person of ordinary skill in the art would be familiar with this description is, these would be something that you could call an informative part of the standard. It's not normative. So it's specifically saying

this is one architecture and one way of doing things.
Q So one way of doing things, would be to store a reference picture so that subsequent pictures can use it to be decoded?
A That's broadly consistent with my understanding of what's being said here.
Q And when it's stored in this example, it would replace the oldest stored reference picture; is that right?
MR. UDICK: Objection to form.
A In this example we're certainly not required by the standard in any way.
Q Yeah, but one of standards is giving the example where it would replace the oldest stored reference picture, right?
A Could replace the oldest. You know, it says you can do it this way but other architectures are possible.
Q But it doesn't say you can do it this way, right?
A In this example architecture, yes.

Q What information about this reference picture would be stored?

MR. UDICK: Objection to form.

A I don't know.

Q Would the temporal_reference be stored?

A I don't know from this description.

Q If you look at the next sentence it says "Before the pictures are displayed they may need to be reordered from the coded order to their natural display order."

Do you see that?

A Yes.

Q The natural display order would be indicated by the temporal_reference, right?

A The temporal_reference syntax element, which we've discussed, as it says in paragraph, sorry, Page 44 of the MPEG-1 standard, the temporal_reference is assigned in sequence to the pictures in display order. That's what the standard says.

Q So the natural display order would be indicated by the temporal_reference, right?

MR. UDICK: Objection to form.
A The standard says display order, it doesn't use the word natural.
Q The display order would be indicated by the temporal_reference, right?
A The temporal_reference is assigned to the pictures in display order, so I think it's the later section we looked at explains it, Page 87 of the MPEG-1 standard temporal_reference, I'm paraphrasing, can be used to define the order in which pictures must be displayed.
Q Okay. MPEG-1 discusses the concept of a group of pictures or a GOP, right?
A That's my recollection, yes.
Q And what does a closed_GOP mean in terms of MPEG-1?
A So Page 43 of the MPEG-1 standard, one of the syntax elements of the group of pictures layer is closed_GOP and the standard describes this as a one bit flag, which may be set to one. And I think there's some

discussion on Page 86 of the MPEG-1 specification title is closed_GOP and it says one bit flag follows the time code and I think that's referring, if I remember correctly, to closed_GOP flag part of the GOP layer syntax. And the closed_GOP flag denotes whether the group of pictures is open or closed. Closed groups and I think that means closed group of pictures can be decoded without using decoded pictures of the previous group and I think that also means group of pictures for motion compensation.

Q Okay. So a closed_GOP can be decoded without using the previous group of pictures -- image from the previous group of pictures for motion compensation?

A That's pretty much what those sections of the standard are saying.

Q And you reference the group of pictures layer that also includes a group_start_code, right?

A Yes. I see that on, for example,

Page 43 of the MPEG-1 standard. I think it also appears in the syntax structures earlier.

Q Okay. And an MPEG-1, the group of pictures header is followed by an I-picture, an encoding order and then by 0 or more P, B or I-pictures, right?

MR. UDICK: Objection to form.

A What do you mean by encoding order?

Q Well, if you look at Paragraph 58 of your second declaration, you state:

"An MPEG-1 bitstream is organized into groups of pictures, each beginning with the syntax element group _start_code and including thereafter (an encoding order, an I-picture followed by 0 or more P, B or I-pictures)."

Do you see that?

A I do. I think, to be honest, that's a typo, a typographical error that I hadn't noticed before. I don't think it's

incorrect, but I think I probably intended to write in "decoding" order.

Q Let me make that correction for the record. So -- and just to be clear for the record, on the second line on Page 21 of your second declaration, the parentheses should read "in decoding order" instead of "in encoding order"?

A I think -- it's a while since I wrote this, but I think "decoding order" or simply "coding order" would be probably how I would have intended to write that.

Q Is there a difference between encoding order and decoding order?

A So decoding order is something that is -- or bitstream order are concepts that are described or defined in the MPEG-1 standard. Encoding order, I don't recall if that's actually defined.

A way of doing encoding is to encode the bitstream in a particular order. But once again, the encoder isn't defined by the

standard, so I guess you can do it in a different order as long as the bitstream itself has the specified order.

Q And you agree that a typical order of a group of pictures consists of -- or the typical display order of a group of pictures consists of an I-picture, optionally preceded by one or more B-pictures and followed by P-pictures?

A Did you "the typical" or "a typical"?

Q A typical.

A That is a -- yeah, known or typical GOP structures. There are others such as the ones shown on Page 85 of the MPEG-1 standard, Page 86 of the MPEG-1 standard, Page 87 of the MPEG-1 standard. So there are a number of, I guess, well-known or typical GOP structures.

Q Okay. And just quickly, so an MPEG-1 talks about three different types of pictures, the first being I-pictures that are intracoded, so they can be decoded without

a -- reference any other coded pictures.

Do you agree? Paragraph 59 of your declaration should discuss that as well.

A Yeah, I'm not sure that I agree.

Q Okay.

A Yeah, I disagree.

Q What do you disagree with?

A I think you said that there were only three types of pictures disclosed in the standard. That's not correct.

Q Apologies about that. I didn't think I said "only," but if I did, I was mistaken.

So one type of picture described in MPEG-1 is an I-picture that is intracoded, which means it can be decoded without reference to neighbor code pictures.

Do you agree with that?

A Yes. Broadly, yes.

Q Okay. And there's another type that's P-picture, which is interpredicted, and so it can be decoded -- sorry. Let me restart that.

P-picture is interpredicted, right?
A The picture as a whole can be -- can use interprediction from the previous I- or P-picture in the bitstream.
Q Can a P-picture be decoded without reference to any other coded pictures?
A I think technically yes.
So for example -- so Section 2.4 to 2.7 on Page 39 of the MPEG-1 standard, "As part of the macroblock level is something called a macroblock _type syntax element. The" -- yeah, I could jump to Page 49 of the MPEG-1 standard. Talks about intracoded macroblocks near the top. And it says, "In I-pictures all macroblocks are intracoded and stored. In P-pictures and B-pictures, some macroblocks may be intracoded as identified by macroblock _type."
I think it's allowed in the standard to construct a P-picture that actually includes only intracoded macroblocks. It would be an unusual thing to do, but it is technically

allowed by the standard, as I recall.
Q A more usual thing would be for P-pictures to be interpredicted where they're determined and based in part on prediction from another image frame?
A That's certainly quite a typical way of using MPEG-1. I wouldn't say it's more typical, but it's certainly a way of using MPEG-1.
Q Okay. If you turn to Paragraph 33 of your first declaration. It's on Page 15.
A Okay.
Q And the second -- or the last sentence of Paragraph 43 you stated, "P-frame is encoded using only one previously encoded frame."
Do you see that?
A Yes.
Q Do you agree with that statement?
A I think to the point where I was intending to talk about using dependency or prediction from only one previously encoded

frame, I think that's correct. The P-frame has, if you like, the capability in the standard to be macroblocks of the P-frame may be decoded using one previously encoded frame, the previous -- the most recent I or P-picture as a prediction reference.

Q Can a P-frame such as the one you describe in that sentence in Paragraph 43 of your first declaration, can it be decoded without using information from the other frame on which it depends?

A So if a P-picture in MPEG-1 is -- if in the bitstream macroblocks within that P-picture are used interprediction, so if they're intercoded macroblocks, then the standard says that they should be decoded using prediction from a preceding I or P-picture in display order.

So in that situation, if macroblocks in that P-picture are interpredicted from a previous picture, then that frame can't be successfully decoded, according to the

standard, without using prediction from that previous picture.

Q And then you referenced this is a little bit earlier, but MPEG-1 also discusses what's called a closed_GOP flag, right?

A Yeah. We talked briefly about that.

Q And if the value of that flag is set to 1, that indicates that the group of pictures is closed, right?

Look at Page 86 of the MPEG-1 if you need to. Actually, I think you'll want to look at 43. Excuse me.

A Yeah, the terminology changed a little bit. Page 43 describes the closed_GOP flag as being set to 1 if the group of pictures has been encoded without motion vectors pointing to the previous group of pictures. And then on Page 86 it states that closed groups -- we talked about this earlier -- can be decoded without using decoded pictures of the previous group for motion compensation.

I think those are intended to be

statements consistent with each other even though they use slightly different terminology. So I think it's correct to say one POSITA reading this would understand that a closed_GOP flag of 1 indicates that this, following the group of pictures that it's part of the header of, is closed. And that means that this group of pictures that it's part of can be decoded without reference to a previous group of pictures.

Q Okay. So closed_GOP flag set to 1 would it indicate that each image frame in that GOP is either intracoded, such as an I-frame, or if it's intercoded, it only depends from other image frames within that GOP, right?

MR. UDICK: Objection to form.

Incomplete hypothetical.

A That's not what the standard says. It certainly doesn't put it that way. I know it's kind of an important point, so I'd rather use the standard's way of describing it or my way of describing it.

Q So you disagree with my statement?
A I didn't say that. I'm just trying to be particular in my answer.
So I describe the closed_GOP flag, for example, in Paragraphs 64 to 69 of my second declaration. I think I also talk about it later on in the declaration.
Q Okay. Just a couple more questions, and then we can take our break for lunch if that's okay?
A Sure.
Q So Page 43 of the MPEG-1 states that the closed_GOP is the one big flag set to 1 if the group of pictures has been encoded without motion vectors pointing to the previous group of pictures, right?
A Pretty much, yes.
Q Okay. And so each image frame in that GOP, if the closed_GOP flag is set to 1, would either have to be intracoded, which means it doesn't have to be decoded without depending on another image frame, or if that

image frame is intracoded, it can only depend on other image frames within a GOP, right?

MR. UDICK: Object to form.

A I think you maybe misspoke in the second part of your question. Or otherwise, I'm not clear about the second part of your question.

Q No problem. Let me repeat it.

So if a closed_GOP flag is set to 1, then each image frame in that GOP would either have to be intracoded or it can be decoded without reference to any other image frame or intercoded, but it would only be able to depend from another image frame within the same GOP, right?

MR. UDICK: Object to form.

A So you're paraphrasing. You know, you're using different words from the standard, different words from the way I describe it in my declaration. I think it's technically correct to say that would -- these would be characteristics of an MPEG-1 closed_GOP, but the standard doesn't place

restrictions on -- in the way that you describe.

That might be a result of what the standard specifies, but it doesn't use -- it doesn't say -- it doesn't put it the way you put it.

Q But is there anything in my statement that you would disagree with?

A So the standard doesn't say frames must be either intracoded or second -- or the other part that you said. The standard puts it in terms of the closed group, or closed GOP, can be decoded without using decoded pictures of the previous group for motion compensation. And this is on Page 86.

And Page 43, it says that the GOP is set to 1 if the group of pictures has been encoded without motion vectors pointing to the previous group of pictures. So there's no reference to intrapictures in either of those.

Q To be able to satisfy --

A Sorry. Just to finish it, a POSITA would be aware from the standard that each GOP has to have at least one intrapicture or intracoded, yeah, I-picture.

Q Sorry. I didn't mean to interrupt.

A Yeah, no problem.

Q But to satisfy what this MPEG-1 is saying for where -- if the closed_GOP flag is 1, then that group of pictures is encoded without motion vectors pointing to the previous group of pictures.

The image frames that make up that GOP would either have to be intracoded or if they're intercoded, they would only be able to depend from image frames within that GOP, right?

A The closed_GOP flag -- I think part of my problem the closed_GOP flag itself doesn't place any restrictions on frames having to be intracoded. Other parts of the MPEG-1 standard do, but the closed_GOP flag doesn't.

It -- the restrictions are sort of

articulated in a couple of ways, as we've discussed. The group of pictures has been encoded without motion vectors pointing to the previous group of pictures. Also the closed group of pictures -- I'm reading from Page 86 of MPEG-1 -- can be decoded without using decoded pictures of the previous group for motion compensation.

That's the way -- it doesn't put any restrictions directly on, for example, intracoded pictures. And then it provides examples. But the closed_GOP is a property of the GOP, and that's the way the standard articulates it. We can talk about what results that might have, but that's different from the way the standard specifies it.

Q Well, if the closed_GOP flag is set to 1, can that GOP include an intercoded picture that depends on another -- an image frame from another GOP?

A So the question would be whether that GOP itself can be decoded without using

decoded pictures of a previous group for motion compensation and/or if the group of pictures that the closed-GOP flag exists in has been encoded without motion vectors pointing to the previous group of pictures.

It's a different requirement. You're -- you seem to be articulating a requirement that is not in the standard. It might be an effect, but it's not a requirement in the standard.

Q I guess my question is a little bit different. If the closed_GOP flag is set to 1, can that GOP have intercoded pictures that depend on image frames outside of the GOP?

A That's not a requirement that the standard places. I think it might be -- it could be a result of generating bitstreams. I can't think at the moment of a bitstream that would -- or a closed_GOP, according to MPEG-1, that would break -- that would contradict what you just said. But it's not the way that the standard puts a requirement

on the bitstream.

MR. KENDRICK: All right. Take a break for lunch?

MR. UDICK: Sure.

(Whereupon, a recess was taken from 12:08 PM until 12:47 PM.)

MR. KENDRICK: Back on the record.

Q All right. Welcome back, Dr. Richardson.

A Thank you.

Q During our lunch break, did you talk about the substance of your testimony with counsel?

A No.

Q Great. Thank you.

If you recall, before our break we were talking about the closed_GOP flag.

A Yes.

Q And in your opinion, the closed_GOP flag is not an indication of any one specific coded frame, right?

A Yes, that's correct.

Q Does Limitation 8A of the '321 Patent require that the indication identify a specific image frame?

A The language of the claim is "Limitation 8A, decoding from the video sequence, an indication of at least one image frame which is the first image frame in decoding order of an independent sequence."

So the way I read it, it describes an indication of the first image frame in decoding order.

Q Okay. But it's not limited to -- that indication isn't limited to the first image frame decoding order, right?

A It's an indication of at least one image frame. So it's got an indicator frame, as I understand it. If it's not indicating at least one image frame, it doesn't seem to be satisfying this limitation.

Q But it can indicate multiple frames as long as one of those frames is the first image frame in decoding order; is that right?

MR. UDICK: Objection to form.
A I don't see how that would work.
Q Do you think it's required to identify a particular frame among multiple frames?
A The first image frame is a specific frame. It's identifying the first image frame, not identifying a frame, a specific frame.
Q Okay. Does Limitation 8A require the indication to be in a specific syntax layer?
A Not that I -- yeah, not in my view.
Q Okay. And does Claim A require the indication to be decoded immediately before the first image frame in decoding order of the independent sequence?
A I don't see that requirement.
Q Okay. And then -- so, if you recall, earlier in our day we talked about random access. Do you recall that?
A I recall discussion of random access.
Q And, in your opinion, MPEG-1 does not fully enable random access at a user

specified playback points, right?
A I don't recall if I say those precise words in my declaration.
Q Paragraph 73.
A So which paragraph?
Q 73. It's at the top -- or Page 26.
A Okay.
Q What do you mean when you say "MPEG-1 does not fully enable random access"?
A So I give some examples in the following paragraph, paragraphs, following Paragraph 73 of my second declaration. I think I talk about another example later on of like problems or potential problems with MPEG-1, random access where -- let me see.

I'm trying to recall, but if I look at Page 26 of my second declaration, there's a number of GOP examples, including, Example C, this is from Figure D19 of the MPEG-1 standard, where you have multiple I-pictures within a GOP.

So MPEG-1 will just attempt to start

decoding at the next I-picture within a GOP. And my recollection is that can cause problems for decoding, displaying pictures unintentional. Yeah. You can have pictures that simply can't be successfully decoded in MPEG-1 because of that.

Q Which pictures are you referring to in your answer?

A Well, for example, Figure D19C, which I reproduce on Page 26 of my second declaration, if you use the word to attempt, attempt to pick a random access point, let's say a picture, a picture, B4 or even P6, according to MPEG-1, a decoder will just look for the next star code, where star code could include an I-picture star code.

So my recollection is that an MPEG-1 decoder will then attempt to start decoding I8. Yeah. It will attempt to decode P6, B7, I8, and it's not going to be able to -- yeah. Let's say it decodes I8, B7 actually depends on I8, but -- no, this is a bad example.

Yeah. B7 depends -- so it's not such a bad example. To decode -- decode attempts to decode B7 and I8 -- yeah. In coding order -- sorry. I'll start that answer again.

If you take the coding order, for example, C in Figure D19 of the MPEG-1 standard, if a decoder attempts to start decoding of I8, or at frame I8, is the start coding after the point at which the user tries to perform random access.

So according to MPEG-1, the decoder would read the start code of I8, and attempt to start decoding. And then B7 can't be successfully decoded because it depends on I8 and also on P6, which is not available. So B7, whether this is a closed or open GOP, B7 is now an indeterminate picture.

So that's an example of where decoder maybe infers an unintentional loss of pictures when, in fact, it simply doesn't have the mechanism to successfully start decoding at a user specified playback point.

Q So just let me see if I can -- if I understand your answer here.

So you're saying that, in your opinion, MPEG-1 does not fully enable random access because if you look at the example C in D19, if a user tried to start or requested to start the video sequence at B5, for example, the image frames would be lost until the decoder got to image frame I8 in that start code?

A I think that's broadly what I was trying to convey. And I was looking for that part in the '321 Patent where it talks about unintentional loss of image frames, and I couldn't find it momentarily, but I think that's an example of the types of problems that the '321 Patent was intending to solve.

Q Column 11 is where you're looking for.

A Thank you.

Q And the PDF on your computer should be text searchable if you need.

A Good point. Thank you for the reminder.

Q No problem.

A So there's also a numbering problem as the '321 Patent points out in Column 11, right about Line 20, discontinuity typically occurs in the image numbering. The decoder typically interprets this as an unintentional loss of image frames.

So I think there are just problems like the one that I tried to articulate in the prior art to the '321 Patent, where you could have a decoder according to the MPEG-1 standard in this case. It tells the decoder, if you like, that to start decoding from, look for the next start code from an I-picture or at a higher level and start decoding. But you then have problematic frames, such as the B7, that I talked about in this example. And a decoder actually doesn't know how to handle those in MPEG-1. It doesn't have all of the information necessary for decoding, even though it is, the subsequent framing coding order, and so

according to the standard the decoder should attempt to decode that.

Q Does MPEG-1 disclose the constant of random access of the video stream?

A It talks about random access and it provides some tools for random access. But I would stand by my characterization that we discussed earlier, that the '321 Patent improves upon -- improves the random access process or the random access capabilities compared with the prior art at the time.

Q One of the tools for random access that MPEG-1 talks about is the use of relatively short groups of pictures.

Would you agree?

A I remember -- I recall the standard does talk about that as examples of how an MPEG-1 might choose to construct groups of pictures. I think it's in Annex D.

Q If you look at Page 33 of the standard. That first paragraph, the second sentence, or the third sentence, excuse me.

A So this is the requirements section.

Yeah. Applications may require in random access, fast forward playback or fast normal reverse playback may use relatively short groups of pictures.

Q Does Claim 8 of the '321 Patent require that a method of decoding that fully enables random access?

A What do you mean by "fully enables"?

Q The way I'm trying to -- you said that, in your declaration, MPEG-1 does not fully enable random access. And the way that you used that term, does Claim 8 require a method for decoding to fully enable random access?

A So as I say in Paragraph 41 of my second declaration, in my view, the '321 Patent is directed to improving random access. So I wouldn't necessarily use the word "fully enable" because I think that's kind of hard to define. But, in my view, the '321 Patent certainly discloses ways of improving random access.

Q That doesn't quite answer my question.
Does Claim 8 require that the method of decoding fully-enable random access?
A I don't think those words appear in the claim.
Q Okay. And I think you got into this a little bit earlier when I asked about fully enable, but you referenced the example of MPEG-1 that you give after Paragraph 74 of your declaration?
A Briefly, yes.
Q Okay. And in this example, it starts where a user requests to start the video sequence at frame B4?
A Yes.
Q Okay. And your opinion is that MPEG-1 would try to find the first I-frame, which in this example would be frame I8 and it would start encoding from there, right?
MR. UDICK: Objection to form.
A The MPEG-1 standard, as I recall, suggests or specifies that a decoder should

or can start decoding from a start code, including an I-picture start code.

Q And so in the example under Paragraph 74 of your second declaration, all the I-frames before Frame I8 would be discarded; is that your opinion?

A So stamped Page 120 of the MPEG-1 standards says that -- and this is specifying or describing how an MPEG-1 comply decoder should carry out decoding. It says:

> "If playback begins from a random point in the bitstream, the decoder should discard all the bits until it finds a sequence start code, a group of pictures start code, or a picture start code which introduces an I-picture."

So my understanding of what's being described or specified there is that, in a case where an MPEG-1 decoder attempts to start decoding from Picture B4, it should decode everything -- sorry, discard everything up to the point at which it, in

this example, encounters the start code associated with Picture I8.

Q If the method for decoding a video sequence recited in Claim 8 was applied to your example under Paragraph 74, would the decoder reset the identifier value of Frame I8? Or, sorry, let me strike that. Let me ask a different question.

If the method for decoding a video sequence recited in Claim 8 is applied to your example under Paragraph 74, would the decoder start decoding from Frame I8?

MR. UDICK: Objection to form.

A My example in Paragraph 74 and onwards refers to a decoder that complies with the MPEG-1 standard. And, in my opinion, the '321 Patent isn't -- the MPEG-1 standard doesn't cover or disclose or practice or even render obvious the '321 Patent. So I don't really understand how it can apply to the method.

Q Well, I'm not asking about what -- the

MPEG-1 decoder you discuss in this paragraph, I'm asking, as a person of ordinary skill, if you take the method for decoding the compressed video sequence recited in Claim 8 and apply it to this bitstream shown in Example D19C of MPEG-1, would that method for decoding start from Frame I8?

MR. UDICK: Objection to form. Scope, incomplete hypothetical.

A So what video decoder do you want me to imagine?

Q One that uses the method recited in Claim 8.

A What is the video decoder?

Q The video decoder practices the limitation -- or the claim recited -- or the method recited in Claim 8, is that not enough specificity for you?

A Claim 8 is -- yeah. We talked about this before, so MPEG-1 is a relatively short video coding standard, as these things go, and it's about 130 pages, plus the other

parts that we mentioned that go together.

Claim 8 of the '321 Patent is, whatever, a hundred words or something like that. It doesn't specify a video decoder in its entirety.

Q Does the MPEG-1 standard that's produced as Exhibit 1005 describe a video decoder in its entirety?

A It describes a method or process for video decoding and accompanying syntax and semantics that, as I recall, is intended to be a complete description.

So one can, and I've done this, one can take the MPEG-1 standard and use it to inform the development of a video decoder to comply with the standard, and if you did it correctly, you'd end up with a functional video decoder. But you need a lot of information about how the syntax works, how the semantics work, how the decoding of that syntax and semantics work.

So I still don't know what video decoder

you want me to imagine here.
Q Well, let's look at the method for Claim 8.
So that method starts with decoding from the video sequence and the indication of at least one image frame, which is the first image frame in decoding order.
Do you see that?
A Yes.
Q If a decoder employing the method recited in Claim 8, if the user requests it to start at Frame B4, in the example you give under Paragraph 74 of your second declaration, what would be the indication of the first image frame in decoding order?
MR. UDICK: Objection to form. Scope and incomplete hypothetical.
A So I'm describing an MPEG-1 decoder, or an MPEG-1 compatible decoder in my example of Paragraph 74 of my second declaration, that doesn't decode an indication of at least one image frame, which is the first image frame

in decoding order, but independent sequence, it just doesn't do it. I've explained in detail in my declaration why, in my opinion, it doesn't do it. So how can it do it? I don't understand.

Q Well, that's not my question.

My question is, are you able -- as a person of ordinary skill in the art, are you able to opine on whether a decoder practicing the method in Claim 8 can decode the video sequence given in Example C?

MR. UDICK: Same objections.

Q Where the user starts at Frame B4?

MR. UDICK: Same objections.

A Are you asking me to answer that question without knowing what the video decoder is?

Q I'm asking you to answer the question where the video decoder practices the method recited in Claim 8.

A But I know nothing else about the video decoder?

Q Yeah. Are you unable to answer that question without knowing anything else?
A And it's decoding an MPEG-1 bitstream?
Q It's decoding a bitstream, such as the example provided in MPEG-1 at Figure D19C.
A Which is an MPEG-1 bitstream?
Q Which is an MPEG-1 bitstream.
A So this is a decoder that's not an MPEG-1 decoder, but it's decoding an MPEG-1 bitstream?
Q Does the '321 Patent practice MPEG-1? Sorry. Does MPEG-1 practice the '321 Patent?
MR. UDICK: Objection to form. Scope.
A Okay. What do you mean does MPEG-1 practice the '321 Patent?
Q Would the '321 Patent be necessary for somebody to practice -- to decode an MPEG-1 to bitstream?
MR. UDICK: Objection to form. Scope.
Calls for a legal conclusion.
A It sounds like you're asking me if an MPEG-1 decoder infringes the '321 Patent. I

haven't been asked to make that determination and I haven't made that analysis.
Q For the bitstream in Example D19C that should be produced under Paragraph 74 of your second declaration.
A Okay.
Q Is that bitstream only available in MPEG-1?
A What do you mean "available in MPEG-1"?
Q Could a bitstream that starts with an I-frame with a temporal_reference of one followed by B-frame with a temporal_reference of 0 and so on, can that bitstream be, can that bitstream follow a different standard outside of MPEG-1?
A What's the bitstream.
Q The bitstream shown in the example you give under Paragraph 74?
A That's just a diagram, what's the bitstream?
Q It would be an I-picture, followed by a B-picture, and this is in coding order,

followed by a P-picture, then a B-picture, a B-picture, a B-picture, a B-picture, an I-picture, a B-picture, an I-picture, a B-picture, a B-picture, and then ending with an I-picture.

MR. UDICK: Objection to form.

A And how are the pictures encoded or coded in the bitstream?

Q Can they be coded under H.264 syntax?

MR. UDICK: Objection to form and scope.

A I don't recall that H.264 specifies a temporal_reference syntax only.

Q Is that a no then?

A I don't have the H.264 Standard in front of me, but my recollection is that, yeah, if you look at Page 26 of my second declaration, which includes the larger extract from the MPEG-1 standard and in that extract it says, some example groups of pictures with their temporal_reference numbers are given below.

So if it's the case that MPEG-1, sorry,

H.264 doesn't include the syntax or semantics for temporal_reference numbers then I don't see how that can be done.

Q Okay. And Dr. Richardson, I believe you said you reviewed the petitioner's petition challenging the validity of the '321 Patent; is that right?

A I recall reading it, yes.

Q And you understand that it discusses what petitioner calls an exemplary closed_GOP, right?

A I recall that, yes.

Q Do you understand that Amazon or the petitioner's exemplary closed_GOP has a few properties?

A So I think I talk about this in my first declaration, so I want to -- so I'm going to my first declaration and then I'll go to the petition and then perhaps you can point me to what you want me to talk about.

Q Paragraph 63 of your first declaration, I believe that's what you're looking for.

A Okay. So the petition at starting around Page 16 talks about an exemplary, what it describes as an exemplary closed_GOP as you say, I provide some discussion of that in Paragraph 63 of my first declaration and I think I carry on throughout later sections of my first declaration. So what was the question.

Q Oh, just that you recall that petitioners petition relies on an exemplary closed_GOP that has a few properties. I believe you answered that.

Moving on to the properties, the first property of the petitioner's exemplary closed_GOP is that an I-frame is the first image frame in decoding order, right?

A So this is as I recall the petitioner's definition. I don't think exemplary closed_GOP appears in the MPEG-1 standard.

Q I'm not representing that it does.

A Okay.

Q I'm asking -- what I'm representing is

that it has certain properties that are disclosed in the MPEG-1 standard and the first would be that an I-frame is the first image frame in decoding order, do you understand that the petitioner's exemplary closed_GOP has that -- petitioners states it has that property?

A I understand that petitioner has stated that there is an exemplary closed_GOP in MPEG-1 that has that property, but I don't recall MPEG-1 stating that property.

Q If you go to Page 84 of MPEG-1 and under Section D.5.2, title group of pictures, there's a property one which states a group of pictures in bitstream order must start with an I-picture and may be followed by any number of I, P, or B pictures in any order?

A I see that.

Q So MPEG-1 here is stating that a group of pictures in bitstream order must start with an I-picture, right?

A Yes.

Q And that's the first property that petitioner identifies for its exemplary closed_GOP is that an I-frame is the first image frame in decoding order, right?
A That's what petitioner says, yes.
Q That property is specifically disclosed in MPEG-1 at Page 84, right?
A That's just a property of GOP in MPEG-1. There's no need to use other words such as exemplary.
Q Okay. The next property of petitioners exemplary closed_GOP is that an I-frame is the first for image frame in display order, do you agree?
MR. UDICK: Objection to form.
A I agree that petitioner has defined, I would go so far as said made up, a so called exemplary closed_GOP and they have said that in their view it has two properties and the second of which is the one you just disclosed. I can't speak for petitioner, but I have read there what I think is a

constructive definition.
Q Okay. And MPEG-1 specifically discloses or specifies that a group of pictures must begin in display order with an I or B-picture, right?
A Yes.
Q And MPEG-1 teaches that if a group of pictures is closed B-pictures preceding an I-picture in display order can be omitted, right?
MR. UDICK: Objection to form.
A Where, I'm sorry.
Q If you look at Page 85, first paragraph, second to last sentence, it states, one way to do this is to omit any B-pictures preceding the first I-picture.
A It's not talking about a closed_GOP if that's what you're asking.
Q Okay. If you look at the first sentence of this paragraph.
It says, "the original concept of a group of pictures was a set of

pictures that can be coded and displayed independently of any other group."

Do you see that?

A I do.

Q Would you agree that original concept is referencing a closed group of pictures?

A No.

Q Why?

A Where does it say that?

Q Do you recall talking about a closed group of pictures earlier?

A As defined in MPEG-1 standard, yes.

Q What is a closed group of pictures as defined in the MPEG-1?

A So I can look at the Page 43 of the MPEG-1 standard.

Q Okay.

A If the closed group of GOP flag is set to one it indicates to a decoder that the group of pictures has been encoded without motion vectors pointing to the previous group

of pictures. Sorry. It's provided -- it goes on to say, this bit is provided for use during any editing, which occurs after encoding.

Q So MPEG-1 states that a closed_GOP is one that -- in MPEG-1 teaches that a closed_GOP is one where the group of pictures has been encoded without motion vectors pointing to the previous group of pictures, right?

A I think we discussed earlier that the actual semantics, the way of describing it is just to say that if closed_GOP is one than that property is true. Then later on there is discussion of a closed_GOP on Page 86 and it uses slightly different terminology, I think we discussed earlier that it might be the terminology of two sections is consistent, it never quite states exactly what you stated.

Q So looking at the description on Page 43. A group of pictures has been

encoded without motion vectors pointing to a previous group of pictures. Then that group of pictures can be decoded and displayed independent of another group of pictures, right?

A Are you asking about a closed_GOP or something else?

Q A closed GOP?

A Can you frame the question again because that wasn't what you asked.

Q Looking at the description on Page 43 a closed group of pictures is one that is -- excuse me.

Looking at a closed_GOP it is a group of pictures that has been encoded without motion vectors pointing to a previous group of pictures, would you agree?

A It -- I disagree with what it says in the standard. If closed_GOP flag in the group of pictures layer, group of pictures layer, sorry, is set to one or that flag may be set to one if the group of pictures has

been encoded without motion vectors pointing to a group of pictures. That's what the standard says.

Q So you disagree that a closed group of pictures is a group of pictures that has been encoded without point using motion vectors pointing to any other group of pictures?

MR. UDICK: Objection to form.

A I see the confusion here. You're turning it around with respect.

Q How am I turning it around?

A You're saying that if the group of pictures has been encoded without motion vectors pointing to the previous group of pictures, it is therefore closed_GOP. That's not what the standard says.

Q What does the standard say?

A It says, that if closed_GOP may be set to one, that's the flag, closed_GOP may be set to one if the group of pictures has been encoded without vectors pointing to the previous group of pictures. It doesn't say

it must be set to one or it is always set to one in that condition.
Q If the closed_GOP flag is set to 0 is the group opened or closed?
A Page 86 of the MPEG-1 standard says, if the closed_GOP flag is set to 0 then the group is open.
Q So is MPEG-1 teaches a difference between an open group of pictures and a closed group of pictures, right?
MR. UDICK: Objection to form.
A Not necessarily.
Q Does MPEG-1 teach that closed groups can be decoded without using decoded pictures of the previous group for motion compensation, whereas open groups require such pictures to be available?
A Yes.
Q But MPEG-1 does not teach that a group of pictures is opened or closed?
A Sorry, say that again.
Q I asked earlier if MPEG-1 teaches a

difference between an open group of pictures and a closed group of picture and you said not necessarily, right?

A Yeah, my confusion is that the way I read the actual specification group of pictures layer semantics, Page 43 of the MPEG-1 Standard. Closed_GOP may be set to one, and "may" I think is a word that I assume was chosen carefully, if the group of pictures has been encoded without motion vectors pointing to the previous group of pictures.

Hypothetical example, if a group of pictures has been encoded without motion vectors pointing to the previous group of pictures, but the closed_GOP flag in the group of pictures layer for that picture is set to 0, it's the same -- it has that same property of being encoded without motion vectors pointing to the previous group of pictures, as a group of pictures for which the closed_GOP flag was set to one.

Q Okay. MPEG-1 later says that a closed_GOP is one bit flag -- or the one bit flag follows the time code. And that one bit flag denotes whether the group of pictures is opened or closed, would you agree?
A It does say that later on, yes, in Annex D.
Q And MPEG-1 states that closed groups can be decoded without using decoded pictures of a previous group for motion compensation, do you agree?
A Okay. In the informative guide to encoding video in Annex D it does, yes.
Q It has a counter-example for open groups which require such pictures to be available.
Do you see that?
A Where does it say counter-example?
Q I'm just using it, counter-examples a colloquial term and if you look at the last clause of that section that we were reading it says, "whereas open groups require such pictures to be available."

Do you see that?
A I see that, again, it's part of informative Annex D.
Q Great, thank you. Informative Annex D also states that the original concept of a group of pictures was a set of pictures that could be coded and displayed independently of any other group, do you agree with that?
A Isn't that on page -- Yeah, I see that sentence. It's not describing the MPEG-1 standard if that's what you're suggesting.
Q I'm not suggesting that.
A Okay.
Q If you look at Page 86, it says "A typical example of a closed group, which refers to a closed group of pictures is shown in Figure D18A."
Do you see that?
A Yes.
Q In this typical example of a closed group of pictures an I-picture is the first picture in display order for that closed

group, do you agree?
A So in the part of the informative Annex D, it doesn't specifically say that the numbers underneath the pictures are temporal_reference numbers. Which would tell me whether or not that initial I-picture is the first picture in display order. It seems consistent with other examples, for example, on Page 87, where the numbers in a similar looking example do refer to temporal_reference numbers, but it's not entirely clear from Page 86 of this informative Annex D.
Q Let's look at Page 87 then, if you look at Figure D19, example A, that shows a group of pictures with an I-picture first in display order, right?
A Yes. It does.
Q Would a person of ordinary skill understand the numbers under this example here are temporal_reference values?
A The example it starts around half way

down Page 87 of the MPEG-1 standard. I would say, yes.

Q But you're unable to state one way or the other whether the numbers under the image frame -- or the I, B and P letters in Figure D18 are temporal_reference values, right?

A It simply doesn't say that in D18 in the text around it. It does say example groups of pictures. In Figure D18, it does say example groups of pictures in display order but unlike the following page, it doesn't mention temporal_reference numbers.

Q And so you're unable to opine on whether those numbers are temporal_reference numbers?

A I'm not disputing that they may be temporal_reference numbers. I'm just pointing out that to the extent that you're saying that they are, it doesn't actually state that in this part of the annex.

Q And absent an explicit statement to that effect, a person of skill would not be able

to determine that?
A It's just a diagram. It's not a bitstream. So there are no actual temporal_reference numbers in either of these. But the difference is it's explicitly called out in the examples on Page 87. It's not explicitly called out in the example on Page 86.
Q An I-picture is the first image frame in display order will have a temporal_reference value of 0, right?
MR. UDICK: Object to form.
A First image frame display order of what?
Q Of a group of pictures?
A So Page 87, still in the informative annex, but my recollection is this is consistent with the normative part of the standard. For example, Page 44 -- Page 44 of the MPEG-1 standard, which we discussed earlier says, for the earliest picture in display order in each group of pictures, the temporal_reference is reset to 0. And the

examples of -- on Page 87, the first example, A, begins with an I-picture. And it's temporal_reference number, according to the example on Page 87, is 0.

In Example B, there's another example GOP where a B-picture is the first picture in display order and has a temporal_reference number, according the figure D19, of 0.

Q I don't think you quite understood my question, so I apologize if it was unclear.

If an I-picture is the first image frame in display order, it will have a temporal_reference of 0, correct?

MR. UDICK: Object to form.

A If an -- an MPEG-1 bitstream, if a GOP -- a GOP begins with an I-picture. If that I-picture, the beginning of a -- the first I-picture in the GOP in coding order or bitstream order is also the first picture in the GOP in display order, then by definition it will have a temporal_reference of 0.

Q Thank you.

And one example of that, shown in the informative annex of MPEG-1, is Figure D19, Example A, right?
A Yes.
Q And then Figure D19, Example A, the I-picture is both first in display order and first in decoding order, right?
A In the five-picture GOP illustrated in that example, yes.
Q And that I-picture, which is the first picture in the display order and decoding order, has a temporal_reference value of 0, right?
A In that one example of Figure D19, which has two other examples, yes.
Q Okay. If you turn back to the '321 Patent.
So looking at MPEG-1 and whether it discloses Limitation 8A, you don't dispute that MPEG-1 discloses a closed GOP, do you?
MR. UDICK: Objection to form.
A No. I don't dispute that. I talk about

MPEG-1's closed GOPs or closed groups of pictures in several places in each of my declarations.

Q And you don't dispute that MPEG-1's closed GOP teaches an independent sequence for an all-motion-compensated temporal prediction references of the independent sequence refer only to image frames within said independent sequence, right?

A Are you just asking me in general or in context of Claim 8 of the '321 Patent.

Q In the context of Claim 8 of the '321 Patent.

A So you're asking to take a sub-phrase out of the element and ignore the rest of the claim or what?

Q I'm asking you whether you dispute that MPEG-1's closed GOP teaches that sub-phrase of Limitation 8A.

A So I'm not a patent lawyer, but my understanding is that -- what I've done in my declarations is gone through -- as Petitioner

did, I think, in their petition, is gone through Claim 8 element by element.

And Petitioner has expressed an argument or an opinion about whether or not Element 8A is satisfied by MPEG-1. Petitioner's position, as I understand it, is that MPEG-1 satisfies or renders obvious -- yeah, it's the other way around. MPEG-1 renders obvious Element 8A of Claim 8 of the '321 Patent. My opinion is that it does not, as I've expressed at length in my declaration.

Q Okay. And the basis for that opinion is that MPEG-1 does not disclose the indication recited in Limitation 8A, right?

A Not just that.

Q Can you turn to the institution decision, please. It's Paper 10 on that laptop.

A Sure.

Q And when you open the PDF, if you want to turn to Page 21. We'll be looking at the very bottom.

A Okay.
Q And under subsection C at the bottom of Page 21, the board states:
"Based on the current record, Petitioner establishes sufficiently that MPEG-1 teaches the claimed independent sequence where an all-motion-compensated temporal prediction references of the independent sequence refer only to images within said independent sequence because MPEG-1 discloses a closed GOP."
Did I read that correctly?
A Yes.
Q Do you disagree with the board?
A In general, I disagree with the board, yes.
Q No, I'm asking you if you disagree with this sentence from the board?
MR. UDICK: Objection to form. And scope.
Q Sorry. Do you disagree with this

sentence that I just read from the board's institution decision?

MR. UDICK: Same objections.

A I haven't formed an opinion in that. I've formed an opinion on whether or not MPEG-1 teaches or discloses or renders obvious Element 8A of the '321 Patent, and I've expressed that opinion in detail in my report.

I'm not a patent lawyer. I don't know the board -- I'm not board. I don't know their frame of reference or what -- how they go about things. And to be honest, I have reviewed the institution decision, but I haven't rendered opinions on the institution decision. I will say that I disagree with at least some of the board's conclusions.

Q Sitting here today, you're not able to offer an opinion on whether you agree or disagree with the statement I read from the board's institution decision?

MR. UDICK: Objection to form.

A Okay. I'd have to know whether it is even relevant to take a claim element, partition it into short phrases and then put a ticker across, as it were -- metaphorical ticker across against each phrase. I mean, does MPEG-1, word by word -- I don't know if that's appropriate for me, and it's certainly not something I've been asked to render an opinion on so far in this proceeding.
Q Okay. So sitting here today, you're not able to offer an opinion on whether you agree or disagree with the statement I just read from the board's institution decision?
A I haven't offered such an opinion. I would have to think about it, and I would probably ask for clarification from counsel, for example, as to how that relates to what I have done, which is rendered an opinion, for example, on whether MPEG-1 one teaches Element 8A, Element 8B and so on in this claim.
Q Okay. In your second declaration, you

opine that MPEG-1's closed_GOP flag does not teach or suggest decoding from the video sequence an indication of the at least one image frame, which is the first image frame in decoding order of an independent sequence, right?

A I discussed that here, starting, I think, around Paragraph 99 of my second declaration.

Q And you disagree for two reasons: First, your opinion is that the closed_GOP flag is a single-bit flag that only indicates whether the entire GOP is open or closed and it does not identify an image frame; and second, the closed-GOP flag does not indicate which image frame is the first, right?

A Yeah. In my opinion, Petitioner is incorrect for at least those two reasons.

Q Okay. And looking at example -- or Figure D19, Example A in MPEG-1, after the decoder decodes the closed_GOP flag in that group of pictures header, what is the first

image frame in decoding order?
A So where are we? Figure D19, Page 87?
Q Example A.
A Okay. Say again.
Q Yeah, sorry. I was about to repeat my question.
Looking at that example, after an MPEG-1 decoder decodes the closed_GOP flag, what is the first image frame in decoding order?
A That's not what comes after the closed_GOP flag.
Q Yeah, but I'm asking what is the first image frame in decoding order that comes after the closed_GOP flag.
A So according to the syntax, Page 37 of the MPEG-1 standard, is the group of pictures layer.
Q Um-hmm.
A So the group of pictures layer syntax starts with a number of syntax elements. One of those is closed_GOP. And then there's a sort of the syntax function or syntax

structure. Next, _start code, open and close, parentheses, which is -- and my recollection is that basically will be interpreted by somebody implementing a decoder as, colloquially, "go and find the next start code."

And once you get to the point where that's a picture start code, you then have one or more picture syntax structures. And each of those would have picture layer syntax structure. And the first of those picture layer syntax structures in the example you asked me about, I think, would be an I-picture, in this case, with temporal_reference 0.

Q Okay. And do you agree that the first image frame received by a decoder after the closed_GOP flag has to be an I-frame?

MR. UDICK: Objection to form.

A So when a decoder encounters a group of pictures layer syntax structure, part of that syntax structure is the closed_GOP flag.

Part of that picture structure that -- that group of pictures syntax structure that follows later on from the closed_GOP flag is a series of picture syntax structures. The first of those picture syntax structures will be a picture that is an I-picture, according to the standard.

Q In your response in your second declaration, you take the opinion that an indication must necessarily inform the decoder of something, right?

A So Paragraph 102 in my second declaration, in my opinion a person of ordinary skill in the art would understand indication in the context of the claims to refer to information in the encoded bitstream that necessarily informs decoders of something. That's my opinion.

Q You also have the opinion that an indication must confirm the thing present to the decoder, right?

A If you're referring to my Paragraph 103,

that's not quite the way I put it. But I'm presenting my opinion of what a person of ordinary skill in the art would understand by the word "indication" in this context. That it would -- if the indication is present and the thing indicated is confirmed necessarily to the decoder.

Q Is informing the decoder of something and confirming that something is present, are those synonymous?

A I don't think I said that confirming that something is present.

Q I apologize. Is necessarily informing the decoder of something synonymous with confirming to the encoder what the thing indicated?

A I don't quite understand the question.

Q So you state that, in your opinion, an indication in the claims refers to information in an encoded bitstream that necessarily informs the decoders of something, right?

A In the context of the challenged claims of the '321 Patent, yes.
Q And then you also take the opinion that if the indication is present, then the thing indicated is confirmed necessarily to decoder, right?
A That's the way I put it in my Paragraph 103.
Q Are those two concepts synonymous, or are they two requirements that the indication must fulfill?
A They are -- my -- they're both my attempt to express my opinion as to what a person of ordinary skill would understand reading the word "indication" in this context, in the context of the challenged claims of the '321 Patent.
Q In your opinion, does an indication always have to confirm the presence of the thing indicated?
A In what context?
Q In the context of the '321 Patent.

A I'm talking about the claims of the '321 Patent. Are you asking about something else, or? I'm specifically talking about the challenged claims of the '321 Patent and how a POSITA, in my view, would interpret those claims.
Q In the context of the challenged claims of the '321 Patent, does the indication always have to confirm the presence of the thing indicated?
A I can't see in the challenge claims the use of the word "the indication" is inconsistent with that.
Q So in the context of the challenge claims of the '321 Patent, the indication always has to confirm the presence of the thing indicated?
A It does confirm the presence of the thing indicated. That's the way a POSITA would understand it, in my view.
Q To support your opinion on how a POSITA would understand the term "indication," you

don't cite to any portion of the '321 Patent, right?
A I don't cite to a portion of the '321 Patent in my Paragraphs 102 and 103 or 99.
I would like to sort of comment on every page of my declaration.
Q In Paragraphs 102 and 103, where you provide your understanding of how a person of skill would interpret the term "indication," you don't cite to the '321 Patent, right?
A So I cite to, in Paragraph 111, and I'm still talking about indication, and I'm still talking about the closed_GOP flag. So I don't think that's strictly correct.
I mean, it's correct -- we can say whether -- we can discuss whether or not I have a citation in a particular paragraph, but I cite to the '321 Patent in other paragraphs, and I cite to other documents. And my discussion carries on to at least in my Paragraph 122.
Q You don't have any citations to the '321

Patent in Paragraphs 102 or 103, right?
A In those two paragraphs, I agree, but I don't recall intending them to be taken out of context in my declaration.
Q Okay. But you do provide this to support your concept of indication. You provide an example from an indication from the H.264 Standard, right?
A I disagree that it's my concept of indication.
Q Sorry. Let me rephrase.
To provide your opinion on the concept of indication, you provide an example from an H.264, correct?
A Yes.
Q And that is a NAL unit; is that right?
A I discussed the NAL unit in Paragraph 106. I think we talked about this earlier.
Q And under Paragraph 106 of your second declaration, there's a table of NAL unit type codes.

Do you see that?
A Yes.
Q It's not the full table of NAL unit type codes from H.264, right?
A I realize earlier I got it wrong when I say that I didn't have the H.264 standard in front of me, because, of course, you provided it on the computer.
Q I did.
A Yeah. Thank you for that.
Q You're welcome.
A Page 57.
So Page 57 of Exhibit 2006, which according to the exhibit marking is Page 73, has the full Table 7-1 NAL unit type codes. And the answer is, no, it's not the complete table in my declaration. As I say in Paragraph 106, it's excerpted.
Q Is there a reason you only provided an excerpt of the table?
A Not that I recall.
Q Okay. And in your second declaration,

you state that:

"The concept of an indication that can be decoded from a coded bitstream applies across video coding systems."

Do you see that?

A I see that in Paragraph 105 of my second declaration.

Q But the only example you give is from H.264, right?

A I gave an example from H.264 in this section, yes.

Q Okay.

A I wouldn't say it's the only example in this section.

Q What's the other example?

A I'm talking in the same section about MPEG-1's closed_GOP flag.

And, as I say in Paragraph 100, MPEG-1's closed_GOP flag is only an indication of whether or not the current group of pictures is open or closed.

That's an example of an indication.

Q So you -- as part of your discussion, the NAL unit, you cite to the H.264 Standard, right?
A Yes. In my Paragraph 1006 -- sorry, 106, and I cite to it as Exhibit 2006.
Q And Exhibit 2006 has a publication date of March 2005, right?
A The particular version that was included as an exhibit does, yes.
Q The priority date of the '321 Patent is January 23rd, 2002, right?
A I can't recall.
Q Feel free to look. I don't want to have it be a memory contest.
A Yeah. That appears to be consistent with the front page of the '321 Patent, if I'm reading it correctly.
Q Okay. So the priority date of the '321 Patent is over three years prior to the publication of the H.264 reference you rely on, right?
A The H.264 Standard was first published

in 2003, not 2005.
Q But the version that you rely on for your discussion of the NAL unit was published in 2005, right?
A To the extent that you're implying that that's only version I rely on in any discussion of H.264, I would disagree with you.
Q Do you cite to another version of H.264 in your declaration?
A I don't recall. But I certainly talk about my -- for example, my -- we talked earlier about my books, I talk about my book that I wrote in 2003 about the published H.264 Standard.

So, yeah, this isn't -- certainly, this version isn't the entirety of my knowledge of the standard.
Q But don't cite to another version of H.264 in your declaration, do you?
A I don't recall.
Q Okay. In your experience of serving as

an expert in patent -- retained expert for patent litigation matters, have you ever opined on the construction of claim terms?

A I think so, yes. Yes.

Q Do you recall how many instances you opined on the construction of claim terms in offering extrinsic evidence from three years after the priority date to support that construction?

MR. UDICK: Objection to form.

A Sorry. Where I am opining on construction claim terms?

Q You're opining on what the term "indication" means, right?

MR. UDICK: Objection to form.

A I'm opining, for example, in Paragraph 102 of my declaration, on a POSITA's understanding of indication in the claims.

Q Okay. You don't think that's a construction?

A I'm not a patent lawyer. I mean, I know

a certain amount about what it means to be a claim construction, but I just say my opinion is what it is. I'm not a technical guy.

Q Okay. And I don't think you quite answered my former question.

Do you recall how many instances you opine on the construction of claim terms where you offered extrinsic evidence from three years after the priority date to inform that construction in your previous testifying -- or expert witness experience?

MR. UDICK: Objection to form. Scope.

A I just can't remember.

MR. UDICK: It's about two. Do you want to --

MR. KENDRICK: Yeah. Let's take a break.

(Whereupon, a recess was taken from 2:02 PM until 2:16 PM.)

Q Welcome back, Dr. Richardson.

During the break, did you discuss the substance of your testimony with counsel?

A No.
Q Thank you.
I'm going to jump down to Paragraph 112 of your second declaration here. And under Paragraph 112, you reference five factors from the board related to MPEG-1.
Do you see that?
A Yes.
Q Okay. And, then, you, in your opinion, at least two of the five factors are premised on incorrect interpretations, right?
A That's what I say in Paragraph 113.
Q And then you go on to discuss factor one, factor three, and factor five, right?
A Yes.
Q So it's three factors that you believe are premised on incorrect interpretations?
A I pointed out issues with all three of them.
Q And looking at factor one.
It's your opinion that the petitioner's argument reads out the requirement to

indicate one image frame of the independent sequence.

Do you see that in Paragraph 119?

A Yes.

Q Limitation 8A requires that an indication at least one image frame, not one image frame, right?

A Limitation 8A of the '321 Patent requires or states an indication of at least one image frame.

Q Okay. And in your analysis of under factor one, in your second declaration, you consider an example where an MPEG-1 decoder starts playback after the first I-picture of the closed_GOP, right? In Paragraph 120.

A Yes.

Q Okay. You don't consider any other situation for an MPEG-1 decoder other than the example talked about on 120, right?

MR. UDICK: Objection to form.

A Well, in my report as a whole?

Q No. In your discussion of factor one

from the board.
A If you look at my Paragraph 114:
If playback begins from a random point in the bitstream, an MPEG-1 decoder will discard all bits until it finds a sequence start code, a group of pictures start code, or a picture start code which introduces an I-picture.
And an I-picture, for example, within a GOP, which is perfectly possible and practical in MPEG-1 -- in an MPEG-1 bitstream, that would be an example of a first image frame in decoding order, which isn't preceded by a GOP header.
Q If the first image frame in decoding order was the first I-picture of the closed group of pictures, the GOP header would precede that first image frame, right?
A I think it's technically possible to start random access player at that first I-picture, picture start code, not the GOP start code. So I think the answer is, not

necessarily.
Q But it could be if the random access started the GOP start code, right?
A What could be?
Q If the random access started at the GOP start code for a closed_GOP, the first image frame in decoding order would be the first I-picture of that closed_GOP, right?
A The first image frame in decoding order after that GOP header would be an I-picture.
Q Okay. Then turning to factor three from the board on Paragraph 121. You agree that factor three may be true in some circumstances, right?
A There are some circumstances during MPEG-1 decoding where a closed_GOP flag and a GOP header would be decoded and evaluated -- sorry, yeah, decoded before the first image frame in that closed_GOP is decoded. But there would also be situations where that didn't happen, as I explained in Paragraph 121.

Q One of the situations where the closed_GOP flag or the GOP header would be decoded before the first image frame in a closed_GOP would be if the random access of the video sequence started at that closed_GOP header, right?
A Yes. But my point is that the board statement doesn't seem to allow for other -- these other possibilities that I mentioned.
Q Understood. Thank you.
Then for factor five, in your opinion, the first image frame decoded is not necessarily the image frame following the closed_GOP flag, right?
A Yes. As I discussed above.
Q But you agree that the first image frame decoded could follow the first image frame, or could follow the closed_GOP flag, right?
A That's not what the board seemed to state, but I agree that it could follow in some circumstances.
Q If random access or if the user decided

to start the video stream at the closed_GOP header, then the next image frame decoded would be the first image frame decoding of that first closed_GOP, right?

A The user doesn't necessarily decide to start decoding at a header. The user perhaps gives some sort of an input, such as asking a decoder to join a stream or to decode from a time point. If the first start code after that is such a GOP start code, then I think what you say would be correct. It wouldn't always be that situation.

Q Okay. In Paragraph 125 of your report, you state that:

> "Dr. Havlicek confirmed he did not consider decoder for MPEG-1 for everything that it teaches by way of technology."

Do you see that?

A Yes.

Q What do you mean by the statement "everything it teaches by way of technology"?

A So I'm referring to Exhibit 2005, Page 97. So I think what I was referring there was to -- yeah. I think I was referring to an MPEG-1 decoder that I -- start again.

I think I was intending to refer to a video decoder that complied with the MPEG-1 standard. Which, as I understand it, at least at this point, Dr. Havlicek's deposition is not what he was describing.

He was, as I understand it, requiring a decoder that does not practice the entirety of the -- or, actually, does not practice MPEG-1 as defined in the standard.

Q Let's turn to your analysis of Limitation 8B for MPEG-1.

And your analysis of Limitation 8B, which starts at Paragraph 129 of your declaration, you give an example of where, or a situation where MPEG-1 decodes starts decoding in a the middle of a GOP, right?

A Yes. For example, Paragraph 133, around

Paragraph 133 of my second declaration.
Q In your analysis of Limitation 8B in your second declaration did you consider whether example -- the Example A from the Figure D19 of MPEG-1 would meet that Limitation 8B?
MR. UDICK: Objection to form.
A That's a bitstream example not a decoder example or a decoding method.
Q It gives the example in display order and decoding order in Figure D19?
A I know what the figure is, but --
Q The limitation?
A The Limitation 8B, as I understand it, involves starting decoding a video sequence.
Q Correct.
A As part of a method of a decoding a video sequence according to the claim.
Q My question is: In your analysis of Limitation 8B, you didn't consider the example of A from Figure D19 of MPEG-1, right?

MR. UDICK: Objection to form.
A Isn't that the wrong way round? The question is, does MPEG-1 render obvious this, teach or render obvious this limitation, and MPEG-1 even in Figure D19 gives three different examples of groups of pictures and all of them are part of, if you like, or consistent with or intend to be consistent with the teachings of MPEG-1.
Q I appreciate that, but that wasn't my question.
My question is: In your analysis of Limitation 8B in your second declaration, you didn't consider the example of Figure A or Example A from Figure D19 of MPEG-1, right?
MR. UDICK: Objection to form.
A I never said I didn't consider that example. Where do I say that?
Q Where in your analysis do you consider Example A from Figure D19 for Limitation 8B?
A Maybe where I'm misunderstanding the question. But my analysis is what I've

written obviously. What I've considered -- sorry, what I've written is what I've written. It's my declaration, but I carried out a lot of analysis to get to this point, and I considered the entirety, for example, of the MPEG-1 standard in reaching the conclusion and opinions that I've written in my declaration.

Q Your declaration from Paragraph 129 through 137 does not discuss Example A of Figure D19 from MPEG-1, right?

A I don't directly talk through that example, but it's -- the entirety of Figure D19, for example, is part of MPEG-1's disclosure, which I talk about at length throughout my declaration.

Q MPEG-1 discloses that for the earliest picture and display order on each group of pictures the temporal_reference is reset to 0, right?

MR. UDICK: Objection to form.

Q On Page 44 of MPEG-1 if you'd like.

A I see that language on Page 44 of MPEG-1.
Q Okay. Then the temporal_reference syntax element is -- has a value that's determined by the encoder, right?
A The temporal_reference syntax element is syntax element in an MPEG-1 bitstream.
Q And it's a syntax element whose value is determined by the encoder?
A What do you mean by the encoder in that question.
Q Looking at Paragraph 141 of your second declaration, you state "Temporal_reference (temporal_reference syntax element) is a syntax element whose value is determined by encoder."
Do you see that?
A Okay. We're talking about the encoder in that paragraph, then yes, I agree with that.
Q Okay.
A I think my recollection is in this part

I'm talking about an encoder that produces an encoded bitstream that is compatible with the MPEG-1 standard.

Q Okay. And the encoder informs the decoder of the display order of decoded pictures, right?

MR. UDICK: Objection to form.

A I don't recall saying that.

Q The temporal_reference that the -- temporal_reference value that is set by the encoder informs the decoder of the display order of decoded pictures, rights?

A According to MPEG-1 the temporal_reference syntax element informs the decoder of the display order of decoded pictures.

Q How does the encoder inform the decoder of the display order of decoded pictures?

A Where do I say that?

Q I'm asking a question as to how the encoder does that in your experience as a POSITA?

A So the MPEG-1 Standard specifies a bitstream, specifies a decoding process. The decoding process is or can decode the temporal_reference syntax element and as I say in Paragraph 141, that syntax element informs the decoder of a display order of decoded pictures.

I think that's pretty clear.

Q So the decoder is -- determines the display -- informs the display order of decoded pictures by the value of temporal_reference syntax element?

A Just stick by -- you're moving my words around, just stick by the way I wrote it in the single sentence in Paragraph 141.

Q What do you disagree with about my question?

A I don't understand why it's necessary to move, why it's necessary to move the words around.

Q Well, I'm just asking is there something that you disagree with about the question

that's technically incorrect?
A I'm not sure. You can ask it again and I can try and work it out.
Q The decoder determines or is informed of the display order of decoded pictures by the value of the temporal_reference syntax element, do you disagree with that?
A Well, if an MPEG-1 decoder, decodes picture layer syntax structure, the second syntax element in that picture layer syntax structure is temporal_reference. It's assigned in the bitstream or in a compliant, in a bitstream compliant with MPEG-1 according to the standard, it's assigned in sequence to the pictures in display order. That's what that -- that means, that's semantics. Yeah, I don't know what one needs beyond that to understand. I think it's really, really clear to a POSITA.
Q Okay. And a decoder will read a temporal_reference value or a syntax element value -- let me start over.

The decoder will read the value of the temporal_reference syntax element from the bitstream, right?

MR. UDICK: Objection to form.

A An MPEG-1 -- a decoder that complies with MPEG-1 can decode temporal_reference syntax element from the bitstream.

Q When the decoder reads the temporal_reference value or the value of the temporal_reference syntax element from the bitstream, does it store that value?

MR. UDICK: Objection to form. Asked and answered.

A I think we talked about this and a decoder that conforms to MPEG-1, the MPEG-1 Standard, has to, yeah, has certain constraints placed on it by the standard. I don't recall reading a constraint as to what the decoder -- whether or not the decoder should store that value, and if so, how it should store it. When I say I don't recall the constraint, I mean, I don't recall the

constraint in the MPEG-1 standard.
Q In your opinion, does the '321 Patent teach a decoder resetting the -- let's see, actually, strike that question.
So we talked earlier, Exhibit 1006 we agreed to refer to as the Kim reference, right?
A I think so, yes.
Q And Kim discloses two types of image frames. Image frames without prediction or I-frames, and image frames with prediction from previously stored video frames or P-frames, right? You can look at Paragraph 80 of your second declaration.
A Kim describes a system that encodes and stores individual frames without prediction. I think it describes them as I-frames or I-pictures or with prediction, which I think describes as P-pictures or P-picture or P-picture frames or I-picture frames.
Q Then you state that Kim describes a system that encodes and stores video frames

with prediction from previously stored video frames in Paragraph 80 of your second declaration?

A Yes. Summarizing Kim of course.

Q Hmm. What do you mean by the phrase "previously stored video frames" in Paragraph 80?

A I think I was intending to provide a high level summary of, for example, Kim's disclosure of compress and generating I-picture frame data and leading Kim around Column 3 around Line 58, and storing such data, also the encoding side in Column 3 and I'm also intending to summarize the encoding Kim describes the second image data.

And storing of a -- Kim describes a P-picture data with, it's long-winded, but it's effectively describing the P-picture data being predicted from I-picture data encoded or compressed and store round about Columns 3 and 4 of Kim.

Q And Figure 5 of Kim discloses a logical

structure for the bitstream which shows an IAD and CAD, right?

A Yes. I think that's the way Kim describes.

Q And in your declaration you opine that the IAD disclosed in Kim can be 0 or absent, right?

MR. UDICK: Objection to form.

Q If you want to look at Paragraph 91 I believe is where you make --

A Thank you.

So Kim Column 4, Line 66, Column 5, Line 1, that's where I struggled. States that when the -- with reference to I-picture stored data, the address of the corresponding I-picture frame, my recollection is that relates to IAD or could relate to the IAD address is 0 or absent.

Q And because an IAD can take the value of 0 or absent, your opinion is that if a POSITA reading the disclosure can understand that the IAD can take any value, right?

A Well, Kim actually -- is silent about what the decoder should do with that IAD. If you look at Figure 4 of Kim and as I explain in my declaration, the address of the -- there is no address of a corresponding I-frame with I-frame data, there is no decoding or detecting such an address. Nothing in Kim discloses actually either looking at or using that address.

Q Thank you. But I don't think that was my question. Your opinion on Paragraph 91 states, "Since an I-picture and Kim does not require a reference to another I-picture, the IAD field of an I-picture can be 0 or absent. A POSITA reading that disclosure would understand that field can take any value."

What values could the IAD take in Kim besides the 0 or beyond a 0 or absent?

A So for example, I'm just thinking of a, an analogy, let's say I allocate memory for a variable in a C program, and I'm a lazy programmer and I never use that -- allocate

memory and I never use that value, it could take any value and actually that I think as I understand it can happen when you -- when memory is initialized if it's allocated but not set to a value it can be any value and it doesn't matter until it's set and used. So I think that the programming analogy would also cover a situation such as Kim Figure 5A where you have an I-picture shown on the left, this is on the top of my Page 33 of my second declaration. It's got an IAD field, which could be 0, it could be absent, and so -- it is absent, it's either not there or it doesn't matter what data is in it. That would be my interpretation.

Q Okay. If the IAD field is set to the value of 0 could it take another value?

MR. UDICK: Objection to form.

Incomplete hypothetical.

A Well, Kim actually says it can be 0 or absent. So Kim seems to not to care what's in it. If it's set to 0 it's set 0, but if

it's not set to 0 it's not set to 0. It doesn't make any difference in Kim's method as far as my reading of Kim's disclosure.

Q The IAD field in Kim has a value of 0, does it have a functional use?

A In an I-picture or elsewhere.

Q In an I-picture. Sorry, in a, yeah, let's go with an I-picture?

A I don't see any disclosure in Kim of the IAD field for an I-picture having any functional use whatsoever. For example, if I look at Figure 4, which is Kim's decoding flow chart as I recall, no where is such a field read or used.

Q Because Figure 4 -- because in your opinion, Figure 4 doesn't disclose reading or using the IAD fields, Kim doesn't discuss that or -- strike that question, excuse me.

Can a decoder begin decoding an image frame multiple times?

MR. UDICK: Objection to form.

Incomplete hypothetical.

A In what scenario?
Q In Kim. Can Kim's decoder begin decoding a single image frame multiple times?
A I don't really understand what you're asking.
Q Looking at the Figure 4 of Kim, when is image data first decoded?
A So Figure 4 is describing Kim Column 5, Line 8, as illustrating the flowchart depicting a reproducing method, a reproducing compressed intermittent data. And then Kim has specific language of, for example, Column 5, Line 17, the image data reproduced from the optical disc and it talks about reproducing the corresponding image data from the optical disc. It doesn't describe it as a decoding, but certainly once we get to the Kim Column 5, Line 32, he's talking about -- I'm going to refer to the flowchart at the same time, step S30, restore original data and Kim in Column 5 describes step S30 as decoding and it says, "The decoding process

of the decoder 111 involves reverse quantization and reverse transformation to restore the original image data prior to data compression carried out during recording." So at least at that point data is being decoded, that's image data is being decoded at step S30 in Kim's Figure 4.

Q But image data -- in your opinion, image data is first decoded at step S22, right? Paragraph 94.

A Thank you.

Q You're welcome.

A The first image -- I'm reading my Paragraph 94: "The first image data decoded is the image data for the frame that users selected, Step S22." That's what I state there. It's not quite what you asked me.

Q How is it different than what I asked you?

A You said it's -- something like it's my opinion that Step S22 is the first decoding step. That's not what I wrote here.

Q Step S22 is where the first image data decoded is the image data for the frame user the user selected, right?

MR. UDICK: Objection to form.

A No. You're misreading what I wrote here.

Q Okay.

A The first image data decoded is the image data for the frame the user selected at, and that's reproduced at Step S22. So that's referring to data that is reproduced at that point.

Q Is data reproduced at that point decoded?

A It is decoded at least at Step S30.

Q If a user selects a P-picture from Kim, would that P-picture be decoded before the I-picture on which it depends?

MR. UDICK: Objection to form.

A My Paragraph 95 -- and I think I talk about this in more detail later on -- if the requested image in Kim is a P-picture, the

P-picture data is first read and decoded. Then the corresponding I-picture is located, read and decoded.

So the P-picture data, which we talked about briefly, reproduced at Step S22, is decoded at least at Step S30 -- really, really clearly. And all of that happens before, for example, Step S35, reproduce the I-frame data at the detected address.

Q So the P-picture is completely decoded before the I-picture on which it depends?

A What do you mean --

MR. UDICK: Objection to form.

A Where did I say "completely decoded"? I don't think I did.

Q I'm just trying to understand what you mean by "decoded." Do you mean that decoded -- for decoded is starting decoding or does the P-picture is completely decoded at Step S30?

A Okay. So my sort of arguments or my opinions related to this are covered at least

in Paragraphs 154 onwards, my second declaration. And here, I'm specifically considering limitations of the challenged claims because that's my task or my objective here.

Limitation 8A requires -- or I'm sorry. I'm ahead or behind myself. 8B, 10B, 11B, starting around Paragraph 166 of my declaration. I'm analyzing the claimed limitations that requires part of the limitations, starting the decoding of the video sequence from said first image frame of the independent sequence. And as I conclude in Paragraph 172, in my view -- I'm going to paraphrase -- Kim does not disclose starting the decoding of the video sequence from said first image frame of the independent sequence when a user selects a P-picture.

So whether or not the decoding of the P-picture is or is not completely concluded by Step S35 doesn't change the fact that starting decoding does not start at the --

start with the I-picture data; it starts with the P-picture data at least at Step S30 of Kim's report.

Q I don't think that was my question.

So my question is, looking back at Paragraph 95 of your declaration, when you state that the P-picture data is first read and decoded, is it decode -- is all the P-picture data decoded before the I-picture? Or is it part of the P-picture data decoded before the I-picture?

A So Kim just -- Kim discloses what Kim discloses. I can't go beyond Kim's disclosure. But it's really clear for Figure 4, and I've excerpted part of it on Page 35 of my second declaration: "In Step S30, the image data corresponding to the user's request is transmitted and decoded in the decoder 111."

And that image data, as I recall, is the image data corresponding to -- compressed image data corresponding to a P-frame or

P-picture. So that's the data for that P-picture. And then there's a description of the decoding process applied to that P-picture data, so that's decoding of that P-picture data.

I don't see anything there that says it stops part way through the chunk of the data or the block of data or the set of data that has been reproduced from the optical disc.

Q So Kim discloses that its P-picture is completely decoded for the I-picture on which it depends is decoded, right?

MR. UDICK: Objection to form.

A I didn't say that. I don't necessarily agree with that. I don't think it actually matters to the conclusion that I've reached.

Q Okay. But you said you don't see anything in Kim that it starts part of the way through or that part of the data is decoded, right?

MR. UDICK: Objection to form.

A Well, I thought that's what you were

asking me, does it just decode part of the data, or does it decode all of it. So there's nothing that says -- for example, it doesn't say, "Reproduce and decode part of the data." It just says, "Reproduce the data corresponding to the frame."

And so reproduce the data corresponding to the user's request, decode the data. And that's talking specifically about the P-frame data that was stored on the optical disc.

Q In Paragraph 95 of your declaration, you state that the P-picture data is first read and decoded, then the corresponding I-picture is located, read and decoded, and the P-picture is reconstructed and then displayed.

What does "reconstructed" mean there?

A I think that's my shorthand for the process that is illustrated in Step S38 of Kim's Figure 4 and described in Column 5 of Kim, starting around Line 55. In Step S38 -- I'll paraphrase -- the P-frame is restored as

a full-screen image using the corresponding I-picture frame data and the moving vectors obtained.

And that's broadly consistent with what I would sometimes call reconstruction and what I think a person of ordinary skill would understand -- would find consistent with reconstruction in a -- for example, known video decoders or known image coders, decoders.

Q In Step 38 that -- the image data is reverse-transformed and recoded by decoder 111 and stored in memory as the original data corresponding to the user's request, correct?

A Where does it say that?

Q Column 5, Line 60 through 63.

A Oh, yes. I see that, yes.

Q Okay. So you agree that the image data produced at Step S38 is then sent to the decoder to be reverse-transformed and reverse-quantized and stored in memory as the original image data corresponding to the

user's request?
A That's interesting. I don't actually understand what Kim is talking about there. It doesn't make a lot of sense.
Q What doesn't make sense about it?
A Because by this stage in the flowchart, Kim has -- has decoded the image data corresponding to the P-picture at Step S30. It has decoded the I-picture in Step S36, the corresponding I-picture, and it's restored the P-frame corresponding to the user's request signal as a full-screen image.
Q So why would that full-screen image need to be reverse-transformed and reverse-quantized in decoder 111 at Step S38?
A It's a really good question. I don't actually know why that part of Kim was drafted that way.
Q As a person of ordinary skill, what would you interpret that to mean?
A Probably if I worked through it, I'd interpret it to be a rather strange piece of

drafting, writing this part. It doesn't make a lot of functional sense to me in the context of the rest of Kim.

Q Would it have to be sent back to the decoder for a reverse-transform or reverse-quantization because the P-picture that depends on the I-picture cannot be fully -- or cannot be completely decoded before the I-picture is decoded?

A I genuinely don't know why they wrote that, again, because they have -- at Step S30 the decoding process involving reverse-quantization and reverse-transformation. And Step S30 specifically states at Kim Line 5 -- sorry -- Column 5, Line 29 to 31, the image data corresponding to the user's request is transmitted and decoded in decoder 111. And if we look at the flowchart, that happens way before all of this Step S38 that we're discussing.

Q Okay.

A So I think a person of ordinary skill, to be honest, reading this part, unless I'm missing something, I think they would find that a confusing piece of drafting.

Q If you look at the bottom of Figure 2, for the D-mux 131, there's a line to the decoder 111.

Do you agree?

A Yes.

Q And then it looks like data from the decoder either goes to frame memory or it goes back to the DSP121, where it's then fed back into the decoder.

Do you see that?

A I see lines there. I don't know exactly what those lines are referring to.

I'm just having a look.

Q No problem.

A The decoder 111, according to Kim, around Line 18 performs a reverse-quantization and reverse-transformation of demultiplexed image

data, reproducing DSP121 generates image in the image frames based on P-picture data and moving vector output.

It says what it says, but yeah.

Q You said Line 18. What column of Kim were you looking at there?

A Oh, I'm sorry. Column 3. I thought I said Column 3.

Q I just want to make sure it's clear for the record. Thank you.

If you look at Column 3, starting at Line 20, the very last letter of Line 20, it says:

"A reproducing DSP121 for generating original image frames based on the P-picture data output of the decoder 111 a moving vector output for the D-mux 131".

Do you see that?

A Yes.

Q What does that mean to you?

A Just what it says. I don't think -- Kim

just describes it as a DSP. So DSP is a digital signal processor. I think a POSITA would understand that in this context. And says it's for generating original image frames based on P-picture data and moving vector output.

I think a person of ordinary skill in the art would say, okay, that's a statement for what it's for, but it doesn't really describe a process. And that's described later in Kim.

Q And just to level settle a little bit, Figure 4 of Kim, you understand there's two flows to that flowchart. Flow a bit of what Petitioners called the first path is where they're -- at Step S32 there's a "no." And then for the second flow path or second path at Step S32, it goes through the "yes."

Would you agree? And if that was --

A Would I agree with what?

Q That that's a discussion -- that Kim Figure 4 shows two flow paths.

A No. I wouldn't agree. I mean, you've also got a flow path from S23 back to the S22. I'm perfectly happy to agree that Petitioner describes two paths, but there's more than just those possible routes, if you like, through Kim.

Q That's fine. I think agreeing that Petitioner describes that would be fine.

And so the first -- I think the petitioner labels the first path as the "no." And if you want to look at Page 49 of the PDF of the petition. It's number Page 42 but PDF Page 49.

A Yeah. Okay.

Q And you will see that "no" in red, right?

A Yeah. Sorry. I just want to check that I'm on the same page because the figure appears at least twice.

Q It does?

A Okay. I'm on 57.

PDF Page No. 49 has an instance of an

annotated Figure 4, and there's a "no" path, yes.

Q And Petitioner refers to that as the first path, which when an I-frame -- or I-picture is requested by the user.

Do you agree with that?

A I agree that Petitioner refers to that path as a "no" path.

Q Would you agree --

A I think I remember that, and it's consistent with this diagram. Sorry.

Q No problem. Sorry about that.

And if you want to look at Paragraph 151 of your second declaration. I think that will clear things up 'cause my questions aren't very clear.

So the first path is when a user requests a stored I-picture frame, and the second path, the petitioner refers to when the user requests a P-picture frame?

A Yes.

Q I knew it was somewhere.

From looking at Limitation 8A, starting with this first path, it's your opinion that in the first path, the indication is not decoded at all, right?

A What's the indication?

Q Sorry. The IAD. Excuse me. Let me re-ask the question. Thank you.

In your opinion, in this first path the IAD value -- or IAD is not decoded?

A Yes.

Q And that's because the IAD would not be decoded until Step S34, right?

A The IAD of an I-picture or an I-frame or compressed I-frame would not be decoded in Kim, in my opinion. It's 0 or absent, as we've already discussed.

Q And at Step S -- if the user requests an I-picture at Step S30 of Kim, that I-picture would be decoded, right?

A So I-picture data would be decoded at Step S30 in that instance.

Q And that -- the decoding of that

I-picture wouldn't include the IAD of that I-picture?

A There's nothing in Kim that says that it would. You know, we've got to go with what Kim discloses, and Kim actually discloses in S34, which we may come to, we may not come to, decoding an IAD address, that's -- sorry. Just a second.

Q No problem.

A Step S34 -- yeah. I'll go with the way Kim describes it.

Kim describes Step S34, Column 5 -- and this is part of the P-picture path or the petitioner's "yes" path -- detect -- the controller detects the address of the corresponding I-picture frame, e.g., data stored in Area 3 of the bitstream B2 shown in Figure 6.

Figure 6, Area 3, which is actually a physical address rather than a logical address, but in the P-picture it's corresponding I-picture address. And even

giving the petitioner the benefit of the doubt in considering Figure 5A, that would correspond to -- for example, with one of the P-pictures decoding that second or IAD field or detecting that field.

There is no corresponding step to S34 in the "no" path. S34 is bypass. There is no detecting an address of a corresponding I-frame. So there's nothing in Figure 4 to suggest decoding that information. There's nothing that I see anywhere in Kim to suggest decoding that information.

As I said, that information Kim says is 0 or absent. So if it's absent, there's nothing to decode. It can be 0 or absent, and it's -- I think a POSITA would consider that equivalent of a "don't care" or something that is -- yeah, it's not necessary to decode it, so there's nothing that makes you infer that Kim decodes it, and there's nothing that discloses Kim decoding that data.

Q If the IAD is zero, would it be decoded?
MR. UDICK: Objection to form.
A At what stage in Kim?
Q At Stage S30.
A Stage S30 is restore original data from the reproduced data by reverse quantization and reverse transmission. It says nothing about IAD being decoded. I'm just looking at the S30 in Column 5 of Kim.
The image data is decoded, the decoding process involves reverse quantization. There's nothing about IAD. Why would it be decoded there? Why would -- I don't see how a POSITA reading that would imagine that IAD value of 000 would be decoded. And why would it be decoded?
It tells the decoder nothing. I mean, it's the basic principle of video or image coding that you don't -- you don't decode things that you already know that you don't need, because that's just an unnecessary step. Yeah. It's just not disclosed in Kim.

Q So Kim does disclose decoding I-picture data at Step S30, right?
A When -- if you like, the -- you take petitioner's no path, so in the case where you requested image is stored as an I-picture or I-picture data, then Step S30 data would be decoding I-picture data.
Q Would decoding I-picture data not include decoding the address of the I-picture?
A Which address?
Q Ask like a logical address first.
MR. UDICK: Objection to form.
A I think I talk about this.
I don't know why -- I don't see anything in Kim that indicates that would be necessary.
Q What about the physical address of the I-picture, would that be decoded at Step S30?
A That doesn't make sense because the data has already been reproduced. So, if I recall correctly, in Step S22, I think, reproduce

image data. The controller controls the -- I'm reading from Kim, Column 5, around Line 15:

"The controller controls the recording reproducing unit to reproduce the corresponding image data from the optical disc one."

So limited disclosure, but my understanding -- my reading of that is that by that time that step is complete, the data has been taken off the optical disc.

Yeah. Kim's system has the data.

Q How would Kim's system know where to go on the optical disc to get that data before Step S22?

A So I think a POSITA looking at Figure 6, again, the disclosure is limited, but Figure 6 does appear to disclose -- it's described as a data structure for the bitstream to be recorded and reproduced from the optical disc medium. I was reading from Kim, Column 4, Line 59.

If one looks at Kim, Figure 6, there's an example of what appears to be I-picture data stored on the optical disc, labeled B1, and it begins with a physical address. And then there's also an example of P-picture data stored on the optical disc, B2, which begins with a physical address.

So I think one could infer that -- yeah. Kim does say the data structure is divided to carry the physical address of the bitstream. And that's again Column 4, round about Line 62.

So the physical address would appear to be the most likely candidate for saying -- for indicating where -- where the data is on the disc.

Q And that physical address would have to be decoded so that Kim's decoder can determine where the data is on the disc?

A Where does it say that?

Q That's just a question I have for you.

A What do you mean by "Kim's decoder"?

Q How would Kim locate the I-frame data that the user's requesting using the physical address?

A Kim, Column 5, around Line 15:

"The controller 150 controls the recording/reproducing unit 140 to reproduce the corresponding image data from the optical disc one."

Now, it doesn't explicitly -- Kim's quite limited disclosure, let's say, it doesn't spell it out, but I think the best guest of a POSITA would be that controller 150 could use physical address shown in Figure 6 to locate and reproduce the corresponding image data.

MR. KENDRICK: Steve, do you want to take our break? We've been going for an hour.

MR. UDICK: Yeah. It works.

MR. KENDRICK: Off the record.

(Whereupon, a recess was taken from 3:20 PM until 3:26 PM.)

MR. KENDRICK: Back on the record.
Q Welcome back, Dr. Richardson.
During our short break, did you discuss your substance of your testimony with counsel?
A No.
Q So before our break we were talking about Kim's physical address and his logical address.
Kim discloses both a logical address and a physical address for a picture, right?
A So Kim, Figure 5A, Figure 5B, describe logical addresses. Kim, Figure 6, discloses physical addresses. And these are talked about in Kim, Column 4 and top of Column 5.
Q Looking at Kim, Figure 5, is the CAD in Figure 5 unique for each image frame?
A So if I consider Figure 5A, I see CAD or X001, X002, and so forth, up to X101. And those are all unique, as far as I can see, in this diagram.
Q Okay. The IAD in Figure 5A is not

unique for each individual frame, right?
A That's correct.
Q Okay.
A So, for example, P1 and P2 each have the same IAD value in Kim Figure 5A.
Q And that indicates that both of those P1 and P2 depend on the I-frame at CADX001, right?
A Kim describes each P-picture frame, so Kim, Column 4, round about Line 48, each P-picture frame has a current address CAD, and an I-picture reference address IAD, indicating the address of the corresponding or previous frame. That's what IAD indicates as the logical structure for a P-picture.
Q So figure 5A, image frame P1 and image frame P2 have the same IAD because they both depend on the I-frame, right?
A Those two pictures have the same IAD, which I understand from Kim would indicate that they both -- that the corresponding or previous I-frame for each of them is the

same.
Q Okay. And an MPEG-1 bitstream allows for multiple P-pictures to depend on the same I-picture, right?
MR. UDICK: Objection to form.
Incomplete hypothetical.
A In what sense?
Q In the sense that an MPEG-1 compliant bitstream could have a first P-picture that depends on an I-picture, and a second P-picture that depends on the same I-picture?
A I'm struggling with what you're asking me to imagine there.
Q If you look at Figure D19A on Page 87.
A Okay.
Q The P-picture -- or the P-pictures here can both -- can they both depend on that I as a picture?
MR. UDICK: Objection to form.
A So I'm assuming that you're referring to an I-picture with the temporal_reference number 0 and P-picture with

temporal_reference number 2 and a P-picture with temporal_reference number 4 in Figure D19 of MPEG-1.

So with that in mind, I don't understand what you mean by "depend" in that question or "depend on."

Q Could frame P2 -- can the reference picture for Frame P2 and the reference picture for Frame P4, both be -- sorry. Let me restart that question.

Can the reference frame for Picture P2 and the reference frame for Picture P4 both be I0, Picture I0?

A That's not my recollection of how MPEG-1 works.

Q Okay.

A Just to add to that answer. I'm not even sure that MPEG-1 uses the term reference picture. I can't recall.

Oh, it does. I'm sorry. Yeah.

MPEG-1, Page 26, reference pictures are the nearest adjacent I or P-pictures to the

current picture in display order.

So that seems to go against your suggestion.

Q And just to clarify your last question for the record, I think you said that that's not your recollection of how MPEG-1 works; is that right?

A I can't remember what I said.

Q Just looks like there may be a typo on the draft transcript. So I just want to correct that for the record.

A Okay. My recollection -- you were asking me something along the lines of for Figure D19 of MPEG-1, considering the first example A, could P2 and P4 both have a reference picture of I0. You said something along those lines, as I recall. And I said, that's not my recollection of how MPEG-1 works. And I think I meant I had in mind the decoding process specified in MPEG-1.

Q I appreciate that. Thank you. That's what I was looking for.

For Limitation 8B, if a user request an I-frame first -- or, sorry.

For Limitation 8B, if the user requests an I-frame in Kim, that I-frame would be the first image frame decoded in the group of pictures, right?

A Sorry. What's the group of pictures?

Q If you look at Kim's Figure 5A, it has an I-picture followed by one, two, three, four, five P-pictures, do you call that a group of pictures?

A I don't think Kim calls that a group of pictures.

Q Would you disagree that it is a group of pictures?

A Why is it a group of pictures?

Q Well, it's a group of -- it's an I-frame followed by a P-frame. Why would you disagree that that's a group of pictures?

A Kim's Figure 5A, as I understand it, is not an MPEG-1 bitstream. A group of pictures, as we've discussed at length, the

term that's used in MPEG-1, group of pictures I think is also used in other standards that I'm aware of. But it's not a term that I recall from Kim.

And Kim's Figure 5A is not describing an MPEG-1 bitstream. So I think it can actually be confusing to refer to it as a group of pictures.

Q Does Kim reference MPEG-1?

A Kim discusses, for example, in Column 4, paragraph beginning at Line 15, discusses that, so-called second image data can be output as a frame that has much less data than the I-picture frame, e.g. as a P-picture frame in MPEG.

And I don't recall that Kim actually clarifies what it means by MPEG-1, given the -- assuming the priority data from 1998, by that time in the public domain was at least MPEG-1, MPEG-1, possibly an early version of MPEG-4. So I'm not sure that Kim actually references MPEG-1.

Q Okay. For Limitation 8B, is it your opinion that Kim does not disclose -- sorry, let me re-ask that.

Is it your opinion that this limitation is not disclosed by Kim if an I-picture is requested by the user?

MR. UDICK: Objection to form.

A Sorry. Could you ask the question again. Just refresh my memory.

Q For Limitation 8B, is it your opinion that this limitation is not disclosed by Kim as an I-picture requested by the user?

A Yes. That's my opinion.

Q Why would this limitation not be disclosed by Kim if an I-picture is requested by the user?

A So at least for the reason I think I was intending to say in my Paragraph 171 that if the user requests an I-picture in Kim, that's all the user gets. There is no decoding of a sequence. There is no sequence.

Q Okay.

A And also for the reason that Kim doesn't show in Figure 5, in the structure Figure 5A, Figure 5B, Kim doesn't show an I-picture with an IAD that ever changes. So it doesn't identify -- couldn't identify anything.

That's paraphrasing what I said also in Paragraph 171.

Q Why can an IAD that never changes not identify anything?

A How does it identify -- I think a POSITA would see -- would struggle to see how it identifies an image if it's basically a constant.

Q It's just -- the IAD changes for, if you look at the P-picture in Figure 5A, right?

A But if we're considering Kim's disclosure -- as I understand it, I don't have a citation to the petition, but my recollection is that the petitioner appears to suggest that Kim could be used with only decoding I-frame or I-pictures. But if that's the case, the IAD never changes. So

it can't identify anything.
Q And you don't have a citation to where you believe petitioner appears to suggest that Kim would be redesigned to only decode I-frames, right?
A Well, I don't have a citation, but, for example, in petition on Page 51, says that, although Kim's disclosure of the first path, that is the I-picture path, is sufficient to teach or suggest this limitation, and I think that is referring to 8B, Limitation 8B.
I think that's consistent with what I said in Paragraph 171 of my second declaration.
Q And if a P-picture is selected, then you disagree that Limitation 8B is disclosed, right?
A Yes. Because -- well, we discussed this already. In that instance Kim's teaching started decoding from the P-picture, that's not the first image from the alleged independent sequence that the petitioner has

identified.
Q In your opinion, starting the decoding of the video sequence would refer to an image frame that Kim's decoder starts to decode even if that decoding is not completed?
A I think that's how a POSITA would read starting decoding. Starting decoding is starting decoding. I mean, it's pretty clear, isn't it.
Q A POSITA wouldn't read starting decoding of the video sequence to mean decoding the first image frame of that video sequence, not a partial decoding?
A Where does POSITA get that from, Kim or the -- yeah, I struggle to see where a POSITA would get, find that from Kim or from the '321 Patent. In fact, particularly from the '321 Patent.
Q Is there somewhere in the -- where in the '321 Patent does it state that starting a decoding of the video sequence can mean starting, but not completing decoding of an

image frame?

MR. UDICK: Objection to form.

A Starting decoding a video sequence, I think is really clear in the context of the '321 Patent, and it's my opinion a POSITA would not read it in the way you just suggested.

Q If it's really clear in the context of the '321 Patent, where does the '321 Patent state that?

A A POSITA reading the '321 Patent in my opinion would not lead to the, what I view as a contorted interpretation of what it means to start decoding something, start decoding of a video consequence.

Q Is there a difference between start decoding a video sequence and start decoding of an image frame?

A In what context?

Q The context of the '321 Patent.

A The '321 Patent, I don't recall if it talks about image frames. Oh, it does talk

about image frames, sorry. But it's, yeah, starting a decoding of the video sequence from said first image frame. What's the question?

Q Is there a difference between starting decoding of a video sequence and a starting decoding of an image frame in the context of the '321 Patent?

A Where in the '321 Patent?

Q Anywhere in the '321 Patent.

A Sorry, we're talking about the claims of the '321 Patent. Claim 8B, which we talked about a number of times, says, "Starting the decoding of the video sequence from first image frame," what are you asking me to analyze here?

Q I'm asking you to analyze whether in the context of the '321 Patent, the phrase starting the decoding of the video sequence is the same as starting the decoding of an image frame?

MR. UDICK: Objection to form and

scope.
A Where is that phrase in the patent, I don't recall it.
Q I'm not asking if that phrase is in the patent, I'm asking you to consider it in the context of the patent whether those two phrases would be synonymous?
A I've not been asked that question before. I've not been asked to consider that as I recall and I'm certainly not going to start considering that question now, unless you want me to take the rest of the time to do so.
Q Well, I just ask because in some of your previous answers you said a person would not lead to what you view as a contorted interpretation of what it means to start decoding something. And I'm asking does that something matter if it's start decoding of an image frame, does that matter verses what --
A Right, I'm sorry to interrupt you, carry on.

Q But does what comes after the term "starting the decoding" would that matter on how you interpret the term "starting the decoding"?

MR. UDICK: Objection to form and scope.

A I don't remember everything I said previously today, but it sounds to me as if you cut my answer short when you read it out there. If you did, could I please have my entire answer and I will try and answer your question whatever it is.

Q So I asked the question, if it was really clear in the context of '321 Patent, where does the '321 Patent state that? And you said, a POSITA reading the '321 Patent in my opinion would not lead to what I view as a contorted interpretation of what it means to start decoding something, start decoding of the produced video consequence.

And my question is: Is the phrase "start decoding something" the same as

starting the decoding of the video consequence?

A I have no idea. You're taking three words, no four words, out of context in one of my answers and trying -- with respect, I don't think that's a fair question because I finished off the question, yes, I kind of ran my words together, but I said start decoding of the video sequence, which is the claim term, the language of Claim term 8B starting the decoding of the video sequence in '321 Patent. So I think taking a few words out of context of my answer is just not a fair question.

Q I apologize, I'm not trying to take your words out of context or ask an unfair question.

My question is more of as a person of skill looking at the context of the '321 Patent, would the term "starting the decoding of the video sequence" be similar to --

Let me as that question differently.

Would a person of skill find the phrase "starting the decoding of the video sequence" and "starting the decoding of an image frame" to be synonymous?

A I don't recall the phrase "starting the decoding of an image frame" to appear in the '321 Patent, if it does please point me to it.

Q And outside of that phrase being in the '321 Patent, you can't answer my question?

A I'm not saying I can't answer your question, but as I already said I don't recall being asked to consider that question. I don't recall considering that question previously. I don't want to take the time to reach an opinion on something that I haven't yet considered in my two declarations. I don't know how much time it would require.

Q Okay. In your opinion, in Kim only the physical address in Figure 6 is decoded, right?

A Where do I say that?

Q Paragraph 178?
A I think this follows on from Paragraph 176, I'm just discussing what Kim does and does not disclose in my opinion. Paragraph 176, Kim does not disclose using -- when beginning the decoding of the user selector frame, Kim does disclose using the logical address. Instead, if anything, the physical address is used. And I think that's what I'm kind of alluding to in Paragraph 178. Only physical addresses are decoded and I think what I meant there was, if anything, is decoded in terms of an address, it's at most a physical address or in terms of Kim's disclosure or decoding of a user selected frame.
Q Does the term "e.g." mean example in your opinion?
MR. UDICK: Objection to form and scope.
A It can depending on the context.
Q All right. I want to turn back to MPEG

for a minute.
A Sure.
Q Does MPEG-1 disclose a NAL unit?
A Not that I recall.
Q Does the NAL unit as defined in H.264, indicate the first image frame in decoding order of an independent sequence?
MR. UDICK: Objection to form. Scope.
A I don't really understand the question.
Q If you look at H.264 in the NAL unit. Is the NAL unit an indication of the first image frame in decoding order of an independent consequence?
MR. UDICK: Objection to form and scope.
A So the NAL unit is not a syntax element, so I'm not quite understanding your question.
Q Is the NAL unit type an indication of a first image frame in decoding order of an independent sequence?
A In the H.264?
Q Yes.

A I haven't made that analysis or been asked to make that analysis.
Q Based on your experience with H.264, is the NAL unit types syntax element an indication of the first image framing decoding order of an independent sequence?
MR. UDICK: Objection to form and scope.
A So you're asking me the question as I understand it based on at least a phrase from an element of the Claim 8 of the '321 Patent, are you asking me in the context of the '321 Patent?
Q Yeah, in the context of the '321 Patent and based on your experience with H.264, is the syntax element NAL_unit_type an indication of the first image frame and decoding order of an independent sequence?
MR. UDICK: Objection to form.
Outside the scope of the witnesses purview.
A I haven't been asked to analyze or

compare in any way the '321 Patent with the H.264 Standards, so I haven't considered that question before.

Q Sitting here today, can you answer that question?

A Yes, well, I don't see why not.

Q Okay. What would the answer to that question be?

A I would have to take the necessary time to, first of all, get a clearly defined question, secondly consider the '321 Patent -- the claims of the '321 Patent, and context of the '321 Patent, and then consider the disclosures of the H.264 Standard, the NAL_unit_types syntax element -- so if the NAL_unit_types, the semantics of the NAL_unit_type how it might affect decoding in H.264. I can't think of reason why I couldn't consider all of those things, but it would take me time and I have no idea how long that would take.

Q Sorry, I want to turn back to Kim for

one more second, if you can get Limitation 8D of Kim.

A Okay.

Q In your opinion, Kim at most teaches that the encoder sets the identifier value, Kim never teaches resetting the decoder resetting identifier value, right?

A Okay. In Paragraph 180 of my second declaration I say pretty much that. I mean, I think my recollection is where I say the identifier value, I'm referring to the alleged identifier value or the identifier value of IAD or IAD equals 000 identifier -- identify proposed by petitioner as satisfying the claimed identifier value.

Q And in Paragraph 184 your opinion is that --

A Sorry, sorry. I do apologize. I think I've mixed things up in my head, I'll restate what I just said.

Q Yeah, no problem, go ahead.

A Yeah, thanks.

Yeah, so Paragraph 174 of my second declaration petitioner points to the logical addresses in Figure 5A of Kim as the identifier values. So I don't agree that these logical addresses are -- satisfy the claims identifier values. That being said, my Paragraph 180 when I say Kim never teaches a decoder resetting any identifier values, I think I had in mind the identifier value that petitioner had proposed.

Q Thank you for that clarification.

MR. KENDRICK: Pass the witness.

MR. UDICK: Just take a couple minutes.

(Whereupon, a recess was taken from 4:02 PM until 4:03 PM.)

MR. UDICK: We have no questions either. All right.

MR. KENDRICK: Great. Think we're done. Thank you so much. Appreciate it.

(Whereupon, this examination was

concluded at 4:03 PM.)

_____

DR. IAIN RICHARDSON

Subscribed and sworn to
before me on this ____ day
of _____, _____.

_____
Notary Public

I N D E X

WITNESS: DR. IAIN RICHARDSON

EXAMINATION BY PAGE

MR. KENDRICK 5

C E R T I F I C A T E

I, KIARA MILLER,
a Shorthand Reporter and Notary Public of the State of New York, do hereby certify:

That the witness whose examination is hereinbefore set forth, was duly sworn or affirmed by me, and the foregoing transcript is a true record of the testimony given by such witness.

I further certify that I am not related to any of the parties to this action by blood or marriage, and that I am in no way interested in the outcome of this matter.

Kiara Miller

KIARA MILLER

A
ability
113:19
able
60:6, 116:13, 126:17, 127:18, 127:22, 129:6, 129:13, 129:16, 133:4, 138:8, 138:16, 143:15, 177:13, 178:22, 179:14, 186:20, 198:7, 198:9, 216:22, 223:18, 224:11
above
130:13, 134:16, 135:20, 135:21, 161:9, 244:15
absent
216:21, 257:6, 257:18, 257:20, 258:14, 258:18, 259:12, 259:13, 259:21, 277:15, 279:14, 279:15
absolute
59:4
absolutely
104:21
abstract
91:20
accept
59:17
accepted
59:15
access
79:16, 82:7, 96:5, 96:8, 96:11, 96:13, 96:18, 96:20, 97:2, 97:20, 99:10, 99:18, 100:10, 123:20, 123:21, 130:2, 130:4, 137:7, 139:17, 139:20, 139:22, 142:8, 142:12, 184:19, 184:20, 184:22, 185:9, 185:15, 186:12, 187:10, 188:4, 190:4, 190:5, 190:6, 190:9, 190:10, 190:12, 191:3, 191:8, 191:12, 191:14, 191:17, 191:22, 192:3, 242:20, 243:2, 243:5, 244:4, 244:22
accompanying
196:10
according
40:2, 40:7, 41:1, 46:14, 47:21, 50:18, 51:14, 55:13, 58:13, 71:11, 79:3, 109:14, 116:4, 118:21, 138:4, 138:8, 148:7, 151:8, 152:4, 152:5, 159:6, 161:1, 173:22, 181:19, 186:14, 187:11, 189:11, 190:1, 218:3, 218:8, 226:15, 228:6, 234:14, 247:18, 251:13, 253:14, 272:19
accordingly
54:6
accuracy
42:2, 42:7, 158:21
accurately
49:15
acronym
25:12
across
5:16, 45:22, 77:8, 82:5, 86:2, 86:3, 224:4, 224:5, 235:4
action
310:14
activated
105:6, 105:11, 106:3, 106:8, 106:14, 106:18, 106:22
actual
208:12, 212:5, 217:3
actually
13:3, 17:4, 38:17, 49:14, 72:20, 76:2, 77:21, 79:2, 116:10, 138:18, 147:4, 152:10, 168:19, 171:20, 174:11, 186:21, 189:18, 216:19, 246:13, 255:4, 258:1, 258:8, 259:2, 259:20, 267:15, 270:2, 270:17, 278:5, 278:19, 291:6, 291:16, 291:22
add
158:10, 288:17
additional
140:20
address
257:15, 257:18, 258:4, 258:5, 258:7, 258:9, 264:9, 278:7, 278:15, 278:20, 278:21, 278:22, 279:8, 281:9, 281:11, 281:12, 281:18, 283:4, 283:7, 283:10, 283:13, 283:17, 284:3, 284:13, 285:8, 285:9, 285:10, 285:11, 286:11, 286:12, 286:13, 301:20, 302:8, 302:9, 302:14
addresses
285:13, 285:14, 302:11, 307:3, 307:5
adequate
64:21
adjacent
288:22
advantage
101:8, 102:3
advice
20:12
affect
52:17, 97:10, 305:17
affecting
9:9
affirmed
310:9
after
5:1, 74:9, 97:9, 140:7, 146:4, 149:10, 187:9, 192:9, 208:3, 225:20, 226:7, 226:10, 226:14, 227:17, 238:8, 239:9, 241:14, 243:10, 245:9, 299:1
again
23:16, 27:15, 29:16, 30:6, 30:17, 34:22, 40:12, 47:16, 66:8, 69:6, 79:10, 80:15, 84:6, 103:11, 104:19, 106:7, 109:12, 111:15, 121:20, 124:18, 133:19, 135:3,

138:12, 168:22, 187:4, 209:9, 211:21, 214:2, 226:4, 246:5, 253:2, 271:11, 282:17, 283:11, 292:9
against
1:10, 2:11, 224:5, 289:2
ago
59:8, 63:17, 75:20, 76:3, 96:3
agree
12:20, 13:7, 31:4, 34:10, 35:21, 36:13, 37:10, 38:6, 41:19, 44:19, 45:19, 46:16, 48:1, 57:22, 72:12, 73:12, 81:7, 91:21, 92:21, 98:5, 98:10, 98:15, 100:8, 106:11, 113:9, 123:5, 130:16, 135:1, 141:18, 142:10, 146:15, 146:20, 147:11, 149:4, 149:14, 156:19, 160:11, 160:15, 169:4, 170:2, 170:4, 170:17, 172:19, 190:15, 205:14, 205:16, 207:6, 209:17, 213:5, 213:11, 214:8, 215:1, 223:19, 224:11, 227:16, 233:2, 243:12, 244:16, 244:20, 250:19, 267:15, 269:18, 272:8, 274:19, 274:20, 275:1, 275:3, 276:6, 276:7, 276:9, 307:4
agreed
255:6
agreeing
275:7
agrees
83:17, 89:10
ahead
265:7, 306:21
all
7:22, 17:14, 17:17, 21:19, 22:12, 22:16, 22:21, 23:18, 31:13, 38:18, 49:21, 51:18, 64:12, 64:13, 68:3, 76:2, 82:1, 85:5, 97:6, 119:18, 129:17, 130:21, 131:13, 138:17, 146:18, 152:17, 152:21, 161:10, 171:15, 182:2, 182:8, 189:20, 193:4, 193:13, 240:18, 242:5, 248:7, 264:7, 266:8, 268:2, 271:20, 277:4, 285:20, 292:20, 302:22, 305:10, 305:19, 307:18
all-motion-compe-nsated
220:6, 222:8
alleged
294:21, 306:12
allocate
258:20, 258:22
allocated
259:4
allow
244:8
allowed
48:17, 171:19, 172:1
allows
88:8, 287:2
alluding
302:10
along
44:12, 59:18, 289:13, 289:17
already
52:15, 85:4, 95:20, 120:17, 123:14, 277:16, 280:20, 281:21, 294:19, 301:12
also
8:19, 9:18, 12:15, 18:15, 20:2, 27:16, 28:21, 33:6, 33:8, 35:4, 35:6, 39:3, 40:12, 49:7, 52:5, 60:17, 71:14, 85:9, 99:2, 100:8, 142:1, 153:6, 166:11, 166:20, 167:2, 174:4, 176:6, 180:4, 187:15, 189:2, 214:5, 218:19, 228:19, 230:3, 243:20, 256:13, 256:14, 259:7, 275:2, 283:5, 291:2, 293:1, 293:6
alston
4:4
although
294:8
always
19:2, 211:1, 230:19, 231:9, 231:16, 245:12
amazon
1:4, 2:4, 5:10, 12:22, 20:20, 21:4, 21:9, 25:18, 26:1, 26:2, 28:11, 30:1, 202:13
amazon's
12:15, 25:16
ambiguous
110:20
americas
2:20
among
44:2, 184:4
amount
239:1
analogy
258:20, 259:7
analysis
20:12, 25:16, 200:2, 241:11, 246:15, 246:17, 247:2, 247:19, 248:12, 248:19, 248:22, 249:4, 304:1, 304:2
analyze
297:16, 297:17, 304:22
analyzing
265:9
annex
190:19, 213:6, 213:13, 214:3, 214:4, 215:2, 215:13, 216:20, 217:16, 219:2
annotated
7:7, 7:9, 276:1
another
19:18, 49:8, 58:15, 108:15, 134:2, 170:19, 172:5, 176:22, 177:13, 180:19, 180:20, 185:13, 209:4, 218:5, 237:9, 237:19, 258:13, 259:17
answer
8:18, 9:3,

18:1, 44:14, 44:16, 49:14, 49:20, 54:7, 56:8, 58:18, 59:18, 62:19, 63:8, 63:10, 67:21, 68:17, 75:19, 76:5, 92:11, 94:1, 96:16, 103:15, 113:18, 116:13, 153:20, 154:8, 176:3, 186:8, 187:4, 188:2, 192:1, 198:15, 198:18, 199:1, 234:16, 242:22, 288:17, 299:9, 299:11, 300:13, 301:10, 301:11, 305:4, 305:7
answered 66:9, 85:13, 122:18, 148:19, 203:12, 239:5, 254:13
answering 15:20, 49:18
answers 8:13, 33:21, 42:13, 92:8, 298:15, 300:5
anticipates 28:12
any 5:22, 7:6, 7:9, 7:10, 9:6, 9:9, 10:1, 15:3, 18:10, 20:12, 20:15, 20:19, 21:2, 21:8, 22:17, 23:7, 31:12, 31:16, 31:20, 32:7, 46:4, 46:5, 59:17, 82:22, 84:9, 85:16, 91:8, 91:10, 92:8, 93:8, 93:13, 114:20, 114:21, 137:20, 163:13, 170:1, 171:6, 177:11, 179:19, 180:9, 182:20, 204:16, 204:17, 206:15, 207:2, 208:3, 210:7, 214:8, 232:1, 232:22, 237:6, 241:17, 257:22, 258:16, 259:2, 259:5, 260:2, 260:9, 260:10, 305:1, 307:8, 310:13
anything 8:10, 13:9, 23:3, 23:7, 23:17, 56:14, 79:2, 79:12, 90:3, 140:16, 178:7, 199:2, 267:6, 267:18, 281:15, 293:5, 293:9, 294:1, 302:8, 302:13
anywhere 89:15, 94:4, 123:20, 157:8, 160:5, 279:11, 297:10
apart 42:8, 143:21
apologies 105:21, 170:11
apologize 121:22, 218:10, 229:13, 300:15, 306:18
appeal 1:2, 2:2
appear 192:4, 282:18, 283:13, 301:6
appears 10:4, 98:12, 167:2, 203:19, 236:15, 275:19, 283:2, 293:19, 294:3
applications 91:14, 92:8, 191:2
applied 25:21, 194:4, 194:10, 267:3
applies 25:16, 65:11, 235:3
apply 194:20, 195:5
appreciate 34:4, 113:20, 248:10, 289:21, 307:20
appropriate 224:7
architecture 162:8, 163:1, 163:22
architectures 162:17, 163:19
area 126:18, 128:2, 128:3, 278:17, 278:19
aren't 40:10, 116:13, 276:16
arguably 39:4
argue 30:22, 100:11
argues 26:14, 28:6, 29:5, 30:1
argument 52:18, 146:19, 146:21, 155:16, 221:3, 240:22
arguments 264:21
around 9:16, 10:11, 19:6, 75:20, 135:12, 203:2, 210:10, 210:11, 215:22, 216:9, 221:8, 225:8, 246:22, 252:14, 252:20, 256:11, 256:12, 265:8, 268:21, 272:20, 282:2, 284:4
arrow 126:21, 136:5, 142:5
arrows 135:12, 141:6
art 12:15, 25:8, 81:16, 119:5, 162:18, 189:10, 190:11, 198:8, 228:14, 229:3, 274:8
article 102:1, 102:5
articulate 189:9
articulated 180:1
articulates 180:14
articulating 181:7
aside 27:20
asked 20:15, 59:8, 63:12, 88:21, 103:7, 109:17, 112:20, 113:1, 116:8, 117:21, 118:7, 120:18, 121:4, 122:17, 131:20, 192:7, 200:1, 209:10, 211:22, 224:8, 227:13, 254:12, 262:17, 262:18, 298:8, 298:9,

299:13, 301:13, 304:2, 304:22
asking
23:11, 56:2, 56:3, 63:5, 75:21, 95:10, 108:10, 108:11, 119:18, 119:20, 120:4, 121:3, 121:6, 135:19, 136:2, 137:18, 145:7, 156:14, 159:17, 159:18, 194:22, 195:2, 198:15, 198:18, 199:21, 203:22, 206:18, 209:6, 220:10, 220:14, 220:17, 222:18, 226:12, 231:2, 245:7, 251:20, 252:21, 261:5, 268:1, 287:12, 289:13, 297:15, 297:17, 298:4, 298:5, 298:18, 304:9, 304:12
aspect
40:1, 62:17
aspects
40:10, 76:14, 99:17
asserting
28:12
assigned
164:18, 165:6, 253:12, 253:14
associated
10:21, 11:9, 154:12, 156:8, 194:2
assume
87:4, 105:1, 155:4, 155:16, 155:17, 156:15, 157:10, 212:9
assuming
72:8, 287:20, 291:18
assumption
155:14
atlanta
4:6
attempt
128:16, 128:21, 132:13, 138:12, 185:22, 186:11, 186:12, 186:18, 186:19, 187:12, 190:2, 230:13
attempted
129:12
attempting
33:21, 113:18, 130:19, 132:11
attempts
127:16, 138:15, 138:21, 187:2, 187:7, 193:19
attention
49:8, 162:10
attorney
9:1, 24:1, 24:2
attorneys
5:10, 8:4
audible
8:13
audio
27:21
author
69:1
available
82:5, 127:13, 128:18, 131:14, 140:4, 187:15, 200:7, 200:9, 211:17, 213:15, 213:22
avenue
2:19
avoided
104:9, 104:16, 105:3
avoiding
97:13, 97:14
aware
7:8, 9:8, 20:5, 20:11, 38:10, 38:11, 38:16, 38:19, 39:12, 45:22, 68:2, 145:8, 179:2, 291:3

B

b-frame
200:12
b-picture
200:22, 201:1, 201:2, 201:3, 201:4, 206:5, 218:6
b-pictures
169:8, 171:16, 206:8, 206:15
b1
283:3
b2
278:17, 283:6
b4
186:13, 192:14, 193:20, 197:12, 198:13
b5
188:7
b7
186:19, 186:21, 187:1, 187:3, 187:13, 187:16, 189:17
back
22:12, 26:15, 42:12, 56:22, 59:6, 62:14, 62:21, 63:9, 63:22, 65:2, 70:10, 78:3, 86:2, 93:6, 96:2, 97:17, 99:19, 100:17, 102:6, 103:17, 112:13, 114:4, 114:9, 124:1, 130:12, 135:7, 147:10, 157:13, 182:7, 182:8, 219:16, 239:20, 266:5, 271:4, 272:12, 272:13, 275:2, 285:1, 285:2, 302:22, 305:22
background
91:20, 98:13, 99:15, 146:10, 146:11, 147:10
backward
80:7
backwards
144:12, 144:19, 145:16, 145:18, 145:22
bad
186:22, 187:2
ballpark
91:12
based
21:17, 22:5, 23:9, 64:3, 94:21, 103:15, 113:19, 172:4, 222:4, 273:2, 273:15, 274:5, 304:3, 304:10, 304:15
basic
135:22, 280:18
basically
227:3, 293:12
basis
221:12
because
28:15, 31:12, 36:8, 40:16, 41:22, 42:17, 43:6, 67:22, 71:8, 78:15, 80:15, 94:15, 129:7, 130:4, 135:11, 137:14, 137:22, 140:3, 140:13, 158:7, 186:6, 187:14,

188:5, 191:19, 209:9, 222:12, 234:7, 257:19, 260:15, 265:4, 270:6, 271:6, 271:11, 275:18, 277:11, 280:21, 281:20, 286:17, 294:18, 298:14, 300:6
become
28:1, 28:2
been
5:2, 14:1, 14:3, 17:12, 18:14, 24:15, 24:17, 24:19, 24:21, 35:2, 56:16, 85:6, 90:19, 92:1, 120:13, 120:17, 121:4, 127:11, 129:16, 174:16, 176:14, 178:17, 180:2, 181:4, 200:1, 207:21, 208:8, 208:22, 209:15, 210:1, 210:5, 210:13, 210:20, 212:10, 212:14, 224:8, 267:9, 281:21, 282:11, 284:17, 298:8, 298:9, 304:1, 304:22
before
1:2, 2:2, 2:21, 5:8, 8:20, 9:12, 14:8, 24:18, 54:8, 57:5, 75:19, 85:5, 121:1, 128:7, 131:7, 131:22, 132:10, 139:11, 146:3, 164:8, 167:22, 182:16, 184:13, 193:5, 195:20, 243:18, 244:3, 263:17, 264:8, 264:11, 266:9, 266:11, 271:9, 271:20, 282:14, 285:7, 298:9, 305:3, 308:9
beforehand
5:19
begin
85:4, 130:5, 142:9, 142:20, 144:11, 147:5, 206:4, 260:19, 261:2
beginning
15:19, 84:20, 138:7, 144:4, 146:1, 167:13, 218:17, 291:11, 302:6
begins
99:4, 142:14, 144:10, 193:11, 218:2, 218:16, 242:3, 283:4, 283:7
behalf
2:19, 3:3, 3:13, 4:3
behind
265:7
being
16:19, 20:5, 34:1, 46:7, 71:12, 84:17, 88:21, 106:1, 108:1, 112:16, 116:8, 116:12, 117:21, 128:17, 150:15, 161:15, 163:7, 169:21, 174:15, 193:17, 212:19, 256:19, 262:5, 262:6, 280:8, 301:9, 301:13, 307:6
believe
6:18, 6:22, 10:14, 57:18, 60:11, 90:18, 112:14, 202:4, 202:22, 203:12, 240:16, 257:10, 294:3
below
124:13, 124:20, 126:10, 126:20, 134:2, 134:11, 134:14, 135:7, 136:4, 139:10, 140:11, 141:1, 141:20, 201:21
benefit
279:1
besides
258:18
best
8:17, 8:19, 26:6, 113:18, 284:11
better
46:19, 101:21
between
17:22, 20:20, 21:3, 21:9, 37:2, 168:13, 211:9, 212:1, 296:16, 297:5
beyond
23:12, 35:14, 93:21, 132:15, 253:18, 258:18, 266:13
big
102:19, 176:13
binder
6:16, 13:16
binders
6:2, 6:4, 6:14, 16:17, 17:3, 26:18
bird
4:4
bit
9:21, 17:1, 53:12, 103:14, 107:12, 116:9, 128:14, 130:22, 156:3, 161:4, 165:21, 166:3, 174:4, 174:14, 181:11, 192:7, 208:2, 213:2, 213:3, 274:12, 274:14
bits
193:13, 242:5
bitstreams
157:17, 181:17
blindingly
144:21
block
267:8
blocks
133:9, 133:13
blood
310:14
board
1:2, 2:2, 222:3, 222:15, 222:16, 222:19, 223:11, 240:6, 242:1, 243:12, 244:7, 244:19
board's
223:1, 223:17, 223:21, 224:13
book
9:20, 10:1, 10:4, 10:7, 60:14, 60:19, 60:20, 61:3, 61:4, 61:14, 61:16, 63:3, 63:4, 75:4, 76:4, 76:6, 76:8, 77:14, 78:8, 78:11, 237:13
books
10:5, 60:13, 69:2, 73:17, 237:13
both
12:6, 12:12,

15:8, 16:21, 18:16, 94:16, 134:19, 219:6, 230:12, 285:10, 286:6, 286:17, 286:21, 287:17, 288:9, 288:12, 289:15
bottom
13:21, 14:17, 90:1, 124:2, 221:22, 222:2, 272:5
bound
83:19, 84:9, 86:18, 87:1, 87:9
boundaries
43:19
bounds
42:2, 42:7, 85:17, 86:4, 158:21
box
126:22
bracket
105:2
break
7:21, 8:2, 56:18, 57:1, 57:5, 61:10, 113:22, 114:5, 176:9, 181:20, 182:3, 182:11, 182:16, 239:17, 239:21, 284:17, 285:3, 285:7
breaks
7:20, 8:3
brief
103:5, 103:6, 162:14
briefly
5:8, 174:6, 192:11, 264:5
bring
58:18
broad
35:15, 125:18
broadly
26:12, 42:20, 54:14, 125:3, 147:17, 163:6, 170:18, 188:11, 269:4
broken
14:13
browser
9:18, 61:2, 75:4
browsing
97:8, 97:10, 101:10
bypass
279:7

C

ca
3:6
cad
257:2, 285:16, 285:18, 286:11
cadx
286:7
call
6:20, 7:2, 12:7, 41:12, 146:21, 146:22, 147:2, 162:20, 269:5, 290:10
called
14:7, 15:11, 69:21, 109:11, 145:20, 171:10, 174:5, 205:17, 217:6, 217:7, 274:15
calls
91:5, 199:20, 202:10, 290:12
camino
3:5
can't
13:9, 17:11, 18:13, 19:2, 23:17, 53:8, 54:12, 57:16, 59:12, 61:6, 62:10, 65:22, 68:11, 74:10, 74:18, 104:2, 173:21, 181:18, 186:5, 187:13, 205:21, 231:11, 236:12, 239:13, 266:13, 288:19, 289:8, 294:1, 301:10, 301:11, 305:18
candidate
283:14
cannot
8:3, 9:6, 99:7, 126:4, 131:17, 137:20, 140:12, 271:7, 271:8
capabilities
190:10
capability
173:2
capable
48:8, 79:4
capture
35:7, 62:18
captured
89:14
car
124:6, 141:10
care
259:21, 279:17
carefully
212:9
carried
39:15, 249:3, 262:4
carries
75:7, 232:20
carry
37:5, 37:7, 37:16, 37:18, 37:22, 39:13, 39:22, 40:3, 48:5, 59:1, 193:10, 203:6, 283:10, 298:21
carrying
48:8
cars
130:14
carter
3:11
case
1:9, 2:9, 161:2, 189:12, 193:19, 201:22, 227:14, 281:4, 293:22
catching
105:22
cause
186:2, 276:15
caused
41:18, 128:14
caution
92:6
cd
55:8, 55:10, 55:13, 56:1, 56:2, 145:14
cds
55:3
certain
49:4, 50:2, 52:12, 53:4, 53:5, 53:7, 54:9, 54:10, 54:11, 67:8, 82:7, 92:3, 112:1, 132:5, 132:8, 132:9, 132:10, 156:15, 156:20, 157:9, 157:10, 160:1, 204:1, 239:1, 254:16
certainly
7:8, 23:18, 24:8, 43:6, 82:1, 87:6, 94:12, 152:14, 160:18, 160:21, 163:12, 172:6, 172:8, 175:19,

191:21, 224:7, 237:11, 237:16, 261:17, 298:10
certify 310:5, 310:13
challenge 130:2, 231:11, 231:14
challenged 13:1, 13:5, 13:12, 230:1, 230:16, 231:4, 231:7, 265:3
challenges 130:3, 139:21
challenging 202:6
change 54:17, 55:20, 56:1, 56:6, 56:15, 57:6, 57:12, 57:13, 59:10, 59:20, 65:3, 65:19, 66:4, 66:18, 67:16, 68:5, 73:2, 74:3, 74:12, 265:21
changed 75:1, 96:15, 174:13
changes 293:4, 293:8, 293:14, 293:22
changing 74:7
chapter 62:6, 62:7, 78:13, 78:14
characteristics 177:21
characterization 59:15, 59:17, 190:7
chart 260:13
check 15:4, 275:17
checking 13:3
choice 53:8, 54:1, 54:3, 54:13, 54:15, 128:19
choices 51:1, 51:4, 53:5, 54:9, 54:10
choose 128:18, 128:21, 190:18
chose 52:6, 147:20
chosen 212:9
choses 52:1, 128:16, 138:14
chrome 9:18, 61:1
chunk 267:7
circumstance 157:9
circumstances 243:14, 243:15, 244:21
citation 98:12, 100:18, 143:22, 232:17, 293:18, 294:2, 294:6
citations 98:4, 98:7, 232:22
cite 99:21, 232:1, 232:3, 232:10, 232:11, 232:18, 232:19, 236:2, 236:5, 237:9, 237:19
cited 21:19, 69:2, 112:21
claim 13:19, 14:6, 14:14, 15:7, 16:11, 30:4, 30:9, 30:22, 31:2, 31:7, 31:8, 31:9, 31:14, 31:20, 31:22, 89:21, 90:3, 90:8, 90:11, 91:4, 93:6, 93:7, 93:11, 93:20, 94:4, 94:6, 94:8, 94:22, 95:4, 95:16, 95:18, 109:4, 109:10, 109:15, 109:18, 111:2, 112:13, 114:9, 114:17, 116:11, 117:20, 118:6, 119:15, 119:17, 119:21, 119:22, 120:7, 121:4, 121:8, 121:10, 121:17, 121:20, 122:6, 123:6, 123:7, 123:19, 123:20, 123:22, 143:4, 143:5, 143:12, 143:13, 143:21, 183:4, 184:12, 191:6, 191:13, 192:2, 192:5, 194:4, 194:10, 195:4, 195:13, 195:16, 195:17, 195:19, 196:2, 197:3, 197:11, 198:10, 198:20, 220:11, 220:12, 220:16, 221:2, 221:9, 224:2, 224:21, 238:3, 238:6, 238:12, 239:2, 239:7, 247:18, 297:12, 300:9, 300:10, 304:11
claimed 143:4, 222:6, 265:9, 306:15
claims 13:1, 13:4, 13:7, 13:11, 13:12, 13:13, 26:10, 28:7, 28:12, 28:20, 29:6, 29:11, 91:19, 103:14, 115:6, 121:5, 228:15, 229:19, 230:1, 230:17, 231:1, 231:4, 231:6, 231:7, 231:11, 231:15, 238:19, 265:4, 297:11, 305:12, 307:6
clarification 16:2, 34:3, 113:1, 224:16, 307:11
clarified 113:4
clarifies 291:17
clarify 289:4
clause 80:12, 213:20
clean 7:15
clear 8:9, 33:20, 95:9, 107:19, 141:14, 168:4, 177:6, 215:12, 252:8, 253:19, 266:14, 273:9, 276:15, 276:16, 295:9, 296:4, 296:8, 299:14
clearly 49:1, 135:18, 264:7, 305:10
close 58:18, 227:2

closed
166:8, 166:9, 174:9, 174:18, 175:7, 178:12, 180:5, 187:16, 206:8, 207:7, 207:11, 207:14, 207:19, 209:8, 209:12, 210:4, 211:4, 211:10, 211:13, 211:20, 212:2, 213:5, 213:8, 214:15, 214:16, 214:20, 214:22, 219:20, 220:1, 220:5, 220:18, 222:12, 225:13, 235:21, 242:16
closed-gop
181:3, 225:15
closed_gop
165:16, 165:20, 166:2, 166:5, 166:6, 166:13, 174:5, 174:14, 175:5, 175:11, 176:4, 176:13, 176:19, 177:8, 177:22, 179:8, 179:17, 179:18, 179:21, 180:12, 180:17, 181:12, 181:19, 182:17, 182:19, 202:11, 202:14, 203:3, 203:11, 203:15, 203:19, 204:6, 204:9, 205:3, 205:12, 205:18, 206:17, 208:5, 208:7, 208:13, 208:15, 209:6, 209:14, 209:19, 210:15, 210:18, 210:19, 211:3, 211:6, 212:7, 212:16, 212:22, 213:2, 225:1, 225:11, 225:21, 226:8, 226:11, 226:14, 226:21, 227:18, 227:22, 228:3, 232:13, 235:17, 235:19, 241:15, 243:6, 243:8, 243:16, 243:19, 244:2, 244:4, 244:5, 244:14, 244:18, 245:1, 245:4
code
55:7, 145:3, 145:4, 146:2, 149:11, 149:12, 166:3, 170:16, 186:15, 186:16, 187:12, 188:10, 189:14, 193:1, 193:2, 193:14, 193:15, 193:16, 194:1, 213:3, 227:1, 227:6, 227:8, 242:6, 242:7, 242:21, 242:22, 243:3, 243:6, 245:9, 245:10
coded
47:12, 48:8, 79:5, 125:19, 125:20, 139:4, 139:13, 149:22, 152:18, 152:21, 153:5, 154:2, 159:4, 164:9, 170:1, 171:6, 182:21, 201:8, 201:9, 207:1, 214:7, 235:3
coders
269:9
codes
145:20, 233:22, 234:4, 234:15
coding
22:11, 22:22, 33:10, 33:22, 34:1, 34:6, 34:16, 34:17, 34:19, 35:1, 35:9, 37:8, 39:3, 57:21, 65:13, 78:12, 79:16, 80:20, 90:13, 94:14, 94:22, 96:5, 96:8, 96:12, 96:14, 96:18, 97:2, 97:21, 99:11, 141:5, 168:11, 187:3, 187:5, 187:9, 189:22, 195:21, 200:22, 218:18, 235:4, 280:19
coie
3:4
coined
131:12
colloquial
82:14, 213:19
colloquially
34:22, 35:19, 227:5
color
141:5
column
13:21, 14:18, 15:10, 84:20, 84:21, 86:5, 90:1, 92:16, 98:9, 98:18, 99:21, 99:22, 100:19, 101:6, 107:10, 108:12, 188:18, 189:3, 256:12, 256:13, 257:12, 261:8, 261:13, 261:18, 261:21, 268:20, 269:16, 271:16, 273:5, 273:7, 273:8, 273:11, 278:12, 280:9, 282:2, 282:22, 283:11, 284:4, 285:15, 286:10, 291:10
columns
256:21
com
1:4, 2:4, 3:7, 3:8, 3:17, 4:7
combination
30:2
come
45:22, 77:8, 278:6
comes
226:10, 226:13, 299:1
commencing
2:20
comment
232:5
communicate
8:4
communication
92:9
communications
36:10
compare
131:13, 305:1
compared
190:11
comparing
87:14
compatible
42:10, 197:19, 251:2
compensation
166:12, 166:16, 174:21, 178:15, 180:8, 181:2, 211:15, 213:10
complaint
157:2
complement
77:20
complementarity
40:4, 40:9, 62:18

complementary
40:11, 42:14, 42:16, 43:3, 57:19, 57:21, 59:2, 61:20, 62:3, 75:8, 77:1, 77:2, 77:9, 77:10, 78:6, 78:9, 78:17, 78:18, 79:13
complete
196:12, 234:16, 282:10
completed
295:5
completely
39:19, 112:6, 131:17, 264:10, 264:14, 264:19, 265:20, 267:11, 271:8
completing
295:22
compliance
27:22, 49:11, 58:14, 158:11, 158:14
compliant
43:10, 44:17, 45:21, 48:4, 53:4, 54:4, 56:11, 57:12, 57:14, 59:9, 59:11, 59:16, 64:3, 65:8, 67:12, 67:14, 70:2, 70:14, 70:19, 70:21, 71:4, 71:17, 71:22, 72:2, 72:17, 74:19, 149:20, 150:17, 150:22, 151:6, 151:12, 151:19, 152:1, 152:2, 157:3, 157:4, 157:6, 157:17, 159:20, 159:22, 253:12, 253:13, 287:8
complied
246:7
complies
43:15, 43:17, 46:6, 46:17, 48:20, 48:22, 51:13, 70:11, 72:15, 76:18, 148:3, 194:15, 254:5
complimentary
39:15
comply
193:9, 196:15
components
53:17
compress
256:10
compressed
14:8, 33:4, 33:6, 33:16, 33:18, 35:5, 35:13, 37:2, 37:4, 37:9, 38:5, 39:1, 39:8, 39:10, 44:3, 55:2, 62:13, 62:14, 62:20, 63:9, 63:22, 64:6, 64:8, 65:1, 67:7, 95:5, 108:16, 108:20, 109:8, 109:20, 110:1, 110:4, 110:7, 110:9, 110:22, 112:15, 112:22, 115:11, 116:1, 121:9, 122:4, 122:7, 122:12, 122:14, 195:4, 256:20, 261:11, 266:21, 277:14
compression
10:10, 32:12, 32:17, 32:19, 32:22, 36:1, 41:12, 60:17, 61:19, 62:9, 69:4, 262:4
compressor
61:21
comprised
95:6, 122:8
comprises
112:9
comprising
14:9
computer
188:20, 234:8
concept
62:7, 125:5, 165:13, 206:21, 207:6, 214:5, 233:6, 233:9, 233:12, 235:2
concepts
168:16, 230:9
concerned
83:13
concerning
20:21, 21:4, 21:10
concerns
27:10, 27:13
conclude
265:13
concluded
265:20, 308:1
conclusion
91:6, 199:20, 249:7, 267:16
conclusions
223:17
condition
211:2
conditions
82:7
confirm
30:11, 228:20, 230:19, 231:9, 231:16, 231:18
confirmed
229:6, 230:5, 245:15
confirming
229:9, 229:11, 229:15
conform
38:13, 38:15, 57:20, 58:6
conformed
45:9, 76:21
conforming
133:12
conforms
38:21, 39:7, 45:18, 58:4, 66:12, 72:16, 79:1, 132:3, 150:2, 254:15
confusing
37:14, 272:4, 291:7
confusion
153:3, 210:9, 212:4
connection
84:17, 90:15
consequence
296:15, 299:20, 300:2, 303:13
consequently
105:10
consider
32:11, 43:16, 63:14, 65:8, 70:13, 71:14, 89:17, 111:15, 121:20, 241:13, 241:17, 245:16, 247:3, 247:20, 248:14, 248:17, 248:19, 279:16, 285:18, 298:5, 298:9, 301:13, 305:11, 305:13, 305:19
considerably
47:4
considered
39:15, 249:1,

249:5, 301:17, 305:2
considering 33:13, 265:3, 279:2, 289:14, 293:16, 298:11, 301:14
consistent 27:8, 34:14, 64:2, 64:11, 85:6, 85:10, 85:20, 86:13, 86:16, 87:22, 108:1, 108:5, 112:7, 114:22, 115:3, 115:6, 122:11, 147:17, 150:7, 163:6, 175:1, 208:19, 215:8, 217:17, 236:15, 248:8, 269:4, 269:7, 276:11, 294:12
consists 169:5, 169:7
constant 190:3, 293:13
constitute 83:2
constraint 254:18, 254:22, 255:1
constraints 254:17
construct 125:21, 126:14, 133:15, 171:20, 190:18
constructed 131:1, 133:10
construction 238:3, 238:6, 238:9, 238:12, 238:21, 239:2, 239:7, 239:10
constructive 206:1
consumer 83:18
contain 12:6, 12:12, 124:15
contained 22:19, 49:12, 158:15
contains 12:16, 124:22, 152:17, 152:21
content 124:15, 124:22
contest 236:14
context 87:7, 87:16, 89:2, 90:9, 90:12, 90:13, 93:20, 99:14, 102:17, 102:18, 102:20, 103:12, 107:19, 108:4, 108:17, 110:22, 112:12, 120:9, 121:20, 133:20, 220:11, 220:12, 228:15, 229:4, 230:1, 230:16, 230:21, 230:22, 231:7, 231:14, 233:4, 271:3, 274:3, 296:4, 296:8, 296:19, 296:20, 297:7, 297:18, 298:6, 299:14, 300:4, 300:13, 300:16, 300:19, 302:21, 304:12, 304:14, 305:13
continue 49:17, 157:21
continues 105:10, 129:22
contorted 296:13, 298:16, 299:18
contradict 52:14, 181:21
controller 278:15, 282:1, 282:4, 284:5, 284:12
controls 282:1, 282:4, 284:5
convert 34:7, 37:2
converting 33:1, 33:6, 33:14, 33:18, 62:12, 62:14, 62:20, 63:8
converts 63:21, 65:1
convey 188:12
copy 6:6, 6:7, 6:8, 6:12, 6:21, 9:22, 10:4, 61:2, 61:4, 61:14
copyright 61:8
correct 10:14, 12:2, 14:2, 42:11, 43:1, 51:10, 106:5, 142:3, 148:14, 148:21, 170:10, 173:1, 175:3, 177:20, 182:22, 218:13, 232:14, 232:15, 233:14, 245:11, 247:16, 269:14, 286:2, 289:11
correction 168:3
corrections 104:15
correctly 104:13, 166:5, 196:17, 222:13, 236:17, 281:22
correspond 134:10, 134:12, 279:3
corresponding 40:15, 131:19, 257:15, 258:5, 261:15, 264:2, 266:17, 266:21, 266:22, 268:6, 268:7, 268:13, 269:1, 269:14, 269:22, 270:8, 270:10, 270:11, 271:17, 278:16, 278:22, 279:6, 279:8, 282:6, 284:7, 284:15, 286:13, 286:21
could 7:20, 33:8, 36:8, 36:11, 36:14, 36:15, 36:16, 36:18, 36:22, 39:9, 41:12, 43:6, 43:10, 43:19, 44:17, 46:19, 52:8, 52:9, 52:10, 55:8, 55:20, 56:1, 56:8, 57:17, 59:13, 66:8, 72:3, 72:5, 73:8, 78:10, 78:19, 78:20, 80:5, 80:14, 92:3, 100:5, 100:11, 100:13, 104:19, 115:2, 115:3, 116:18, 116:22, 124:18, 128:13, 128:18, 128:20, 132:14, 132:17, 135:4, 136:19, 154:6, 160:22, 162:20, 163:17, 171:12, 181:17, 186:15, 189:10, 200:10, 214:7, 243:2,

243:4, 244:17, 244:18, 244:20, 257:17, 258:17, 259:1, 259:12, 259:17, 283:8, 284:13, 287:9, 288:7, 289:15, 292:8, 293:20, 299:10
couldn't 188:15, 293:5, 305:19
counsel 16:21, 18:16, 18:19, 18:20, 18:22, 19:4, 19:9, 19:12, 20:14, 24:6, 24:15, 24:19, 24:22, 57:2, 92:9, 94:19, 114:6, 182:13, 224:16, 239:22, 285:5
counter-example 213:14, 213:17
counter-examples 213:18
couple 16:4, 59:8, 176:8, 180:1, 307:13
course 234:7, 256:4
court 3:15, 5:18
cover 59:7, 73:17, 194:18, 259:8
covered 7:7, 60:15, 68:2, 264:22
covers 26:9
create 45:17, 46:3, 56:8, 86:9, 87:3, 87:5, 88:4, 88:7, 88:12, 88:20, 127:16, 127:17, 127:18, 127:22
created 43:11, 45:6, 45:14, 53:21
creates 53:21, 65:14
creating 45:15
crescent 3:15
current 23:1, 222:4, 235:20, 286:11, 289:1
cut 58:17, 299:9
cv 91:15

D

d-mux 272:6, 273:18
da 214:17, 287:14
dallas 3:16
date 61:8, 236:6, 236:10, 236:18, 238:8, 239:9
day 19:12, 19:13, 184:18, 308:9
days 17:12
dc 186:9, 195:6, 199:5, 200:3
decide 245:5
decided 244:22
decision 17:9, 18:7, 221:17, 223:2, 223:14, 223:16, 223:21, 224:13
decisions 50:9, 51:22, 52:5, 52:9, 52:11, 52:13, 52:16, 52:22, 147:20
declarations 10:13, 10:19, 12:6, 12:12, 15:8, 29:18, 120:21, 220:3, 220:22, 301:17
decodable 46:7, 46:8, 58:7, 58:13, 101:14, 102:11, 102:15, 103:3, 105:5, 105:14, 105:20, 106:18, 139:6, 157:17
decode 38:5, 39:1, 39:10, 46:13, 51:2, 51:4, 52:19, 55:13, 56:12, 60:6, 67:10, 72:21, 72:22, 74:1, 74:22, 76:20, 112:3, 123:1, 126:12, 126:15, 126:17, 128:15, 128:17, 128:21, 129:13, 129:14, 129:17, 133:3, 138:8, 152:6, 152:11, 153:5, 153:6, 154:11, 159:15, 186:19, 187:2, 187:3, 190:2, 193:21, 197:21, 198:10, 199:17, 245:8, 252:3, 254:6, 266:8, 268:1, 268:2, 268:4, 268:8, 279:15, 279:19, 280:19, 294:4, 295:4
decoders 38:9, 38:12, 38:14, 38:19, 39:17, 42:10, 49:12, 64:3, 64:14, 74:20, 145:1, 145:7, 158:12, 158:14, 160:7, 160:19, 160:20, 162:16, 228:17, 229:21, 269:9, 269:10
decodes 39:9, 50:8, 50:17, 51:13, 56:7, 65:10, 65:15, 65:17, 66:3, 66:17, 67:15, 68:6, 72:16, 73:1, 74:13, 95:1, 111:22, 152:2, 156:6, 157:3, 157:5, 159:5, 159:10, 159:19, 159:21, 186:21, 225:21, 226:8, 246:20, 253:8, 279:20
decompressed 39:11
decompressor 61:21
define 34:13, 156:4, 165:11, 191:20
defined 47:7, 58:9, 59:3, 67:8, 79:9, 154:2, 158:1, 158:5, 158:20, 168:17, 168:19, 168:22, 205:16, 207:13, 207:15, 246:14,

303:5, 305:10
defines
69:14, 153:15, 154:4
defining
81:16, 105:3, 106:16
definition
25:7, 25:17, 25:20, 25:21, 81:9, 81:11, 81:13, 81:19, 82:19, 83:5, 84:19, 86:12, 88:10, 88:15, 88:18, 88:22, 153:16, 203:18, 206:1, 218:20
demultiplexed
272:22
denoted
14:12
denotes
166:7, 213:4
depend
177:1, 177:13, 179:15, 181:14, 286:7, 286:18, 287:3, 287:17, 288:5, 288:6
dependencies
130:3, 136:20, 137:20, 137:22
dependency
127:3, 127:4, 135:12, 136:16, 172:21
dependent
141:16, 141:22, 142:2
depending
31:7, 34:13, 128:4, 135:13, 176:22, 302:21
depends
31:8, 127:6, 129:8, 130:15, 131:16, 139:3, 144:15, 173:11, 175:14, 180:19, 186:21, 187:1, 187:14, 263:18, 264:11, 267:12, 271:7, 287:10, 287:11
depicting
261:10
deposition
1:14, 2:18, 5:8, 7:4, 7:19, 9:22, 16:16, 17:19, 19:10, 19:15, 21:6, 21:7, 246:10
depositions
19:16
describe
78:11, 144:4, 144:16, 173:8, 176:4, 177:19, 178:2, 196:7, 261:16, 274:10, 285:12
described
14:12, 33:2, 33:8, 34:1, 87:15, 90:16, 92:4, 103:9, 106:1, 108:2, 139:17, 142:19, 143:2, 150:7, 168:17, 170:13, 193:18, 268:20, 274:10, 282:19
describes
84:16, 87:20, 89:7, 89:8, 89:12, 100:9, 104:8, 165:21, 174:14, 183:9, 196:9, 203:3, 255:15, 255:17, 255:19, 255:21, 256:15, 256:16, 257:4, 261:21, 274:1, 275:4, 275:8, 278:11, 278:12, 286:9
describing
78:17, 175:21, 175:22, 193:9, 197:18, 208:12, 214:10, 246:10, 256:18, 261:8, 291:5
description
99:9, 103:6, 125:16, 141:14, 145:4, 162:12, 162:14, 162:19, 164:6, 196:12, 208:21, 209:11, 267:2
descriptions
92:13
design
51:17, 128:19
designation
27:5
designed
145:13
designer
46:5, 51:18, 52:10, 138:12
detail
198:3, 223:8, 263:21
detect
278:14
detected
264:9
detecting
258:7, 279:5, 279:8
detects
278:15
determination
200:1
determine
217:1, 283:19
determined
172:4, 250:5, 250:9, 250:15
determines
252:9, 253:4
development
196:15
deviate
48:17
device
58:19, 82:8, 82:16, 145:22
diagram
126:10, 126:11, 135:15, 135:20, 135:21, 136:10, 139:15, 200:19, 217:2, 276:11, 285:21
diagrams
141:20
diego
3:6
difference
17:21, 137:2, 168:13, 211:8, 212:1, 217:5, 260:2, 296:16, 297:5
different
14:14, 17:13, 66:7, 68:15, 89:9, 91:9, 91:18, 96:6, 100:14, 118:17, 123:10, 127:10, 135:15, 141:4, 141:5, 141:6, 141:7, 141:11, 162:16, 169:2, 169:20, 175:2, 177:17, 177:18, 180:15, 181:6, 181:12, 194:8, 200:14, 208:16, 248:6, 262:18
differently
48:3, 300:22
difficult
103:14
digital
55:3, 55:4, 61:2, 274:2

directed
54:22, 96:11, 96:19, 97:13, 97:21, 99:10, 162:13, 191:17
directly
180:10, 249:12
disagree
13:10, 170:6, 170:7, 176:1, 178:8, 209:18, 210:4, 222:15, 222:16, 222:18, 222:22, 223:16, 223:20, 224:12, 225:10, 233:9, 237:7, 252:16, 252:22, 253:7, 290:14, 290:19, 294:16
disc
55:9, 85:12, 145:17, 261:14, 261:16, 267:9, 268:10, 282:7, 282:11, 282:14, 282:21, 283:3, 283:6, 283:16, 283:19, 284:8
discard
193:13, 193:21, 242:5
discarded
193:5
disclose
92:10, 190:3, 194:18, 221:13, 260:16, 265:15, 281:1, 282:18, 292:2, 302:4, 302:5, 302:7, 303:3
disclosed
170:9, 204:2, 205:6, 205:21, 257:6, 280:22, 292:5, 292:11, 292:15, 294:16
discloses
191:21, 206:2, 219:19, 219:20, 222:12, 223:6, 249:17, 255:9, 256:22, 258:8, 266:12, 266:13, 267:10, 278:5, 279:21, 285:10, 285:13
disclosure
86:20, 86:22, 249:15, 256:10, 257:21, 258:15, 260:3, 260:9, 266:14, 282:8, 282:17, 284:10, 293:17, 294:8, 302:15
disclosures
305:14
discontinuity
103:20, 104:2, 104:8, 189:4
discuss
57:1, 69:10, 78:8, 92:3, 92:20, 103:19, 104:7, 114:5, 146:16, 170:3, 195:1, 232:16, 239:21, 240:13, 249:10, 260:17, 285:3
discussed
41:3, 43:21, 52:15, 60:11, 64:15, 123:14, 150:8, 153:11, 160:1, 160:3, 164:16, 180:2, 190:8, 208:11, 208:17, 217:19, 225:7, 233:17, 244:15, 277:16, 290:22, 294:18
discusses
25:6, 93:4, 106:16, 165:13, 174:4, 202:9, 291:10, 291:11
discussing
68:21, 83:6, 100:13, 108:17, 137:3, 137:4, 147:5, 271:21, 302:3
discussion
36:8, 96:2, 137:2, 146:8, 166:1, 184:20, 203:4, 208:15, 232:20, 236:1, 237:3, 237:7, 241:22, 274:21
display
35:7, 52:12, 52:17, 52:19, 128:22, 131:10, 132:5, 140:16, 154:13, 154:17, 155:3, 155:7, 155:11, 155:21, 156:10, 164:10, 164:13, 164:19, 164:21, 165:2, 165:4, 165:7, 169:6, 173:18, 205:13, 206:4, 206:9, 214:22, 215:7, 215:17, 216:11, 217:10, 217:13, 217:21, 218:7, 218:12, 218:20, 219:6, 219:11, 247:10, 249:18, 251:5, 251:11, 251:15, 251:18, 252:6, 252:10, 253:5, 253:15, 289:1
displayable
33:9, 33:15, 33:19, 53:17, 65:16, 138:16
displayed
87:5, 128:7, 129:7, 129:11, 131:7, 132:1, 132:10, 140:13, 156:5, 164:8, 165:12, 207:2, 209:3, 214:7, 268:16
displaying
82:16, 129:1, 186:3
displays
87:18
dispute
219:19, 219:22, 220:4, 220:17
disputing
216:16
disruption
138:22
distortions
128:4
disturbances
139:6
divided
283:9
document
6:9, 12:19, 17:8, 27:12, 27:18, 28:15, 29:2, 69:6, 71:6, 73:19, 96:21, 158:15, 158:17, 160:4
documents
5:22, 6:3, 6:5, 6:17, 7:6, 7:10, 12:18, 16:17, 16:19, 17:20, 18:2, 18:4, 18:10, 21:18, 22:5, 22:14, 22:15, 22:17, 23:8, 26:18, 89:3, 232:19
doing
38:18, 87:13, 163:1, 163:3, 168:20

domain
93:17, 291:19
done
29:17, 68:14, 100:6, 101:3, 133:21, 196:13, 202:3, 220:21, 224:18, 307:20
doublecheck
27:1
doubt
279:2
down
16:5, 25:3, 103:18, 216:1, 240:3
dr
1:15, 2:18, 5:1, 5:6, 6:8, 7:4, 56:22, 83:16, 83:20, 85:9, 86:11, 89:9, 114:4, 182:9, 202:4, 239:20, 245:15, 246:9, 285:2, 308:5, 309:3
draft
289:10
drafted
270:18
drafting
271:1, 272:4
draw
49:7, 162:9
driving
124:6, 141:10
dsp
272:12, 273:1, 273:14, 274:1
duly
5:2, 310:8
during
8:3, 45:7, 57:1, 90:22, 114:5, 140:18, 182:11, 208:3, 239:21, 243:15, 262:4, 285:3
dvd
55:9

E

e
302:17
each
5:16, 57:22, 67:9, 68:14, 77:20, 78:6, 78:7, 111:4, 111:19, 120:1, 167:13, 175:1, 175:12, 176:18, 177:9, 179:2, 217:21, 220:2, 224:5, 227:10, 249:18, 285:17, 286:1, 286:4, 286:9, 286:10, 286:22
earlier
18:1, 32:15, 49:9, 57:19, 59:1, 59:7, 60:11, 64:15, 90:18, 95:8, 109:11, 112:20, 113:10, 158:22, 167:3, 174:4, 174:19, 184:18, 190:8, 192:7, 207:12, 208:11, 208:17, 211:22, 217:20, 233:19, 234:5, 237:13, 255:5
earliest
217:20, 249:17
early
291:20
easier
68:17
editing
208:3
education
21:21, 22:9, 23:14
edward
5:13
effect
158:18, 181:9, 216:22
effectively
256:18
either
71:9, 79:7, 95:10, 120:20, 142:16, 142:20, 143:6, 143:15, 144:5, 158:7, 175:13, 176:20, 177:10, 178:10, 178:20, 179:13, 217:4, 258:8, 259:13, 272:11, 307:18
el
3:5
electronic
9:19, 10:4, 61:13
elements
44:6, 44:8, 47:21, 53:7, 53:11, 53:16, 54:12, 65:10, 65:18, 66:3, 66:18, 67:9, 68:1, 68:10, 70:4, 71:8, 95:6, 112:1, 112:5, 112:9, 115:5, 119:14, 119:16, 119:17, 119:18, 122:9, 165:19, 226:20
else
23:7, 36:18, 70:15, 110:11, 113:3, 120:5, 132:12, 146:22, 198:21, 199:2, 209:7, 231:2
elsewhere
260:6
email
3:7, 3:17, 4:7
embodiment
103:13
embodiments
92:3, 92:14, 92:15, 92:18, 92:21, 93:5
employed
38:4, 38:22
employing
99:6, 197:10
enable
184:22, 185:9, 188:4, 191:12, 191:14, 191:19, 192:8
enables
79:20, 80:21, 101:9, 191:7, 191:9
encode
45:17, 149:20, 168:20
encoded
34:7, 43:8, 44:18, 50:10, 72:6, 150:6, 151:18, 172:15, 172:22, 173:4, 174:16, 176:14, 178:18, 179:9, 180:3, 181:4, 201:7, 207:21, 208:8, 209:1, 209:15, 210:1, 210:6, 210:13, 210:21, 212:10, 212:14, 212:19, 228:16, 229:20, 251:2, 256:20
encoder
35:20, 36:2, 36:3, 36:4, 36:6, 36:8, 37:7, 37:11, 37:12, 37:15, 37:17, 38:5,

39:1, 40:10, 40:21, 41:5, 41:14, 41:17, 42:14, 42:15, 43:4, 45:6, 45:7, 45:11, 45:15, 45:20, 46:2, 50:10, 51:22, 53:3, 53:20, 53:22, 54:9, 54:15, 55:19, 56:4, 57:8, 57:14, 57:19, 58:3, 58:12, 58:18, 59:11, 59:16, 59:22, 61:21, 62:12, 70:2, 70:6, 70:7, 70:11, 70:13, 70:18, 70:20, 71:17, 71:20, 71:21, 72:1, 72:5, 72:8, 73:3, 76:16, 76:17, 77:16, 78:1, 78:2, 78:19, 79:3, 79:13, 102:13, 103:1, 147:20, 148:2, 149:19, 150:1, 150:3, 150:18, 150:21, 151:6, 156:18, 156:21, 157:1, 158:4, 168:22, 229:15, 250:5, 250:9, 250:10, 250:16, 250:18, 251:1, 251:4, 251:11, 251:17, 251:21, 306:5

encoders
37:13, 37:21, 38:1, 39:13, 39:16, 39:22, 40:5, 40:16, 64:13, 157:16

encodes
72:8, 150:3, 255:15, 255:22

encoding
32:13, 32:17, 32:20, 34:6, 35:18, 35:22, 37:6, 39:18, 40:6, 43:16, 43:20, 45:8, 47:2, 47:5, 52:1, 58:3, 58:19, 77:16, 82:12, 147:12, 147:21, 149:19, 158:1, 158:5, 167:6, 167:9, 167:16, 168:8, 168:13, 168:18, 168:20, 192:19, 208:4, 213:13, 256:13, 256:14

encounters
194:1, 227:20

end
51:3, 91:16, 98:22, 196:17

ending
201:4

enough
93:22, 195:17

entire
73:20, 225:13, 299:11

entirely
41:15, 77:3, 215:12

entirety
13:4, 13:7, 196:5, 196:8, 237:17, 246:12, 249:5, 249:13

entitled
99:15, 103:6

entropy
77:16, 77:18, 78:1

equals
306:13

equivalent
279:17

error
129:11, 129:21, 131:10, 131:11, 131:18, 132:13, 132:14, 137:13, 138:10, 138:20, 139:8, 139:11, 140:2, 140:17, 167:21

errors
125:10, 126:1, 126:7, 128:14, 132:18, 133:4, 138:6, 138:18, 139:7

esq
3:10, 3:11, 3:18, 4:8

essentially
38:4, 48:11, 111:18, 130:18

establishes
222:5

estimate
19:20

estimation
40:1, 47:6

evaluated
243:17

even
46:19, 68:11, 76:10, 91:7, 93:14, 146:1, 175:1, 186:13, 189:21, 194:18, 224:2, 248:5, 278:22, 288:18, 295:5

ever
238:2, 293:4

every
7:21, 24:14, 42:4, 50:3, 71:4, 77:11, 113:17, 150:19, 232:5

everything
22:21, 23:13, 24:13, 193:21, 193:22, 245:17, 245:22, 299:7

evidence
238:7, 239:8

exact
44:20, 77:11, 78:7, 93:15, 94:9, 94:17, 122:16, 147:16

exactly
17:11, 42:1, 50:5, 63:18, 68:14, 74:17, 87:19, 148:9, 158:16, 208:19, 272:15

examination
5:4, 307:22, 309:4, 310:7

examined
5:3

examples
38:1, 43:2, 55:4, 81:8, 86:5, 86:12, 87:11, 180:12, 185:10, 185:18, 190:17, 215:8, 217:6, 218:1, 219:15, 248:6

excerpt
234:20

excerpted
234:18, 266:15

exclude
23:18, 136:18, 139:5

excuse
8:18, 16:11, 46:11, 78:20, 174:12, 190:22, 209:13, 260:18, 277:6

executing
115:1

exemplary
202:10, 202:14, 203:2, 203:3, 203:10, 203:14, 203:18, 204:5, 204:9, 205:2, 205:10, 205:12, 205:18
exhaustive
17:4
exhibit
6:6, 6:7, 6:9, 6:12, 6:17, 6:21, 7:3, 11:6, 11:15, 11:18, 13:16, 15:6, 17:8, 18:8, 21:13, 26:17, 26:21, 27:4, 29:12, 29:13, 29:14, 30:10, 30:14, 47:1, 49:10, 53:10, 53:14, 55:1, 84:13, 91:16, 149:7, 150:9, 153:16, 158:13, 196:7, 234:13, 234:14, 236:5, 236:6, 236:9, 246:1, 255:5
exhibits
7:16, 9:13, 16:18, 17:2, 17:11, 17:16, 17:20
exists
181:3
expand
130:1
experience
21:21, 22:9, 23:2, 23:15, 23:20, 32:16, 34:18, 83:21, 83:22, 94:16, 237:22, 239:11, 251:21, 304:3, 304:15
expert
14:1, 24:1, 24:3, 32:12, 90:19, 90:22, 238:1, 239:11
expertise
93:17, 94:7, 94:13, 94:21
explain
33:21, 44:21, 77:19, 258:3
explained
94:19, 97:3, 198:2, 243:21
explains
105:2, 165:8
explanation
125:17
explicit
216:21
explicitly
217:5, 217:7, 284:9
express
230:13
expressed
120:22, 221:3, 221:11, 223:8
expressing
120:19
extent
22:8, 22:21, 59:14, 216:18, 237:5
extract
201:18, 201:19
extremely
62:11
extrinsic
238:7, 239:8

F

facial
128:3
fact
31:21, 46:18, 49:7, 53:3, 62:12, 64:12, 77:5, 139:5, 151:18, 187:20, 265:21, 295:17
factor
240:13, 240:14, 240:20, 241:12, 241:22, 243:11, 243:13, 244:11
factors
35:12, 240:5, 240:10, 240:16
fail
128:15
fails
31:13, 31:15
fair
24:7, 32:14, 300:6, 300:13
familiar
38:1, 39:17, 39:18, 39:21, 40:5, 40:21, 64:4, 64:14, 73:11, 73:18, 82:1, 112:7, 148:13, 148:15, 160:7, 160:8, 162:19
far
7:8, 20:11, 205:17, 224:9, 260:3, 285:20
fast
80:11, 81:6, 191:3
february
1:18, 2:21
fed
272:12
feel
236:13
few
18:21, 45:22, 81:22, 90:6, 90:20, 202:14, 203:11, 300:12
fewer
47:6
field
22:11, 23:16, 33:11, 33:22, 34:15, 35:1, 35:15, 36:9, 37:1, 40:19, 84:16, 90:16, 258:14, 258:16, 259:11, 259:16, 260:4, 260:10, 260:14, 279:4, 279:5
fields
94:16, 260:17
figure
77:15, 77:18, 145:5, 154:20, 155:1, 185:19, 186:9, 187:6, 199:5, 214:17, 215:15, 216:6, 216:10, 218:8, 219:2, 219:5, 219:14, 225:20, 226:2, 247:5, 247:11, 247:12, 247:21, 248:5, 248:14, 248:15, 248:20, 249:11, 249:14, 256:22, 258:3, 259:8, 260:12, 260:15, 260:16, 261:6, 261:8, 262:7, 266:15, 268:20, 272:5, 274:13, 274:22, 275:18, 276:1, 278:18, 278:19, 279:2, 279:9, 282:16, 282:18, 283:1, 284:14, 285:12, 285:13, 285:16, 285:17, 285:18, 285:22, 286:5, 286:16, 287:14, 288:3, 289:14, 290:8, 290:20,

291:5, 293:2, 293:3, 293:15, 301:20, 307:3
figures 77:14
file 1:23, 80:7, 97:8, 105:6, 106:19
filed 6:18, 6:22, 7:16, 11:9, 17:17
files 86:6, 87:3, 88:3, 90:14
filing 11:12
filling 92:7
final 77:15, 120:11
finally 7:3, 16:10
find 146:2, 146:3, 188:15, 192:17, 227:5, 269:7, 272:3, 295:16, 301:1
finding 104:9
finds 193:14, 242:5
fine 8:7, 11:17, 15:3, 114:1, 275:7, 275:8
fingers 100:3, 154:6
finish 8:1, 179:1
finished 300:7
firm 18:22
fist 75:5
five 135:1, 136:19, 240:5, 240:10, 240:14, 244:11, 290:10
five-picture 219:8
flag 165:21, 166:3, 166:5, 166:6, 174:5, 174:7, 174:14, 175:5, 175:11, 176:4, 176:13, 176:19, 177:8, 179:8, 179:17, 179:18, 179:21, 180:17, 181:3, 181:12, 182:17, 182:19, 207:19, 209:19, 209:21, 210:19, 211:3, 211:6, 212:16, 212:22, 213:2, 213:3, 213:4, 225:1, 225:12, 225:15, 225:21, 226:8, 226:11, 226:14, 227:18, 227:22, 228:3, 232:13, 235:17, 235:19, 243:16, 244:2, 244:14, 244:18
flipping 24:13
flow 260:13, 274:14, 274:17, 274:22, 275:2
flowchart 261:9, 261:19, 270:6, 271:19, 274:14
flows 274:14
fly 89:1, 95:21, 116:21
focused 60:16, 60:20
focusing 35:16
follow 67:13, 150:6, 157:6, 200:14, 244:17, 244:18, 244:20
followed 50:5, 98:3, 167:5, 167:17, 169:8, 200:12, 200:21, 201:1, 204:16, 290:9, 290:18
following 84:20, 111:20, 175:6, 185:10, 185:11, 216:12, 244:13
follows 5:3, 42:4, 42:5, 48:9, 130:19, 162:12, 162:14, 166:3, 213:3, 228:3, 302:2
foregoing 310:9
forget 44:20, 87:19
forgot 59:7
forgotten 47:15, 58:15, 122:16
formats 37:3, 82:12
formed 116:20, 223:4, 223:5
former 239:5
forth 61:8, 95:8, 285:19, 310:8
forward 40:16, 41:16, 80:6, 81:6, 191:3
forwarding 80:11
forwards 145:16, 145:18, 145:22
four 19:1, 100:19, 136:21, 160:4, 290:10, 300:4
frames 39:11, 41:5, 41:6, 41:7, 41:9, 52:13, 52:19, 64:10, 64:19, 65:15, 65:16, 78:20, 79:7, 83:1, 83:15, 84:4, 84:10, 85:17, 86:7, 86:19, 88:11, 88:19, 89:15, 104:3, 109:13, 111:3, 111:7, 111:9, 111:16, 111:17, 111:18, 112:2, 112:10, 116:3, 116:6, 117:5, 117:10, 117:13, 117:15, 118:3, 118:12, 118:21, 119:1, 119:10, 119:12, 121:11, 121:13, 124:6, 124:7, 124:11, 125:20, 130:5, 133:22, 134:2, 134:6, 134:10, 138:18, 155:18, 155:22, 175:15, 177:2, 178:9, 179:12, 179:15, 179:19, 181:14, 183:20, 183:21, 184:4, 188:8, 188:14, 189:7,

189:17, 220:8, 255:10, 255:11, 255:12, 255:16, 255:20, 255:22, 256:2, 256:6, 273:2, 273:15, 274:5, 296:22, 297:1
framing 189:22, 304:5
free 131:10, 131:11, 131:18, 236:13
front 5:22, 6:2, 6:14, 7:10, 7:11, 27:3, 61:7, 67:4, 68:18, 73:7, 73:10, 79:11, 113:19, 201:15, 234:7, 236:16
fulfill 230:11
full 8:19, 19:12, 19:13, 40:15, 98:22, 234:3, 234:15
full-screen 269:1, 270:12, 270:13
fully 8:17, 129:7, 129:10, 184:22, 185:9, 188:4, 191:7, 191:9, 191:11, 191:14, 191:18, 192:7, 271:8
fully-enable 192:3
function 226:22
functional 196:17, 260:5, 260:11, 271:2
further 31:16, 42:10, 310:13

G

ga 4:6
gave 42:15, 81:8, 85:8, 235:10
general 22:12, 36:9, 37:1, 37:12, 43:14, 49:18, 65:7, 91:17, 107:22, 119:15, 145:7, 220:10, 222:16
generally 11:2, 24:8, 28:10, 37:16, 40:19, 89:18, 100:11, 103:10, 103:11, 125:8
generate 67:11, 67:15
generates 273:1
generating 181:17, 256:10, 273:15, 274:4
genuinely 271:10
give 14:3, 80:4, 80:14, 81:4, 88:21, 95:20, 95:21, 99:21, 100:19, 102:20, 103:15, 120:18, 121:4, 130:13, 153:20, 185:10, 192:9, 197:12, 200:18, 235:8, 246:19
given 86:19, 87:11, 88:15, 95:11, 95:20, 198:11, 201:21, 291:17, 310:10
gives 87:6, 87:16, 245:7, 247:10, 248:5
giving 81:13, 86:5, 163:14, 279:1
go 13:19, 59:6, 103:13, 104:6, 140:19, 195:21, 196:1, 202:18, 204:12, 205:17, 223:13, 227:5, 240:13, 260:8, 266:13, 278:4, 278:10, 282:13, 289:2, 306:21
goes 22:12, 35:14, 146:14, 208:2, 272:11, 272:12, 274:18
going 28:14, 28:22, 29:19, 56:16, 58:11, 64:17, 64:18, 76:4, 78:2, 113:13, 138:21, 149:5, 153:19, 162:7, 186:20, 202:17, 240:3, 261:19, 265:14, 284:17, 298:10
gone 220:22, 221:1
good 5:6, 5:7, 5:17, 113:21, 188:22, 270:16
gop 99:5, 105:5, 153:7, 154:13, 154:18, 156:15, 156:19, 157:10, 165:14, 166:5, 169:12, 169:17, 175:12, 175:15, 176:19, 177:2, 177:9, 177:14, 178:13, 178:16, 179:3, 179:12, 179:15, 180:13, 180:18, 180:20, 180:22, 181:13, 181:14, 185:18, 185:21, 186:1, 187:16, 205:8, 207:19, 209:8, 218:6, 218:16, 218:18, 218:20, 219:8, 219:20, 220:5, 220:18, 222:12, 225:13, 242:10, 242:14, 242:17, 242:21, 243:3, 243:5, 243:10, 243:17, 244:2, 245:10, 246:21
gops 220:1
graphic 124:10
great 10:12, 12:5, 15:22, 16:10, 16:15, 27:9, 28:2, 29:3, 29:19, 57:4, 98:20, 154:5, 182:15, 214:4, 307:19
ground 7:18, 26:14, 28:6, 28:19, 29:4, 29:10, 30:1, 31:11, 31:14, 31:18, 32:5, 32:8
grounds 12:16, 26:9, 147:6, 147:7
group_start_code 166:20

grouping
90:13
groups
166:8, 167:13, 174:19, 190:14, 190:18, 191:5, 201:20, 211:13, 211:16, 213:8, 213:14, 213:21, 216:9, 216:11, 220:1, 248:6
guess
26:15, 36:8, 38:11, 95:14, 101:21, 114:18, 169:1, 169:17, 181:11
guest
284:12
guide
213:12
guy
239:3

H

h
60:13
half
19:6, 215:22
hand
68:3
handle
132:13, 189:19
handles
128:5
happen
59:22, 73:8, 73:9, 74:3, 127:14, 128:6, 128:13, 129:5, 243:21, 259:3
happened
19:17, 19:19
happens
66:1, 264:7, 271:19
happy
16:3, 27:15, 29:16, 30:19, 49:17, 68:19, 275:3
hard
9:22, 23:2, 43:13, 43:18, 191:19
hardware
39:22, 51:17, 145:4, 160:20
havlicek
6:8, 7:4, 83:16, 83:20, 89:9, 245:15
havlicek's
85:9, 86:11, 86:12, 246:9
head
8:14, 306:19
header
53:10, 55:16, 56:6, 149:3, 153:8, 167:5, 175:7, 225:22, 242:14, 242:17, 243:10, 243:17, 244:2, 244:6, 245:2, 245:6
hear
5:15
help
76:11
helpful
8:6, 8:13, 8:21, 15:18, 109:5
here
9:7, 14:13, 24:14, 26:1, 26:15, 34:5, 73:8, 76:5, 76:13, 77:16, 78:16, 81:4, 81:17, 81:18, 88:17, 95:13, 106:2, 107:6, 108:2, 112:12, 117:3, 117:13, 125:18, 128:11, 129:1, 132:17, 146:5, 148:1, 153:12, 153:14, 162:5, 163:7, 188:2, 197:1, 204:19, 210:9, 215:21, 223:18, 224:10, 225:7, 240:4, 262:22, 263:6, 265:2, 265:5, 287:16, 297:16, 305:4
hereby
310:5
hereinbefore
310:8
hesitate
89:1
high
32:22, 33:10, 62:8, 62:11, 76:14, 77:20, 85:7, 127:5, 135:4, 256:9
high-level
125:16
higher
86:8, 189:15
hmm
256:5
hold
23:22
honest
33:11, 57:16, 59:12, 113:13, 167:20, 223:13, 272:2
honestly
17:10
hopefully
64:19, 65:15
horizontal
53:15, 54:2, 55:16, 55:20, 55:21, 56:4, 56:5
horizontal_size
53:11
hour
7:21, 7:22, 56:17, 284:18
hours
19:7
however
133:10
huh
41:21
human
35:12
hundred
196:3
hundreds
145:3
hypothesize
145:21
hypothesizing
146:5
hypothetical
54:20, 60:2, 65:6, 65:21, 74:5, 123:13, 128:9, 132:20, 175:17, 195:9, 197:17, 212:13, 259:19, 260:22, 287:6

I

i-frame
136:9, 136:14, 136:22, 175:13, 192:17, 200:11, 203:15, 204:3, 205:3, 205:12, 227:18, 258:6, 264:9, 276:4, 277:13, 277:14, 279:9, 284:1, 286:7, 286:18, 286:22, 290:2, 290:4, 290:17, 293:21
i-frames
193:4, 255:11, 255:17, 294:5
i-picture
161:14, 161:20,

162:6, 167:5, 167:16, 169:7, 170:14, 179:4, 186:1, 186:16, 189:15, 193:2, 193:16, 200:21, 201:3, 201:5, 204:16, 204:21, 206:9, 206:16, 214:21, 215:6, 215:16, 217:9, 218:2, 218:11, 218:16, 218:17, 218:18, 219:6, 219:10, 227:14, 228:6, 241:14, 242:8, 242:9, 242:16, 242:21, 243:8, 243:10, 255:20, 256:11, 256:19, 257:14, 257:16, 258:12, 258:13, 258:14, 259:9, 260:6, 260:7, 260:8, 260:10, 263:18, 264:2, 264:11, 266:1, 266:9, 266:11, 267:11, 268:13, 269:2, 270:9, 270:10, 271:7, 271:9, 276:5, 276:18, 277:13, 277:18, 277:20, 278:1, 278:2, 278:16, 278:22, 281:1, 281:5, 281:6, 281:7, 281:8, 281:10, 281:19, 283:2, 286:12, 287:4, 287:10, 287:11, 287:21, 290:9, 291:14, 292:5, 292:12, 292:15, 292:19, 293:3, 294:9

i-pictures
167:7, 167:18, 169:21, 171:14, 185:20, 255:18, 293:21

i0
288:13, 289:16

i8
186:19, 186:20, 186:21, 186:22, 187:3, 187:8, 187:12, 187:14, 188:9, 192:18, 193:5, 194:2, 194:7, 194:12, 195:7

iad
257:2, 257:6, 257:17, 257:19, 257:22, 258:2, 258:14, 258:17, 259:11, 259:16, 260:4, 260:10, 260:17, 277:6, 277:9, 277:11, 277:13, 278:1, 278:7, 279:4, 280:1, 280:8, 280:12, 280:14, 285:22, 286:5, 286:12, 286:14, 286:17, 286:19, 293:4, 293:8, 293:14, 293:22, 306:13

iain
1:15, 2:18, 5:13, 308:5, 309:3

idea
55:6, 60:3, 60:9, 300:3, 305:20

identical
41:8, 64:17, 64:22, 158:5, 158:19

identified
22:6, 99:17, 115:13, 171:17, 295:1

identifier
16:7, 16:12, 105:12, 105:18, 106:4, 106:10, 107:2, 107:7, 107:11, 107:14, 107:15, 107:20, 107:21, 107:22, 108:3, 108:7, 108:13, 108:19, 109:1, 109:7, 109:13, 109:16, 110:2, 110:8, 110:14, 110:18, 111:8, 111:13, 112:14, 112:21, 113:6, 114:11, 114:15, 115:10, 115:15, 115:18, 115:20, 116:3, 116:5, 116:6, 116:14, 116:16, 117:4, 117:6, 117:9, 117:12, 117:14, 117:16, 118:2, 118:12, 118:20, 119:1, 119:9, 119:11, 119:12, 119:22, 120:3, 120:11, 120:12, 121:10, 121:13, 121:14, 122:3, 122:10, 122:13, 122:21, 122:22, 123:5, 123:8, 123:16, 123:17, 123:18, 194:6, 306:5, 306:7, 306:11, 306:12, 306:13, 306:15, 307:4, 307:6, 307:8, 307:9

identifiers
109:12

identifies
97:6, 205:2, 293:12

identify
79:21, 80:21, 100:9, 100:15, 183:2, 184:3, 225:14, 293:5, 293:9, 293:10, 294:1, 306:14

identifying
142:7, 157:22, 184:6, 184:7

iec
6:9, 26:22, 27:6, 29:2, 47:8, 47:11, 49:12, 158:15

ignore
42:3, 159:16, 220:15

illustrate
125:4, 129:15, 130:17, 130:20, 135:3, 136:11, 139:21

illustrated
139:14, 142:11, 219:8, 268:19

illustrating
127:4, 128:11, 129:2, 135:22, 136:15, 139:16, 261:9

illustration
131:5, 135:4, 135:5, 135:17, 135:19, 136:17, 141:22, 142:3, 142:6, 144:8, 145:6

illustrative
139:2, 139:19

images
33:13, 35:13, 87:19, 117:5, 134:6, 134:22, 140:20, 141:1, 141:9, 141:15, 222:11

imagine
32:19, 57:16, 59:12, 74:10, 146:5, 159:18, 160:18, 160:21, 161:5, 195:11, 197:1, 280:14, 287:13
imagining
133:11
immediately
149:10, 184:13
impairing
9:10
implement
138:4
implementation
41:14, 52:22, 127:15, 127:16
implementations
38:12, 38:14, 38:16, 160:8
implemented
144:22
implementing
115:8, 227:4
implies
158:8
implying
77:11, 237:5
import
91:3
important
104:20, 175:20
impressed
73:21
impression
86:9, 87:4, 87:5, 88:4, 88:7, 88:12, 88:20
improper
91:2
improve
96:8, 96:13, 96:17
improvement
80:20, 96:4
improvements
97:1
improves
190:9
improving
79:16, 96:11, 96:20, 97:22, 99:10, 191:17, 191:21
in-person
16:21, 18:16, 19:9
inc
1:4, 2:4
include
34:19, 34:21, 35:3, 35:4, 35:6, 58:8, 58:10, 93:8, 111:8, 116:6, 116:15, 119:12, 121:14, 150:11, 151:3, 151:9, 152:8, 180:18, 186:16, 202:1, 278:1, 281:9
included
72:4, 110:8, 110:14, 110:16, 113:7, 118:15, 149:3, 151:21, 236:8
includes
46:7, 67:8, 69:17, 69:20, 82:10, 82:11, 82:15, 117:5, 117:16, 150:9, 151:1, 151:7, 152:3, 152:16, 153:9, 158:17, 158:20, 162:12, 166:20, 171:20, 201:18
including
12:18, 21:19, 118:9, 167:15, 185:18, 193:2
incomplete
54:20, 60:2, 65:6, 65:21, 74:5, 123:13, 128:9, 132:20, 175:17, 195:9, 197:17, 259:19, 260:22, 287:6
inconsistent
148:10, 148:11, 231:13
incorporates
11:22
incorrect
140:9, 168:1, 225:18, 240:11, 240:17, 253:1
independent
50:9, 114:12, 115:19, 115:22, 142:15, 144:13, 183:8, 184:15, 198:1, 209:4, 220:5, 220:7, 220:9, 222:7, 222:10, 222:11, 225:5, 241:1, 265:13, 265:17, 294:22, 303:7, 303:13, 303:20, 304:6, 304:18
independently
99:8, 101:13, 101:19, 102:11, 102:14, 103:3, 105:4, 105:14, 105:19, 106:17, 207:2, 214:7
indeterminate
187:17
indicate
154:17, 155:6, 155:9, 155:10, 175:12, 183:20, 225:15, 241:1, 286:20, 303:6
indicated
114:11, 115:18, 115:21, 116:7, 117:6, 117:17, 119:13, 120:1, 123:9, 164:14, 164:22, 165:4, 229:6, 229:16, 230:5, 230:20, 231:10, 231:17, 231:19
indicates
155:21, 174:8, 175:5, 207:20, 225:12, 281:16, 286:6, 286:14
indicating
183:17, 283:15, 286:13
indication
14:20, 141:19, 142:1, 182:20, 183:2, 183:6, 183:10, 183:13, 183:15, 184:10, 184:13, 197:5, 197:14, 197:21, 221:13, 225:3, 228:10, 228:15, 228:20, 229:4, 229:5, 229:19, 230:4, 230:10, 230:15, 230:18, 231:8, 231:12, 231:15, 231:22, 232:9, 232:12, 233:6, 233:7, 233:10, 233:13, 235:2, 235:19, 235:22, 238:14, 238:18, 241:6, 241:9, 277:3, 277:5, 303:11, 303:18, 304:5, 304:17
indicator
183:16
individual
87:19, 110:15, 111:5, 113:7,

255:16, 286:1
industry
69:3
infer
279:20, 283:8
infers
187:19
influence
23:14, 53:6, 54:10
influenced
23:19
inform
196:14, 228:10, 239:9, 251:17
information
20:19, 21:3, 21:8, 23:4, 23:8, 101:12, 101:18, 102:10, 102:13, 103:2, 112:3, 126:18, 152:18, 152:22, 153:7, 164:1, 173:10, 189:20, 196:19, 228:16, 229:20, 279:10, 279:12, 279:13
informative
162:21, 213:12, 214:3, 214:4, 215:2, 215:13, 217:15, 219:2
informed
23:1, 24:15, 24:19, 24:22, 253:4
informing
229:8, 229:13
informs
228:17, 229:21, 251:4, 251:11, 251:14, 252:6, 252:10
infringes
199:22
initial
215:6
initialized
259:4
initiation
104:10, 104:11, 105:4, 105:7, 106:17
input
41:4, 41:5, 48:12, 64:6, 78:1, 245:7
inside
38:18, 43:19
instance
40:4, 65:22, 66:10, 74:20, 275:22, 277:21, 294:19
instances
238:5, 239:6
instead
8:14, 147:13, 168:7, 302:8
institution
17:9, 18:7, 221:16, 223:2, 223:14, 223:15, 223:21, 224:13
instruction
158:18
instructs
9:2
integer
161:4
intend
248:8
intended
124:10, 133:16, 136:11, 168:1, 168:12, 174:22, 196:11
intending
77:19, 135:2, 136:18, 172:21, 188:17, 233:3, 246:6, 256:8, 256:14, 292:18
intercoded
173:15, 175:14, 177:12, 179:14, 180:18, 181:13
interested
310:15
interesting
152:10, 270:2
intermittent
83:15, 261:11
intermittently
83:14, 85:11, 87:18, 88:9, 89:13, 89:14
internet
61:6, 82:5, 82:11, 82:13
interpredicted
170:20, 171:1, 172:3, 173:20
interprediction
171:3, 173:14
interpret
104:2, 231:5, 232:9, 270:20, 270:22, 299:3
interpretation
259:15, 296:13, 298:17, 299:18
interpretations
240:11, 240:17
interpreted
227:4
interprets
189:6
interrupt
179:5, 298:21
interrupting
121:22
intervals
83:16
intracoded
169:22, 170:14, 171:13, 171:15, 171:17, 171:21, 175:13, 176:20, 177:1, 177:10, 178:10, 179:4, 179:13, 179:20, 180:11
intrapicture
179:3
intrapictures
178:20
introduce
41:11, 42:10
introduced
41:16
introduces
193:16, 242:7
introduction
62:9, 76:10
invalidity
26:10
invention
81:21, 84:17, 87:7, 90:17, 92:4, 98:14, 99:15, 99:18, 100:10, 101:9, 102:2, 102:4, 102:7, 102:9, 103:7, 103:9
inventor
29:14, 84:13, 84:15, 84:16, 85:21, 86:3, 87:2, 87:15, 91:10
inventors
84:14, 99:17, 100:9, 100:15, 104:5
inverse
40:14, 42:17, 42:20, 47:2, 75:9, 77:12, 78:4, 78:5, 78:15
inverses
78:7
involved
20:2, 20:7, 20:9, 24:17, 35:2, 35:8, 35:9, 89:3, 92:1
involves
33:1, 34:6,

61:20, 111:3, 161:3, 247:15, 262:1, 280:11
involving 271:12
ipr 1:9, 2:9, 7:17, 8:5, 9:13, 10:14, 17:13, 17:17, 17:22, 19:2, 19:5, 19:21, 20:15, 20:18, 21:2
iso 6:9, 26:22, 27:6, 29:2, 47:8, 47:11, 49:12, 158:15
isolation 101:22
issues 35:6, 89:3, 240:18
it'll 75:4
itc 20:3, 20:6, 20:9, 20:19, 21:3, 21:8
items 115:4
itself 42:22, 68:1, 84:12, 169:3, 179:18, 180:22
iv 24:12

J

january 236:11
jcarter@perkinsc-oie 3:8
join 245:8
joined 81:4
joining 80:10
jon 3:11
jpeg 37:15
judgment 21:20, 22:8, 23:2, 23:15
jump 171:12, 240:3
jumping 93:1

K

keep 17:14, 26:4, 138:21, 154:1
kendrick 3:10, 5:5, 5:9, 7:14, 56:16, 61:9, 114:1, 182:2, 182:7, 239:16, 284:16, 284:20, 285:1, 307:12, 307:19, 309:6
keyword 9:14
kiara 1:22, 2:21, 8:14, 310:3, 310:20
kiara's 100:3, 154:6
kim's 88:6, 256:9, 260:2, 260:3, 260:12, 261:2, 262:7, 266:3, 266:13, 268:20, 282:12, 282:13, 283:18, 283:22, 284:9, 285:8, 290:8, 290:20, 291:5, 293:16, 294:8, 294:19, 295:4, 302:15
kind 34:5, 47:15, 49:15, 51:17, 53:8, 54:13, 82:19, 86:7, 111:10, 112:11, 116:21, 120:21, 125:3, 125:18, 127:13, 128:10, 129:4, 131:12, 146:9, 175:20, 191:19, 300:7, 302:10
knew 276:22
know 7:18, 7:22, 8:11, 9:18, 10:2, 16:2, 17:21, 18:8, 20:7, 22:12, 36:10, 38:17, 43:10, 44:13, 54:21, 67:22, 70:7, 74:6, 74:18, 78:10, 85:13, 87:16, 91:7, 93:1, 93:2, 94:12, 95:16, 100:5, 101:3, 104:20, 114:22, 115:4, 116:12, 116:14, 119:7, 119:8, 130:5, 131:11, 143:18, 143:20, 156:17, 163:17, 164:4, 164:6, 175:19, 177:16, 189:19, 196:22, 198:21, 223:10, 223:11, 224:1, 224:6, 238:22, 247:12, 253:17, 270:17, 271:10, 272:15, 278:4, 280:20, 281:15, 282:13, 301:18
knowing 198:16, 199:2
knowledge 21:21, 22:9, 23:1, 23:15, 23:20, 49:18, 93:22, 237:17
known 34:14, 35:19, 37:21, 76:15, 77:6, 81:15, 81:21, 89:18, 97:20, 169:12, 269:8, 269:9

L

labeled 10:22, 283:3
labels 141:6, 275:10
language 112:11, 183:4, 250:1, 261:12, 300:10
laptop 7:11, 7:15, 9:13, 13:17, 30:15, 60:22, 221:18
large 43:11, 44:18, 73:19
larger 201:18
last 13:20, 16:10, 18:21, 19:3, 23:13, 66:9, 80:12, 89:22, 91:15, 100:18, 120:1, 130:9, 172:13, 206:14, 213:19, 273:12, 289:4
later 47:9, 58:10, 78:11, 121:10, 165:8, 176:7,

185:13, 203:6, 208:14, 213:1, 213:6, 228:3, 263:21, 274:11
law
94:18
lawyer
90:6, 94:20, 103:11, 220:20, 223:10, 238:22
layer
149:9, 149:10, 149:16, 149:17, 150:3, 150:6, 151:1, 151:7, 151:8, 152:3, 152:16, 152:17, 152:21, 153:7, 153:9, 165:20, 166:6, 166:20, 184:10, 209:20, 209:21, 212:6, 212:17, 226:17, 226:19, 227:10, 227:12, 227:21, 253:9, 253:10
lazy
258:21
lead
296:12, 298:16, 299:17
leading
256:11
least
65:12, 76:7, 99:3, 179:3, 183:6, 183:15, 183:18, 197:6, 197:21, 223:17, 225:3, 225:18, 232:20, 240:10, 241:6, 241:9, 246:9, 262:5, 263:15, 264:6, 264:22, 266:2, 275:19, 291:20, 292:17, 304:10
left
259:9
legal
24:1, 24:2, 24:5, 24:9, 24:12, 91:6, 114:21, 199:20
length
40:6, 40:8, 221:11, 249:15, 290:22
less
57:10, 103:22, 155:5, 155:17, 157:10, 291:13
let's
26:15, 40:20, 51:3, 63:15, 131:10, 131:13, 131:15, 155:16, 186:12, 186:21, 197:2, 215:14, 239:16, 246:15, 255:3, 258:20, 260:8, 284:10
letter
273:12
letters
216:5
level
23:1, 23:19, 32:22, 33:10, 39:14, 40:4, 62:8, 62:12, 76:14, 77:20, 85:7, 127:5, 135:4, 171:10, 189:15, 256:9, 274:12
likely
283:14
likes
46:4, 46:5
limit
83:19
limitation
14:16, 14:19, 15:2, 15:11, 15:16, 15:19, 16:6, 16:8, 16:11, 16:13, 83:1, 110:3, 112:22, 114:10, 115:12, 116:1, 116:2, 116:15, 117:11, 117:16, 117:18, 118:3, 118:11, 118:14, 118:16, 118:18, 118:19, 118:20, 118:22, 119:8, 120:11, 120:12, 122:4, 123:11, 183:1, 183:5, 183:19, 184:9, 195:16, 219:19, 220:19, 221:14, 241:5, 241:8, 246:16, 246:17, 247:2, 247:6, 247:13, 247:14, 247:20, 248:4, 248:13, 248:20, 265:6, 277:1, 290:1, 290:3, 292:1, 292:4, 292:10, 292:11, 292:14, 294:10, 294:11, 294:16, 306:1
limitations
14:14, 31:14, 91:3, 93:7, 93:11, 93:13, 265:3, 265:10, 265:11
limited
183:12, 183:13, 282:8, 282:17, 284:10
limiting
94:2
limits
84:9, 85:17
line
15:10, 16:5, 57:9, 84:21, 86:6, 92:16, 98:21, 99:1, 99:21, 99:22, 107:10, 168:5, 189:4, 256:12, 257:12, 261:9, 261:13, 261:18, 268:21, 269:16, 271:15, 271:16, 272:6, 272:20, 273:5, 273:12, 282:2, 282:22, 283:11, 284:4, 286:10, 291:11
lines
59:19, 98:9, 98:18, 100:19, 145:2, 145:3, 272:15, 272:16, 289:13, 289:17
list
17:5, 17:6, 35:14
listed
22:15, 23:9, 91:14
listing
26:17
lists
55:4
litigation
14:4, 91:1, 238:2
litigations
14:2, 90:20
little
9:21, 16:22, 96:2, 103:14, 141:10, 174:3, 174:13, 181:11, 192:7, 274:12
llc
1:5, 2:5
locate
284:1, 284:14
located
264:2, 268:14
logical
120:10, 256:22,

278:20, 281:12, 285:8, 285:10, 285:13, 286:15, 302:8, 307:2, 307:5
long
18:19, 19:8, 19:20, 45:8, 45:17, 46:4, 46:5, 49:20, 50:6, 51:18, 52:9, 52:13, 58:17, 99:14, 153:16, 169:2, 183:21, 305:21
long-winded
256:17
look
14:6, 14:17, 15:9, 16:5, 28:9, 49:16, 53:12, 60:22, 61:4, 63:12, 78:10, 78:12, 78:14, 92:16, 93:6, 97:17, 99:19, 103:8, 103:17, 109:3, 109:4, 109:18, 114:9, 124:20, 134:19, 135:11, 154:19, 159:1, 161:8, 164:7, 167:10, 174:10, 174:12, 185:16, 186:14, 188:5, 189:14, 190:20, 197:2, 201:17, 206:13, 206:19, 207:16, 213:19, 214:14, 215:14, 236:13, 242:2, 255:13, 257:9, 258:3, 260:12, 271:19, 272:5, 272:17, 273:11, 275:11, 276:13, 287:14, 290:8, 293:15, 303:10
looked
165:8
looking
27:3, 28:5, 28:18, 29:9, 38:20, 42:12, 66:11, 77:13, 103:18, 112:13, 123:19, 134:1, 135:7, 136:4, 147:10, 188:12, 188:18, 202:22, 208:21, 209:11, 209:14, 215:10, 219:18, 221:21, 225:19, 226:7, 240:20, 250:12, 258:9, 261:6, 266:5, 273:6, 277:1, 280:8, 282:16, 285:16, 289:22, 300:19
looks
61:13, 64:20, 272:10, 283:1, 289:9
loss
41:11, 41:15, 41:18, 42:11, 104:3, 128:14, 187:19, 188:14, 189:7
lossy
40:17, 40:18, 41:12, 42:18, 42:22, 64:15
lost
58:17, 188:8
lot
67:22, 94:15, 140:7, 196:18, 249:4, 270:4, 271:2
lower
86:18, 87:1, 87:9
luminance
53:17
lunch
176:9, 182:3, 182:11

M

m-a-l
69:21
macroblock
171:10, 171:11, 171:18
macroblocks
133:10, 161:10, 171:13, 171:15, 171:17, 171:21, 173:3, 173:13, 173:15, 173:19
made
9:21, 200:2, 205:17, 304:1
make
13:10, 14:15, 32:4, 50:8, 52:5, 52:9, 52:21, 52:22, 53:5, 53:8, 54:10, 54:13, 100:4, 126:13, 138:11, 147:20, 155:13, 168:3, 179:12, 200:1, 257:10, 260:2, 270:4, 270:5, 271:1, 273:9, 281:20, 304:2
makes
51:22, 52:11, 54:15, 82:4, 137:1, 279:19
making
32:7, 54:1, 89:11, 89:19, 114:20
mal
69:21
manner
58:7, 125:19, 131:18
many
22:13, 22:14, 38:11, 47:6, 64:12, 64:13, 83:1, 91:12, 238:5, 239:6
march
236:7
marked
7:7, 7:9
marking
234:14
markings
15:5
marriage
310:15
materials
21:19
matter
12:14, 12:19, 25:1, 28:19, 29:10, 61:7, 88:16, 88:22, 89:3, 259:6, 259:14, 298:19, 298:20, 299:2, 310:16
matters
14:5, 24:19, 238:2, 267:16
maybe
70:10, 77:9, 80:15, 83:6, 91:19, 138:13, 146:1, 157:11, 177:4, 187:19, 248:21
mckool
3:14
mean
12:20, 44:8, 45:11, 50:12, 51:12, 54:21, 62:3, 62:22, 63:14, 63:20, 69:4, 70:11, 75:17, 80:18, 90:12, 102:2, 105:1, 106:13, 107:16, 110:6,

110:10, 110:11, 110:18, 110:20, 111:4, 116:22, 117:9, 118:4, 123:3, 134:12, 159:12, 160:6, 165:16, 167:9, 179:5, 185:8, 191:9, 199:14, 200:9, 224:5, 232:15, 238:22, 245:21, 250:10, 254:22, 256:5, 264:12, 264:17, 268:17, 270:20, 273:21, 275:1, 280:17, 283:22, 288:5, 295:8, 295:11, 295:21, 302:17, 306:9
meaning
44:5
meanings
36:9
means
75:22, 76:1, 91:7, 107:18, 166:9, 166:11, 170:15, 175:8, 176:21, 238:14, 239:1, 253:16, 291:17, 296:13, 298:17, 299:18
meant
62:8, 63:5, 63:7, 75:21, 76:9, 77:4, 113:2, 113:3, 113:4, 118:8, 289:19, 302:12
mechanism
145:16, 187:21
media
55:3, 55:5
medications
9:9
medium
282:21
meet
18:19, 19:8, 159:22, 247:5
meeting
19:11
meets
58:21
memory
9:10, 67:6, 73:20, 159:15, 236:14, 258:20, 259:1, 259:4, 269:13, 269:21, 272:11, 292:9
mention
216:13
mentioned
9:12, 18:5, 32:15, 34:17, 42:7, 60:12, 69:1, 69:5, 158:12, 158:22, 196:1, 244:9
merely
34:19
met
5:8, 16:20, 18:15, 18:20, 19:4, 20:13, 51:19
metaphorical
224:4
method
14:8, 14:9, 88:6, 89:12, 95:4, 95:6, 109:22, 110:21, 111:3, 112:13, 114:18, 115:1, 115:8, 121:8, 122:6, 122:7, 122:12, 143:4, 143:5, 143:9, 143:10, 143:12, 191:7, 191:13, 192:2, 194:3, 194:9, 194:21, 195:3, 195:6, 195:12, 195:17, 196:9, 197:2, 197:4, 197:10, 198:10, 198:19, 247:9, 247:17, 260:2, 261:10
methods
145:9, 145:11
middle
36:7, 154:22, 246:21
might
6:20, 7:1, 15:17, 27:16, 33:1, 34:1, 39:14, 46:18, 52:17, 53:9, 61:5, 76:16, 77:8, 82:2, 82:20, 115:9, 127:18, 128:2, 129:13, 132:15, 138:16, 138:17, 138:21, 178:3, 180:15, 181:8, 181:16, 190:18, 208:17, 305:17
miller
1:22, 2:22, 310:3, 310:20
mind
60:10, 68:18, 76:20, 87:21, 288:4, 289:19, 307:9
minimal
88:11, 88:19, 93:8, 93:10, 93:14
minute
303:1
minutes
307:14
mirror
79:2
mischaracterized
80:16
misreading
263:5
missed
105:16
missing
125:12, 126:2, 126:16, 127:7, 128:2, 129:7, 138:6, 138:11, 157:11, 272:3
misspoke
177:4
misstates
74:16
mistaken
170:12
misunderstanding
248:21
mixed
306:19
moment
181:18
momentarily
188:15
monday
19:12
money
9:21
months
18:21
more
16:4, 22:13, 57:10, 58:9, 103:22, 131:21, 146:7, 167:6, 167:17, 169:8, 172:2, 172:7, 176:8, 227:9, 263:21, 275:5, 300:18, 306:1
morning
5:6, 5:7
most
173:5, 283:14, 302:14, 306:4
motion
37:15, 40:1, 47:6, 166:12, 166:16, 174:16, 174:21, 176:15,

178:14, 178:18, 179:10, 180:3, 180:8, 181:2, 181:4, 207:22, 208:8, 209:1, 209:15, 210:1, 210:6, 210:13, 211:15, 212:10, 212:14, 212:19, 213:10
move
25:3, 80:6, 252:19
moving
84:2, 86:10, 87:5, 88:5, 88:7, 88:13, 88:20, 203:13, 252:13, 269:2, 273:3, 273:17, 274:5
mpeg
43:14, 161:5, 291:15, 302:22
mpeg-1's
220:1, 220:4, 220:18, 225:1, 235:17, 235:18, 249:14
mpeg-1s
157:18
mpeg-2
161:3
mpeg-4
10:9, 60:17, 69:3, 291:21
much
5:14, 30:7, 63:3, 79:17, 107:9, 119:8, 137:1, 154:22, 166:17, 176:17, 291:13, 301:18, 306:9, 307:20
multimedia
85:3, 86:2, 90:14
multiple
100:13, 109:2, 145:9, 183:20, 184:4, 185:20, 260:20, 261:3, 287:3
must
45:6, 45:14, 152:11, 156:5, 157:22, 158:18, 165:12, 178:9, 204:15, 204:20, 206:3, 211:1, 228:10, 228:20, 230:11
myself
15:4, 265:7

N

n-a-l
69:12
nal
69:11, 69:15, 69:17, 69:19, 70:8, 71:1, 71:3, 71:15, 71:16, 71:18, 72:9, 73:1, 74:1, 74:13, 233:16, 233:17, 233:21, 234:3, 234:15, 236:2, 237:3, 303:3, 303:5, 303:10, 303:11, 303:16, 303:18, 304:4
nal_unit_type
70:1, 70:4, 71:3, 72:4, 72:18, 304:16, 305:17
nal_unit_types
305:15, 305:16
name
5:9, 5:11
named
91:10
names
68:10, 68:13, 81:22
natural
164:10, 164:13, 164:21, 165:3
navigate
61:15
near
171:14
nearest
146:2, 288:22
necessarily
17:21, 19:2, 22:15, 40:11, 41:8, 58:22, 64:16, 64:21, 77:10, 83:19, 99:7, 139:1, 152:19, 154:15, 191:18, 211:12, 212:3, 228:10, 228:17, 229:6, 229:13, 229:21, 230:5, 243:1, 244:13, 245:5, 267:14
necessary
120:10, 124:22, 189:21, 199:16, 252:18, 252:19, 279:18, 281:17, 305:9
need
7:21, 16:1, 47:5, 57:21, 61:5, 67:20, 70:7, 79:2, 103:12, 153:6, 158:4, 164:9, 174:11, 188:21, 196:18, 205:9, 270:13, 280:21
needs
153:6, 253:17
neighbor
170:16
netflix
82:3, 84:1
network
86:3
never
208:19, 248:17, 258:22, 259:1, 293:8, 293:22, 306:6, 307:7
new
1:17, 2:20, 5:3, 310:5
next
27:19, 29:19, 77:18, 94:2, 99:21, 148:19, 164:7, 186:1, 186:15, 189:14, 205:11, 227:1, 227:6, 245:2
nicholas
4:8
nick
4:7
nodding
8:14
nokia
1:11, 2:13, 20:20, 21:4, 21:9, 24:7
nokia's
10:21, 11:9
nonstandard
56:9
normal
191:4
normative
162:22, 217:17
nos
8:14
notary
5:2, 308:13, 310:4
nothing
138:6, 198:21, 258:8, 268:3, 278:3, 279:9, 279:11, 279:15, 279:19, 279:21, 280:7, 280:12, 280:17
notice
28:21

noticed
167:22
nowadays
82:1, 86:8
nuance
41:22
number
6:10, 12:18, 14:4, 17:11, 17:12, 18:20, 24:18, 30:11, 30:12, 30:16, 39:2, 43:11, 67:5, 73:16, 74:19, 84:3, 84:9, 85:17, 88:11, 88:19, 95:6, 105:7, 111:22, 119:16, 128:12, 141:7, 156:3, 169:16, 185:18, 204:17, 218:3, 218:8, 226:20, 275:12, 287:22, 288:1, 288:2, 297:13
numbering
97:8, 103:21, 104:11, 109:14, 116:4, 118:21, 189:2, 189:5
numbers
201:21, 202:2, 215:4, 215:5, 215:9, 215:11, 215:20, 216:4, 216:13, 216:15, 216:17, 217:4
numeral
24:11
numerous
14:1

O

oath
5:18, 5:20
object
9:1, 43:5, 45:10, 48:19, 86:21, 92:5, 132:2, 177:3, 177:15, 217:12, 218:14
objection
20:4, 32:21, 34:11, 36:20, 37:19, 38:7, 50:11, 54:19, 57:15, 60:1, 60:8, 65:5, 65:20, 66:5, 66:14, 66:20, 67:2, 67:17, 68:7, 70:5, 71:19, 72:13, 73:4, 73:13, 74:4, 74:15, 75:14, 75:18, 78:22, 83:4, 88:14, 91:5, 92:22, 94:11, 95:3, 96:9, 99:12, 102:16, 103:4, 106:6, 106:12, 107:17, 108:22, 112:17, 113:11, 117:8, 117:19, 120:16, 123:12, 126:8, 127:8, 128:8, 131:9, 132:19, 138:3, 139:12, 140:15, 142:22, 143:8, 143:17, 144:14, 154:14, 159:11, 160:17, 163:11, 164:3, 165:1, 167:8, 175:16, 184:1, 192:20, 194:13, 195:8, 197:16, 199:13, 199:19, 201:6, 201:10, 205:15, 206:11, 210:8, 211:11, 219:21, 222:20, 223:22, 227:19, 238:10, 238:15, 239:12, 241:20, 247:7, 248:1, 248:16, 249:21, 251:7, 254:4, 254:12, 257:8, 259:18, 260:21, 263:4, 263:19, 264:13, 267:13, 267:21, 280:2, 281:13, 287:5, 287:19, 292:7, 296:2, 297:22, 299:5, 302:19, 303:8, 303:14, 304:7, 304:19
objections
133:6, 198:12, 198:14, 223:3
objective
265:4
obtain
65:11, 65:18, 66:3, 66:18
obtained
269:3
obvious
28:8, 28:20, 29:7, 29:11, 30:4, 30:9, 31:1, 31:3, 31:20, 32:1, 128:3, 129:21, 144:21, 194:19, 221:7, 221:8, 223:7, 248:3, 248:4
obviously
11:11, 15:5, 67:4, 99:13, 249:1
obviousness
24:20
occur
80:5, 100:16, 125:10, 126:1, 126:7, 139:11
occurs
103:21, 137:13, 140:2, 140:18, 189:5, 208:3
offer
223:19, 224:11
offered
224:14, 239:8
offering
118:1, 238:7
office
1:1, 2:1
official
27:5
often
37:5, 86:8
oh
69:9, 125:3, 203:9, 269:17, 273:7, 288:20, 296:22
oldest
162:1, 163:9, 163:15, 163:17
omit
206:15
omitted
206:9
once
55:10, 68:6, 76:7, 168:21, 227:7, 261:17
ones
12:1, 17:2, 91:15, 94:18, 169:14
online
16:21
only
30:22, 56:1, 58:5, 59:4, 170:8, 170:12, 171:21, 172:15, 172:22, 175:14, 177:1, 177:12, 179:14, 200:7, 201:13, 220:8, 222:10, 225:12,

234:19, 235:8, 235:13, 235:19, 237:6, 293:20, 294:4, 301:19, 302:11
onwards 194:14, 265:1
open 9:15, 13:16, 28:14, 61:2, 61:6, 166:7, 187:16, 211:7, 211:9, 211:16, 212:1, 213:14, 213:21, 221:20, 225:13, 227:1, 235:21
opened 9:18, 30:14, 211:4, 211:20, 213:5
operates 41:1
operation 47:3
opine 94:8, 198:9, 216:14, 225:1, 239:7, 257:5
opined 31:2, 238:3, 238:6
opining 238:11, 238:13, 238:16
opinion 31:13, 31:22, 79:15, 82:22, 89:8, 95:11, 95:20, 96:3, 96:7, 96:10, 112:19, 116:9, 116:20, 117:4, 117:14, 117:22, 118:1, 120:15, 120:18, 120:20, 120:22, 121:1, 121:16, 122:2, 123:15, 143:19, 182:19, 184:21, 188:3, 192:16, 193:6, 194:16, 198:3, 221:4, 221:10, 221:12, 223:4, 223:5, 223:8, 223:19, 224:9, 224:11, 224:14, 224:18, 225:11, 225:17, 228:9, 228:13, 228:18, 228:19, 229:2, 229:18, 230:3, 230:13, 230:18, 231:21, 233:12, 239:2, 240:9, 240:21, 244:11, 255:2, 257:20, 258:11, 260:16, 262:8, 262:21, 277:2, 277:8, 277:15, 292:2, 292:4, 292:10, 292:13, 295:2, 296:5, 296:12, 299:17, 301:16, 301:19, 302:4, 302:18, 306:4, 306:16
opinions 12:1, 12:7, 12:12, 12:17, 14:4, 21:17, 22:4, 22:19, 23:6, 23:9, 31:17, 31:20, 32:8, 94:13, 121:5, 223:15, 249:7, 264:22
optical 85:12, 145:17, 261:14, 261:16, 267:9, 268:10, 282:6, 282:11, 282:14, 282:21, 283:3, 283:6, 284:7
optionally 169:7
options 45:7, 45:16, 47:7, 52:1, 52:6, 147:20
orders 132:6, 132:8
ordinary 25:7, 81:15, 82:20, 111:1, 162:18, 195:2, 198:8, 215:19, 228:14, 229:3, 230:14, 269:6, 270:19, 272:1, 274:7
organized 167:12
orient 13:15
original 41:8, 62:16, 64:1, 64:10, 64:17, 64:19, 64:22, 65:3, 206:21, 207:6, 214:5, 261:20, 262:3, 269:13, 269:22, 273:15, 274:4, 280:5
other 5:16, 6:16, 7:10, 10:7, 10:10, 16:18, 17:2, 17:16, 18:2, 18:10, 19:16, 20:16, 23:8, 24:19, 31:21, 32:7, 32:9, 33:5, 34:21, 36:9, 37:3, 43:2, 44:3, 55:4, 57:22, 59:6, 77:20, 78:6, 78:7, 82:6, 94:18, 118:10, 125:20, 136:19, 162:16, 163:18, 170:1, 171:6, 173:10, 175:1, 175:15, 177:2, 177:11, 178:11, 179:20, 195:22, 205:9, 207:2, 210:7, 214:8, 215:8, 216:4, 219:15, 221:8, 232:18, 232:19, 235:15, 241:17, 241:18, 244:8, 244:9, 291:2
others 18:13, 18:14, 30:16, 71:12, 169:13
otherwise 59:18, 92:11, 177:5
ourselves 13:15
out 9:17, 23:3, 23:17, 23:22, 37:5, 37:7, 37:16, 37:18, 37:22, 39:13, 39:15, 39:22, 40:3, 48:5, 48:8, 49:1, 50:5, 58:7, 59:1, 71:5, 75:8, 92:7, 97:6, 98:19, 105:16, 145:8, 189:3, 193:10, 216:18, 217:6, 217:7, 220:15, 233:3, 240:18, 240:22, 249:4, 253:3, 262:4, 284:11, 299:9, 300:4, 300:12, 300:16
outcome 310:16

output
41:6, 42:6, 48:13, 49:3, 49:5, 50:1, 50:6, 64:9, 64:16, 65:15, 77:21, 79:6, 127:17, 128:1, 273:3, 273:16, 273:17, 274:6, 291:13
outside
8:7, 17:2, 23:4, 35:10, 66:11, 71:11, 87:10, 93:16, 94:7, 143:1, 181:14, 200:15, 301:9, 304:20
over
18:21, 22:22, 32:16, 35:3, 45:22, 46:11, 70:19, 90:21, 91:1, 92:1, 142:11, 236:19, 253:22
overall
111:15
overlap
33:12
overview
84:19, 85:7, 162:8, 162:15
owner
1:12, 2:14, 3:13, 4:3, 10:21, 11:10
owner's
7:1
owners
6:19
oy
1:11, 2:13

P

p-frame
135:9, 136:9, 136:14, 136:22, 172:14, 173:1, 173:3, 173:7, 266:22, 268:9, 268:22, 270:11, 290:18
p-frames
255:13
p-picture
161:15, 161:21, 162:6, 170:20, 171:1, 171:4, 171:5, 171:20, 173:6, 173:12, 173:14, 173:18, 173:20, 201:1, 255:19, 255:20, 256:17, 256:18, 263:16, 263:17, 263:22, 264:1, 264:4, 264:10, 264:19, 265:18, 265:20, 266:2, 266:7, 266:9, 266:10, 267:1, 267:2, 267:4, 267:5, 267:10, 268:12, 268:15, 270:8, 271:6, 273:2, 273:16, 274:5, 276:20, 278:13, 278:21, 283:5, 286:9, 286:11, 286:15, 287:9, 287:11, 287:16, 287:22, 288:1, 291:14, 293:15, 294:15, 294:20
p-pictures
169:9, 171:16, 172:3, 255:19, 279:4, 287:3, 287:16, 288:22, 290:10
p1
286:4, 286:6, 286:16
p2
286:4, 286:7, 286:17, 288:7, 288:8, 288:11, 289:15
p4
288:9, 288:12, 289:15
p6
186:13, 186:19, 187:15
packet
128:13
pages
68:3, 124:3, 146:16, 195:22
pair
61:20, 62:3
paper
221:17
papers
7:16, 9:13
paragraphs
24:21, 32:3, 32:10, 140:7, 176:5, 185:11, 232:4, 232:7, 232:19, 233:1, 233:2, 265:1
parameter
69:21
parameters
71:9
paraphrase
46:19, 63:3, 154:1, 265:15, 268:22
paraphrasing
47:13, 63:11, 107:13, 165:10, 177:16, 293:6
parentheses
126:2, 126:3, 168:6, 227:2
parse
160:22
parses
159:4
part
12:17, 17:6, 19:13, 19:15, 26:21, 27:6, 27:7, 27:10, 41:2, 42:3, 47:8, 47:11, 49:8, 53:17, 54:7, 60:17, 71:12, 71:16, 76:4, 104:1, 108:7, 108:14, 108:20, 109:8, 109:16, 110:4, 110:6, 112:15, 112:22, 114:18, 118:14, 122:4, 122:14, 122:20, 122:22, 124:14, 124:15, 125:16, 125:17, 129:3, 129:4, 129:8, 130:21, 134:7, 145:15, 150:3, 150:22, 152:4, 152:14, 160:3, 160:4, 162:21, 166:5, 171:9, 172:4, 175:7, 175:9, 177:5, 177:6, 178:11, 179:17, 188:12, 214:2, 215:2, 216:20, 217:17, 227:21, 228:1, 236:1, 247:17, 248:7, 249:14, 250:22, 265:10, 266:10, 266:15, 267:7, 267:18, 267:19, 268:1, 268:4, 270:17, 271:1, 272:2, 278:13
partial
98:21, 129:2, 138:5, 295:13
particular
15:6, 19:21,

22:17, 59:17, 108:2, 136:11, 142:12, 168:21, 176:3, 184:4, 232:17, 236:8
particularly 84:17, 90:14, 90:15, 295:17
parties 14:13, 310:14
partition 224:3
parts 146:9, 179:20, 196:1
partway 80:10, 81:5
pass 307:12
passage 108:12
past 18:6
patents 90:6, 91:8, 91:10, 91:13, 91:22
path 274:15, 274:17, 275:2, 275:10, 276:1, 276:4, 276:8, 276:17, 276:19, 277:2, 277:3, 277:8, 278:13, 278:14, 279:7, 281:4, 294:8, 294:9
paths 274:22, 275:4
pdf 9:15, 188:20, 221:20, 275:11, 275:12, 275:22
peachtree 4:5
peles 53:18
pending 8:1
perfect 29:20
perfectly 242:10, 275:3
perform 36:18, 47:5, 187:10
performed 35:19, 36:1, 36:12, 36:17
performs 272:20
perhaps 41:15, 58:9, 76:20, 77:7, 77:19, 83:22, 202:19, 245:6
period 22:22
perkins 3:4
person 25:7, 81:15, 82:20, 111:1, 114:19, 162:18, 195:2, 198:8, 215:19, 216:22, 228:13, 229:2, 230:14, 232:8, 269:6, 270:19, 272:1, 274:7, 298:15, 300:18, 301:1
perspective 15:21, 64:20, 128:1, 140:10
petition 12:16, 12:19, 17:8, 21:18, 26:9, 28:9, 28:14, 28:18, 28:19, 29:5, 29:9, 30:6, 30:7, 31:10, 147:6, 202:5, 202:19, 203:1, 203:10, 221:1, 275:12, 293:18, 294:7
petitioner 1:7, 2:7, 2:19, 3:3, 25:18, 26:1, 26:2, 26:4, 26:7, 26:14, 28:5, 28:21, 29:11, 31:10, 202:10, 204:8, 205:2, 205:5, 205:16, 205:21, 220:22, 221:3, 222:5, 225:17, 275:4, 275:8, 275:10, 276:3, 276:7, 276:19, 279:1, 293:19, 294:3, 294:22, 306:14, 307:2, 307:10
petitioner's 25:20, 26:9, 28:6, 29:4, 202:5, 202:14, 203:14, 203:17, 204:5, 221:5, 240:21, 278:14, 281:4
petitioners 30:18, 203:10, 204:6, 205:11, 274:15
phrase 51:6, 77:2, 86:15, 119:3, 119:7, 119:9, 224:5, 256:5, 297:18, 298:2, 298:4, 299:21, 301:1, 301:5, 301:9, 304:10
phrases 224:3, 298:7
physical 278:20, 281:18, 283:4, 283:7, 283:10, 283:13, 283:17, 284:2, 284:13, 285:8, 285:11, 285:14, 301:20, 302:9, 302:11, 302:14
pick 186:12
piece 270:22, 272:4
pixel 126:13
pixels 53:18
place 84:9, 177:22, 179:19
placed 254:17
places 46:22, 181:16, 220:2
placing 83:5, 85:16, 161:6
planned 19:17
playback 85:2, 85:4, 185:1, 187:22, 191:3, 191:4, 193:11, 241:14, 242:3
player 242:20
playing 86:1
please 5:11, 7:22, 8:11, 13:19, 16:2, 103:18, 106:7, 124:19, 221:17, 299:10, 301:7
plus 23:13, 63:17, 68:2, 195:22
point 10:3, 18:6, 89:10, 89:19,

97:9, 101:11, 105:11, 106:3, 106:9, 106:14, 106:19, 107:1, 137:8, 142:8, 142:12, 146:3, 146:4, 172:20, 175:20, 186:12, 187:9, 187:22, 188:22, 193:12, 193:22, 202:19, 210:6, 227:7, 242:3, 244:7, 245:9, 246:9, 249:4, 262:5, 263:12, 263:13, 301:7
pointed 240:18
pointing 126:21, 174:17, 176:15, 178:18, 179:10, 180:3, 181:5, 207:22, 208:9, 209:1, 209:16, 210:1, 210:7, 210:14, 210:21, 212:11, 212:15, 212:20, 216:18
points 84:11, 105:6, 185:1, 189:3, 307:2
portion 78:14, 126:9, 126:11, 126:12, 126:15, 232:1, 232:3
portions 27:21, 91:19, 129:17
posita 25:12, 25:17, 81:20, 87:22, 89:17, 112:7, 175:4, 179:1, 231:5, 231:19, 231:21, 251:22, 253:19, 257:20, 258:15, 274:2, 279:16, 280:14, 282:16, 284:12, 293:10, 295:6, 295:10, 295:14, 295:15, 296:5, 296:11, 299:16
posita's 238:18
position 221:6
possibilities 244:9
possibility 136:19
possible 15:18, 36:7, 49:15, 76:15, 99:2, 162:15, 162:17, 163:19, 242:10, 242:19, 275:5
possibly 6:10, 10:10, 77:4, 92:14, 291:20
post-filtering 35:11
potential 139:19, 185:14
practical 64:12, 242:11
practice 194:18, 199:11, 199:12, 199:15, 199:17, 246:12, 246:13
practices 195:15, 198:19
practicing 198:9
pre-filtering 35:11
preamble 14:7, 14:12, 109:18, 114:17, 114:21, 122:9
precede 242:18
preceded 169:7, 242:14
preceding 99:3, 173:17, 206:8, 206:16
precise 49:21, 63:10, 68:12, 104:21, 185:2
precisely 59:1, 62:10
predecessor 55:9
predicted 126:18, 130:20, 131:5, 133:8, 133:14, 256:19
predicting 140:5
prediction 37:8, 37:16, 37:18, 37:22, 125:6, 126:3, 126:15, 131:3, 133:20, 136:16, 136:20, 172:4, 172:22, 173:6, 173:17, 174:1, 220:7, 222:9, 255:10, 255:11, 255:16, 255:18, 256:1
predictions 125:21, 131:2, 131:3, 133:15
preface 76:5
prefer 26:4, 36:6
preferably 105:12, 105:13, 105:17, 105:19, 107:1, 107:11, 107:13
preference 147:1
preferred 92:3, 92:15, 92:17, 92:21, 93:4
preliminary 6:19, 10:22, 11:16
premised 240:10, 240:17
preparation 17:18
prepare 16:15, 19:9
preparing 19:14, 19:15, 20:17, 21:1, 21:7
presence 230:19, 231:9, 231:16, 231:18
present 127:12, 130:3, 228:20, 229:5, 229:9, 229:12, 230:4
presented 158:8
presenting 229:2
presumably 145:15
pretty 30:7, 63:3, 79:17, 107:9, 129:20, 141:13, 154:22, 166:17, 176:17, 252:8, 295:8, 306:9
previous 42:12, 44:16, 97:9, 120:3, 120:13, 166:11, 166:14, 166:15, 171:3, 173:5, 173:21, 174:2, 174:17, 174:21, 175:10, 176:16, 178:14, 178:19,

179:11, 180:4, 180:7, 181:1, 181:5, 207:22, 208:9, 209:2, 209:16, 210:14, 210:22, 211:15, 212:11, 212:15, 212:20, 213:10, 239:10, 286:14, 286:22, 298:15
previously
133:1, 133:21, 137:15, 140:4, 140:14, 172:15, 172:22, 173:4, 255:12, 256:1, 256:6, 299:8, 301:15
primarily
41:15
principle
280:18
prior
12:15, 74:16, 99:18, 100:10, 189:10, 190:11, 236:19, 262:3
priority
236:10, 236:18, 238:8, 239:9, 291:18
probably
17:10, 35:14, 49:16, 68:2, 75:20, 76:7, 168:1, 168:11, 224:16, 270:21
problem
27:2, 28:17, 30:13, 47:18, 69:7, 83:11, 96:1, 97:7, 97:10, 97:12, 99:17, 100:9, 103:20, 104:1, 104:8, 105:2, 139:16, 139:19, 140:6, 140:10, 177:7, 179:6, 179:18, 189:1, 189:2, 272:18, 276:12, 278:9, 306:21
problematic
189:16
problems
100:13, 100:16, 185:14, 186:3, 188:16, 189:8
procedure
101:8, 102:2, 102:4, 102:6, 102:9, 102:21, 102:22, 103:8
proceed
127:15
proceeding
16:20, 18:3, 18:11, 20:3, 20:6, 20:8, 20:10, 20:20, 21:3, 21:9, 66:16, 224:9
proceedings
17:13, 19:2, 20:16, 24:18
process
33:14, 35:18, 35:22, 36:1, 36:11, 36:16, 36:19, 38:4, 38:22, 39:12, 40:17, 40:18, 41:9, 41:13, 41:17, 41:18, 42:17, 42:18, 42:21, 42:22, 43:9, 43:16, 43:20, 44:11, 44:15, 45:1, 45:4, 46:9, 46:10, 46:12, 47:7, 47:14, 47:20, 48:5, 48:7, 48:11, 48:14, 49:5, 50:2, 50:4, 54:5, 55:13, 56:9, 58:2, 58:3, 58:14, 65:12, 65:13, 67:12, 72:19, 72:21, 75:8, 77:11, 79:7, 79:9, 79:21, 80:22, 92:7, 111:20, 111:21, 115:7, 144:16, 145:6, 147:12, 157:7, 157:18, 158:1, 158:6, 158:20, 190:10, 196:9, 252:2, 252:3, 261:22, 267:3, 268:19, 271:12, 274:10, 280:11, 289:20
processer
147:14
processes
33:5, 34:6, 34:12, 34:14, 34:15, 34:20, 34:21, 35:4, 35:10, 35:17, 37:6, 37:8, 37:22, 39:14, 39:18, 50:16, 59:2, 76:3, 77:2, 77:3, 77:5, 79:13, 111:20, 112:1, 148:6
processing
37:3, 74:9
processor
274:2
produce
37:9, 39:9, 39:11, 42:6, 48:13, 50:1, 58:12, 58:20, 75:10, 79:6, 157:16, 157:22, 158:4, 158:19
produced
48:14, 150:16, 150:17, 196:6, 200:4, 269:19, 299:20
produces
43:9, 43:17, 45:21, 48:6, 49:3, 53:3, 58:4, 58:19, 59:4, 65:12, 70:13, 70:18, 70:20, 71:21, 72:2, 76:18, 79:8, 148:2, 150:1, 150:21, 151:6, 157:1, 157:3, 251:1
professional
21:20
profiles
71:14
program
258:21
programmed
74:22
programmer
258:22
programming
259:7
prohibit
132:11
properties
202:15, 203:11, 203:13, 204:1, 205:19
property
180:12, 203:14, 204:7, 204:10, 204:11, 204:14, 205:1, 205:6, 205:8, 205:11, 208:14, 212:19
proposed
306:14, 307:10
proprietary
38:17

provide
8:18, 16:3, 20:12, 31:20, 55:7, 94:13, 102:13, 117:21, 203:4, 232:8, 233:5, 233:7, 233:12, 233:13, 256:8
provided
7:12, 12:13, 13:17, 20:13, 24:6, 24:9, 30:15, 32:9, 61:1, 82:9, 82:13, 84:2, 129:18, 199:5, 208:1, 208:2, 234:7, 234:19
provider
82:3, 82:9
provides
82:4, 101:11, 101:17, 102:9, 103:1, 180:11, 190:6
providing
12:17, 31:16, 32:7
proviso
15:17, 16:9, 16:14
public
5:2, 291:19, 308:13, 310:4
publication
236:6, 236:20
published
39:2, 39:3, 60:15, 60:18, 61:3, 236:22, 237:3, 237:14
purchased
9:20
purposes
17:18
purview
304:21
put
27:20, 43:18, 48:2, 55:8, 86:4, 86:18, 87:10, 129:3, 175:19, 178:5, 178:6, 180:9, 224:3, 229:1, 230:7
puts
178:11, 181:22
putting
87:1

Q

quality
35:12, 64:21
quantization
40:14, 40:16, 40:17, 41:16, 42:17, 42:20, 78:13, 262:2, 280:6, 280:11
quantize
78:3, 78:5
questioning
57:9
questions
8:9, 10:1, 16:1, 16:4, 28:15, 29:20, 49:18, 59:8, 176:8, 276:15, 307:17
quickly
37:20, 169:19
quite
35:15, 39:2, 43:13, 54:21, 73:16, 85:13, 90:19, 95:9, 95:10, 96:21, 99:14, 153:16, 172:6, 192:1, 208:19, 218:9, 229:1, 229:17, 239:4, 262:17, 284:10, 303:17
quoting
97:5, 106:15

R

rain
8:7
ran
300:7
random
79:16, 96:4, 96:8, 96:11, 96:13, 96:18, 96:20, 97:1, 97:9, 97:20, 99:10, 99:18, 100:10, 101:11, 105:6, 105:11, 106:3, 106:9, 106:14, 106:19, 107:1, 123:20, 123:21, 130:2, 130:4, 137:7, 139:17, 139:20, 139:22, 142:8, 142:12, 184:18, 184:20, 184:22, 185:9, 185:15, 186:12, 187:10, 188:4, 190:4, 190:5, 190:6, 190:9, 190:10, 190:12, 191:3, 191:8, 191:12, 191:14, 191:17, 191:21, 192:3, 193:11, 242:3, 242:20, 243:2, 243:5, 244:4, 244:22
randomly
79:22, 81:1, 81:3
rate
90:4, 90:10, 93:9, 93:14
rather
26:13, 30:12, 146:21, 175:21, 270:22, 278:20
raw
33:2, 33:8, 33:16, 33:19, 34:7, 34:9
rd
236:11
re-ask
277:7, 292:3
reach
301:16
reached
267:16
reaching
249:6
read
5:18, 54:5, 55:12, 56:1, 63:3, 90:6, 90:8, 97:6, 98:19, 100:4, 100:5, 101:2, 103:12, 104:13, 104:19, 110:4, 145:13, 168:7, 183:9, 187:12, 205:22, 212:5, 222:13, 223:1, 223:20, 224:12, 253:20, 254:1, 260:14, 264:1, 264:3, 266:7, 268:12, 268:14, 295:6, 295:10, 296:6, 299:9
reader
87:7
readers
76:11
reading
30:6, 54:22, 63:2, 63:11, 77:7, 101:22, 111:1, 114:18, 117:11, 122:8, 147:22, 156:2, 175:4, 180:5, 202:8, 213:20,

230:15, 236:17, 254:18, 257:21, 258:15, 260:3, 260:16, 262:13, 272:2, 280:14, 282:2, 282:9, 282:21, 296:11, 299:16
reads
65:9, 93:19, 138:14, 240:22, 254:8
real
3:5
realistic
124:11
realize
234:5
really
19:22, 23:2, 23:12, 58:5, 62:8, 76:14, 124:9, 194:20, 253:19, 261:4, 264:6, 264:7, 266:14, 270:16, 274:9, 296:4, 296:8, 299:14, 303:9
realtime
52:20
reanswer
113:14
reason
9:6, 71:7, 74:8, 128:13, 234:19, 292:17, 293:1, 305:18
reasonable
100:12, 102:8, 108:18
reasons
31:1, 31:21, 43:20, 78:7, 225:10, 225:18
rebit
146:1
receive
151:13
received
50:17, 85:6, 127:21, 133:21, 227:17
receives
46:13, 47:21, 148:7, 151:14, 151:20, 152:1, 159:3
receiving
82:15
recent
18:6, 173:5
recess
56:20, 61:11, 114:2, 182:5, 239:18, 284:21, 307:15
recipient
85:4
recite
109:19
recited
110:3, 112:14, 115:5, 117:15, 117:18, 194:4, 194:10, 195:4, 195:12, 195:16, 195:17, 197:11, 198:20, 221:14
recites
109:22, 121:8, 121:10, 143:12, 143:13, 143:21
recoded
269:12
recollect
14:11
recollection
10:18, 20:13, 26:13, 28:10, 53:1, 62:6, 69:22, 97:11, 99:13, 99:16, 108:4, 134:21, 136:1, 150:5, 158:9, 158:11, 158:17, 165:15, 186:2, 186:17, 201:16, 217:16, 227:3, 250:22, 257:16, 288:14, 289:6, 289:12, 289:18, 293:19, 306:10
reconstructed
154:3, 268:15, 268:17
reconstruction
75:10, 269:5, 269:8
record
5:12, 6:13, 7:14, 100:4, 168:4, 168:5, 182:7, 222:4, 273:10, 284:20, 285:1, 289:5, 289:11, 310:10
recorded
83:14, 85:11, 88:9, 145:14, 282:20
recording
89:13, 262:4, 282:4, 284:6
records
87:18
red
136:5, 142:5, 275:15
redesigned
294:4
refer
6:11, 11:14, 11:19, 13:11, 15:1, 15:7, 15:15, 16:7, 16:13, 25:11, 26:2, 27:15, 27:16, 27:20, 29:1, 29:17, 29:20, 30:17, 30:18, 30:19, 38:22, 125:5, 146:10, 147:7, 215:10, 220:8, 222:10, 228:16, 246:6, 255:6, 261:19, 291:7, 295:3
reference
29:12, 30:9, 83:7, 99:5, 99:6, 125:11, 125:14, 126:1, 126:4, 133:22, 147:19, 148:14, 149:12, 161:16, 161:21, 162:2, 162:6, 163:4, 163:9, 163:16, 164:1, 166:19, 170:1, 170:16, 171:6, 173:6, 175:9, 177:11, 178:20, 223:12, 236:20, 240:5, 255:6, 257:14, 258:13, 286:12, 288:7, 288:8, 288:11, 288:12, 288:18, 288:21, 289:16, 291:9
referenced
22:6, 23:10, 108:19, 109:7, 174:3, 192:8
references
220:7, 222:9, 291:22
referencing
207:7
referred
10:14, 30:2, 69:11, 116:15, 146:18
referring
69:20, 83:21, 101:20, 144:18, 166:4, 186:7, 228:22, 246:1, 246:2, 246:4, 263:11, 272:16,

287:20, 294:11, 306:11
refers 25:13, 38:4, 85:1, 102:1, 194:15, 214:16, 229:19, 276:3, 276:7, 276:19
reflect 24:21, 78:9, 102:6
refresh 292:9
regard 20:14, 24:22, 32:2
regarding 21:22, 22:10
region 129:21
relate 17:22, 28:16, 95:15, 257:17
related 10:7, 22:10, 79:16, 95:12, 240:6, 264:22, 310:13
relates 31:12, 74:8, 80:20, 90:4, 96:4, 224:17, 257:17
relating 10:6, 21:22, 35:6, 35:9, 60:19, 60:20, 94:14
relation 18:21, 19:1, 19:5, 20:16, 23:16
relatively 190:13, 191:4, 195:20
relevant 16:20, 18:3, 18:11, 28:1, 28:2, 32:4, 47:17, 49:22, 66:15, 153:19, 154:7, 224:2
relied 23:6
relies 31:11, 131:1, 203:10
reluctant 95:19, 95:21, 116:21, 120:21, 121:1
rely 92:9, 125:21, 236:20, 237:2, 237:6
relying 22:18
remember 17:4, 17:6, 17:11, 62:10, 68:9, 68:11, 74:17, 74:20, 83:8, 166:4, 190:16, 239:13, 276:10, 289:8, 299:7
remind 15:4
reminder 188:22
render 30:8, 31:2, 31:22, 194:19, 224:8, 248:3, 248:4
rendered 223:15, 224:18
renders 28:7, 28:20, 29:6, 29:10, 30:3, 221:7, 221:8, 223:6
reordered 164:9
repeat 177:7, 226:5
rephrase 132:22, 161:18, 233:11
replace 162:1, 163:9, 163:15, 163:17
report 50:14, 84:8, 145:21, 223:9, 241:21, 245:13, 266:3
reported 1:22
reporter 5:18, 310:4
represent 13:6, 55:2
representation 47:12, 55:7, 62:15, 62:21, 62:22, 63:9, 63:22, 64:10, 64:18, 65:2, 129:19, 129:20
representing 33:14, 140:12, 203:20, 203:22
reproduce 34:8, 186:10, 264:8, 268:4, 268:5, 268:7, 281:22, 282:5, 284:6, 284:14
reproduced 78:21, 261:13, 263:10, 263:11, 263:13, 264:5, 267:9, 280:6, 281:21, 282:20
reproducing 85:10, 89:13, 261:10, 261:15, 273:1, 273:14, 282:5, 284:6
reproduction 88:8, 127:20
request 266:18, 268:8, 269:14, 270:1, 270:12, 271:17, 290:1
requested 188:6, 263:22, 276:5, 281:5, 292:6, 292:12, 292:15
requesting 284:2
requests 192:13, 197:11, 276:18, 276:20, 277:17, 290:3, 292:19
require 93:14, 95:1, 183:2, 184:9, 184:12, 191:2, 191:6, 191:13, 192:2, 211:16, 213:15, 213:21, 258:13, 301:18
required 83:2, 84:10, 85:18, 88:12, 88:19, 90:4, 95:13, 112:3, 119:21, 121:17, 163:13, 184:3
requirement 49:2, 59:4, 71:4, 93:8, 109:15, 110:13, 113:6, 120:10, 181:6, 181:7, 181:9, 181:15, 181:22, 184:16, 240:22
requirements 49:11, 51:18, 52:14, 58:21, 94:17, 158:12, 158:14, 160:1, 160:2, 191:1, 230:10
requires 72:20, 94:8,

95:17, 95:18, 241:5, 241:9, 265:6, 265:10
requiring 246:11
rescale 78:5
rescaling 40:14, 42:21, 78:15
reset 123:9, 123:10, 194:6, 217:22, 249:19
resetting 16:12, 114:10, 114:15, 115:14, 115:17, 123:18, 255:3, 306:6, 306:7, 307:8
respect 12:13, 12:15, 31:10, 32:5, 83:7, 210:10, 300:5
responding 8:20
response 6:19, 7:1, 10:22, 11:10, 11:20, 113:12, 228:8
responsible 53:20, 54:1
rest 44:13, 220:15, 271:3, 298:12
restart 170:21, 288:10
restate 306:19
restore 261:20, 262:3, 280:5
restored 268:22, 270:10
restriction 58:5
restrictions 178:1, 179:19, 179:22, 180:10
result 48:6, 65:13, 112:2, 112:10, 178:3, 181:17
resulting 45:8
results 79:8, 157:22, 158:5, 158:19, 180:15
resumed 97:9
retained 14:3, 20:11, 90:19, 238:1
reversal 40:15
reverse 39:19, 191:4, 262:1, 262:2, 280:6, 280:7, 280:11
reverse-quantization 271:6, 271:13, 272:21
reverse-quantized 269:21, 270:15
reverse-transform 271:5
reverse-transformation 271:14, 272:22
reverse-transformed 269:12, 269:20, 270:14
review 17:3, 20:19, 21:2, 21:8
reviewed 12:18, 16:17, 16:18, 17:17, 21:18, 22:13, 202:5, 223:14
reviewing 17:7, 17:9, 17:10, 17:20, 18:2, 18:5, 18:12, 92:12
rewind 81:5
rewinding 80:11
richardson 1:15, 2:18, 5:6, 5:13, 56:22, 114:4, 182:9, 202:4, 239:20, 285:2, 308:5, 309:3
rights 251:12
rigorously 58:9
risk 144:20
roman 24:11
roughly 68:9, 111:11
round 248:2, 256:20, 283:11, 286:10
routes 275:5
rules 7:18

S

s 3:1
said 18:1, 18:15, 23:12, 27:8, 40:9, 44:16, 44:22, 45:13, 54:8, 54:12, 59:1, 59:11, 62:19, 63:8, 66:10, 69:15, 74:18, 79:19, 80:16, 96:19, 104:16, 113:12, 118:5, 120:17, 151:5, 163:7, 170:8, 170:12, 178:11, 181:21, 191:10, 202:5, 205:17, 205:18, 212:2, 220:9, 222:11, 229:11, 248:17, 262:20, 265:12, 265:16, 267:17, 273:5, 273:8, 279:13, 289:5, 289:8, 289:16, 289:17, 293:6, 294:13, 297:3, 298:15, 299:7, 299:16, 300:8, 301:12, 306:20, 307:6
sake 52:18, 155:16
same 14:15, 15:17, 16:9, 16:14, 18:22, 21:6, 28:22, 31:1, 33:13, 42:6, 48:6, 48:13, 49:4, 50:1, 50:6, 53:14, 65:12, 66:20, 67:2, 75:18, 77:22, 78:14, 79:8, 113:14, 127:20, 133:6, 134:19, 134:22, 135:16, 135:18, 135:22, 141:1, 148:5, 177:14, 198:12, 198:14, 212:18, 223:3, 235:16, 261:20, 275:18, 286:5, 286:17, 286:19, 287:1, 287:3, 287:11, 297:20, 299:22

sames
79:8
san
3:6
satisfied
221:5
satisfies
221:7
satisfy
116:11, 119:17, 119:18, 119:21, 121:3, 132:8, 178:22, 179:7, 307:5
satisfying
183:19, 306:14
save
100:3, 154:5
saying
50:3, 107:5, 108:13, 140:16, 140:17, 147:16, 156:18, 162:22, 166:18, 179:8, 188:3, 210:12, 216:19, 251:8, 283:14, 301:11
says
30:8, 47:1, 47:10, 49:10, 53:14, 56:15, 79:4, 101:17, 106:22, 107:9, 107:10, 107:13, 108:3, 110:19, 111:11, 111:12, 114:10, 116:2, 117:12, 118:20, 123:7, 123:18, 133:17, 153:3, 153:17, 153:22, 157:8, 158:7, 158:16, 160:5, 160:9, 160:13, 160:14, 161:19, 163:18, 164:7, 164:16, 164:20, 165:2, 166:2, 171:14, 173:16, 175:18, 178:16, 193:8, 193:10, 201:19, 205:5, 206:21, 209:18, 210:3, 210:16, 210:18, 211:5, 213:1, 213:21, 214:14, 217:20, 259:20, 261:22, 267:6, 268:3, 268:5, 273:4, 273:13, 273:22, 274:4, 278:3, 279:13, 280:7, 294:7, 297:13
scan
145:22
scanning
145:16, 145:17
scenario
55:11, 55:12, 57:16, 59:12, 73:7, 74:6, 74:11, 79:14, 86:7, 127:13, 127:19, 129:3, 129:12, 156:11, 261:1
scenarios
100:14
schematic
135:5
scheme
109:14, 116:4, 118:22
scope
47:10, 66:5, 66:14, 67:17, 68:7, 72:13, 73:5, 74:5, 75:15, 143:1, 143:8, 143:17, 195:8, 197:16, 199:13, 199:19, 201:11, 222:21, 239:12, 298:1, 299:6, 302:20, 303:8, 303:15, 304:8, 304:20
search
9:16, 69:4
searchable
9:15, 9:17, 188:21
secondly
305:11
section
24:10, 24:11, 44:4, 44:9, 44:12, 49:16, 58:16, 61:16, 61:18, 63:2, 63:4, 63:11, 75:3, 75:6, 91:20, 98:13, 98:14, 98:18, 99:15, 100:8, 101:5, 103:6, 146:10, 146:11, 146:19, 146:22, 147:7, 147:8, 147:11, 149:8, 150:8, 153:1, 161:9, 162:11, 165:8, 171:8, 191:1, 204:13, 213:20, 235:11, 235:14, 235:16
sections
91:22, 166:17, 203:6, 208:18
seeing
91:22, 92:13, 118:5, 157:12
seem
102:5, 114:22, 115:6, 119:20, 181:7, 183:18, 244:8
seemed
244:19
seems
95:9, 215:7, 259:21, 289:2
select
72:6, 81:1, 99:3, 142:7, 144:11
selected
262:16, 263:3, 263:9, 294:15, 302:16
selecting
142:11
selection
99:7
selector
302:7
selects
72:8, 79:22, 137:7, 263:16, 265:18
semantics
44:5, 45:3, 48:10, 53:13, 67:7, 147:13, 196:11, 196:20, 196:21, 202:1, 208:12, 212:6, 253:17, 305:16
sending
85:2, 86:1
sense
43:8, 270:4, 270:5, 271:2, 281:20, 287:7, 287:8
sent
57:7, 269:19, 271:4
sentence
50:15, 51:20, 53:1, 61:19, 75:5, 75:17, 80:12, 97:18, 97:19, 98:3, 98:21, 98:22, 99:14, 103:7, 103:16, 130:9, 144:22, 153:13, 161:13, 161:19, 164:7, 172:13, 173:8, 190:21, 190:22, 206:14,

206:19, 214:10, 222:19, 223:1, 252:15
sentences 108:2
separate 23:3, 119:14
sequence 14:9, 14:20, 15:12, 53:9, 55:16, 56:5, 95:5, 95:7, 101:10, 101:14, 101:19, 102:12, 102:15, 103:3, 105:14, 105:20, 106:2, 106:8, 106:13, 108:16, 108:21, 109:9, 109:21, 110:1, 110:5, 110:7, 110:9, 110:22, 112:16, 113:1, 114:12, 115:12, 115:19, 115:22, 121:9, 122:5, 122:7, 122:13, 122:15, 135:22, 142:9, 153:8, 164:19, 183:6, 183:8, 184:15, 188:7, 192:14, 193:14, 194:4, 194:10, 195:4, 197:5, 198:1, 198:11, 220:5, 220:8, 220:9, 222:7, 222:10, 222:11, 225:3, 225:5, 241:2, 242:5, 244:5, 247:15, 247:18, 253:15, 265:12, 265:13, 265:16, 265:17, 292:21, 294:22, 295:3, 295:11, 295:12, 295:21, 296:3, 296:17, 297:2, 297:6, 297:14, 297:19, 300:9, 300:11, 300:21, 301:2, 303:7, 303:20, 304:6, 304:18
sequences 142:16
series 112:4, 228:4
service 82:3, 82:7, 82:9, 84:1, 138:19
services 1:5, 2:5, 81:21
serving 237:22
set 49:1, 50:5, 57:14, 58:7, 59:10, 59:21, 70:3, 70:22, 71:8, 71:10, 71:15, 71:17, 72:3, 73:3, 105:7, 107:15, 165:22, 174:7, 174:15, 175:11, 176:13, 176:19, 177:8, 178:16, 180:17, 181:12, 206:22, 207:19, 209:21, 209:22, 210:18, 210:20, 211:1, 211:3, 211:6, 212:7, 212:18, 212:22, 214:6, 251:10, 259:5, 259:6, 259:16, 259:22, 260:1, 267:8, 310:8
sets 55:19, 56:4, 70:8, 105:12, 105:17, 107:1, 107:11, 107:14, 108:3, 306:5
setting 106:4, 106:9, 107:6, 107:20, 107:21, 108:5, 146:6
settle 274:12
seven 6:3, 6:5
several 46:22, 220:2
short 56:8, 154:1, 190:14, 191:5, 195:20, 224:3, 285:3, 299:9
shorthand 11:14, 14:16, 27:17, 29:1, 36:3, 36:5, 268:18, 310:4
should 9:19, 42:5, 43:8, 44:1, 44:6, 44:8, 48:5, 48:7, 48:12, 49:3, 49:22, 54:2, 61:7, 67:13, 68:14, 138:8, 150:6, 151:9, 152:5, 152:7, 158:16, 158:19, 160:6, 168:6, 170:3, 173:16, 188:20, 190:1, 192:22, 193:10, 193:13, 193:20, 200:4, 254:20, 254:21, 258:2
show 124:14, 293:2, 293:3
showing 124:14, 127:3, 141:16
shown 77:16, 77:17, 129:1, 169:14, 195:5, 200:17, 214:16, 219:1, 259:9, 278:17, 284:13
shows 124:21, 135:13, 154:20, 215:15, 257:1, 274:22
side 82:6, 256:13
sign 92:17
signal 37:4, 72:5, 270:12, 274:2
signature-b7fzp 310:18
silent 258:1
similar 7:19, 16:4, 32:5, 33:3, 75:19, 85:20, 215:9, 300:21
simple 139:2
simpler 47:4
simply 128:15, 161:3, 168:10, 186:5, 187:20, 216:8
simultaneous 85:2
simultaneously 86:1
since 5:15, 62:5, 99:2, 168:9, 258:12
single 111:5, 144:21, 147:14, 252:15, 261:3
single-bit 225:12

single-image
83:15
sit
73:8
sitting
5:16, 81:18, 88:17, 117:3, 117:13, 223:18, 224:10, 305:4
situation
131:13, 151:16, 151:17, 160:21, 173:19, 241:18, 245:12, 246:20, 259:8
situations
243:20, 244:1
size
53:15, 54:2, 55:16, 55:20, 55:21, 56:4, 56:5, 156:15, 156:20, 157:10
skill
25:7, 81:15, 82:20, 111:1, 162:18, 195:2, 198:8, 215:19, 216:22, 228:14, 229:3, 230:14, 232:9, 269:6, 270:19, 272:1, 274:7, 300:19, 301:1
skip
107:12
slight
41:21
slightly
48:2, 80:15, 175:2, 208:16
smith
3:14
so-called
291:12
software
39:21, 40:21, 46:1, 51:16, 52:7, 52:8, 52:10, 52:11, 52:19, 138:14, 145:2, 160:19, 161:5
solution
140:8
solve
188:17
solving
97:14
some
14:16, 18:6, 35:8, 37:3, 39:14, 39:17, 40:9, 55:4, 68:9, 71:8, 71:9, 76:14, 87:7, 87:16, 102:20, 128:3, 129:14, 133:8, 133:9, 136:16, 138:21, 140:19, 165:22, 171:16, 185:10, 190:6, 201:20, 203:4, 223:17, 243:13, 243:15, 244:21, 245:7, 298:14
somebody
199:17, 227:4
something
9:16, 44:7, 56:10, 64:20, 70:15, 83:2, 85:12, 95:11, 104:17, 108:6, 110:11, 113:3, 115:1, 115:8, 120:4, 127:18, 128:19, 128:22, 129:14, 132:12, 133:20, 138:11, 140:17, 146:22, 157:11, 162:20, 168:15, 171:10, 196:3, 209:7, 224:8, 228:11, 228:18, 229:8, 229:9, 229:12, 229:14, 229:22, 231:2, 252:21, 262:20, 272:3, 279:18, 289:13, 289:16, 296:14, 298:18, 298:19, 299:19, 299:22, 301:16
sometime
19:3
sometimes
33:12, 269:5
somewhere
159:14, 276:22, 295:19
sooner
7:21
sorry
19:1, 27:1, 29:13, 31:7, 41:5, 51:3, 58:10, 60:9, 70:19, 75:16, 80:8, 80:17, 81:11, 84:21, 86:11, 91:9, 91:13, 93:1, 96:5, 96:17, 101:16, 102:19, 106:2, 107:20, 109:12, 115:2, 122:16, 122:21, 124:18, 132:22, 151:2, 153:18, 153:19, 158:10, 161:18, 164:17, 170:21, 179:1, 179:5, 187:4, 193:21, 194:7, 199:12, 201:22, 206:12, 208:1, 209:21, 211:21, 222:22, 226:5, 233:11, 236:4, 238:11, 243:18, 249:2, 260:7, 265:6, 271:15, 273:7, 275:17, 276:11, 276:12, 277:6, 278:7, 288:9, 288:20, 290:2, 290:7, 292:2, 292:8, 297:1, 297:11, 298:21, 305:22, 306:18
sort
43:13, 43:18, 58:9, 59:2, 64:7, 84:18, 131:11, 146:5, 179:22, 226:22, 232:5, 245:7, 264:21
sounds
199:21, 299:8
source
41:4, 41:6, 62:13, 62:15, 62:21, 63:1, 78:19, 78:20, 154:2
space
75:11
spaced
83:15
spans
124:3
speak
205:21
specific
37:13, 41:14, 55:19, 56:5, 60:3, 60:9, 67:20, 68:17, 70:7, 110:10, 110:12, 110:15, 110:17, 113:2, 113:4, 113:7, 182:20, 183:3, 184:5, 184:7, 184:10, 261:12
specifically
19:4, 22:6,

29:1, 60:21, 162:22, 205:6, 206:2, 215:3, 231:3, 265:2, 268:9, 271:15
specification 44:15, 91:3, 91:19, 92:2, 92:20, 93:4, 149:9, 166:2, 212:5
specifications 39:4, 92:13
specificity 195:18
specified 42:1, 42:2, 46:8, 48:7, 48:10, 48:15, 49:5, 65:14, 71:12, 79:5, 128:19, 148:7, 149:1, 149:17, 169:3, 185:1, 187:22, 193:18, 289:20
specifies 44:2, 44:4, 44:10, 44:22, 45:1, 45:2, 45:4, 47:12, 47:14, 58:1, 67:6, 67:10, 68:13, 72:19, 132:4, 145:19, 147:13, 178:4, 180:16, 192:22, 201:12, 206:3, 252:1, 252:2
specify 28:3, 43:7, 43:22, 45:5, 45:14, 58:2, 132:16, 147:12, 196:4
specifying 45:3, 193:8
speculate 68:12
spell 284:11
spent 19:14, 19:15
spoke 19:21, 37:20
stacks 115:4
stage 77:15, 77:17, 270:6, 280:3, 280:4, 280:5
stamped 47:1, 47:9, 49:9, 53:10, 53:13, 55:1, 149:6, 193:7
stand 89:5, 190:7
standard's 175:21
standards 24:5, 24:9, 24:12, 27:12, 38:13, 38:15, 38:21, 39:3, 39:5, 50:3, 58:10, 64:3, 68:11, 68:13, 69:3, 70:3, 163:14, 193:8, 291:2, 305:2
stands 23:17
star 186:15, 186:16
start 39:9, 46:11, 63:15, 64:5, 70:19, 79:21, 80:1, 80:22, 81:1, 81:3, 98:22, 101:10, 137:19, 138:7, 138:13, 138:15, 139:9, 143:5, 143:9, 143:15, 144:12, 145:8, 145:20, 146:1, 146:2, 146:3, 146:4, 149:11, 162:10, 185:22, 186:18, 187:4, 187:7, 187:8, 187:12, 187:13, 187:21, 188:6, 188:7, 188:9, 189:13, 189:14, 189:15, 192:13, 192:19, 193:1, 193:2, 193:14, 193:15, 193:20, 194:1, 194:12, 195:7, 197:12, 204:15, 204:20, 227:1, 227:6, 227:8, 242:6, 242:7, 242:20, 242:21, 242:22, 243:3, 243:6, 245:1, 245:6, 245:9, 245:10, 246:5, 253:22, 265:22, 266:1, 296:14, 296:16, 296:17, 298:11, 298:17, 298:19, 299:19, 299:22, 300:8
start_code 167:15
started 5:9, 138:1, 243:3, 243:5, 244:5, 294:20
starting 15:11, 15:19, 44:4, 84:11, 90:1, 98:8, 124:2, 146:7, 146:11, 146:19, 147:3, 147:4, 203:1, 225:7, 247:15, 264:18, 265:8, 265:11, 265:15, 265:22, 268:21, 273:11, 277:1, 295:2, 295:7, 295:8, 295:10, 295:20, 295:22, 296:3, 297:2, 297:5, 297:6, 297:13, 297:19, 297:20, 299:2, 299:3, 300:1, 300:10, 300:20, 301:2, 301:3, 301:5
starts 13:20, 14:6, 14:18, 14:19, 16:6, 16:11, 24:14, 39:8, 61:19, 101:6, 137:12, 140:1, 161:10, 192:12, 197:4, 198:13, 200:10, 215:22, 226:20, 241:14, 246:18, 246:20, 266:1, 267:18, 295:4
state 5:2, 5:11, 12:3, 21:16, 25:15, 31:5, 31:8, 46:19, 50:15, 51:21, 69:19, 80:19, 95:18, 97:19, 98:17, 104:22, 125:10, 125:22, 130:2, 137:6, 142:13, 148:5, 157:16, 159:3, 167:11, 216:3, 216:20, 229:18, 235:1, 244:20, 245:14, 250:13, 255:21, 262:16, 266:7, 268:12, 295:20, 296:10, 299:15, 310:5
stated 23:5, 28:22,

113:5, 172:14, 204:8, 208:20
statement 32:5, 63:19, 65:7, 172:19, 176:1, 178:7, 216:21, 223:20, 224:12, 244:8, 245:21, 274:8
statements 114:20, 114:21, 175:1
states 1:1, 2:1, 75:6, 118:11, 152:20, 158:13, 161:13, 162:11, 174:18, 176:12, 204:6, 204:14, 206:14, 208:5, 208:19, 213:8, 214:5, 222:3, 241:9, 257:13, 258:12, 271:15
stating 107:6, 112:19, 120:7, 144:20, 162:5, 162:9, 204:11, 204:19
status 20:7
step 40:14, 40:16, 50:4, 261:20, 261:21, 262:7, 262:9, 262:16, 262:21, 262:22, 263:1, 263:10, 263:15, 264:5, 264:6, 264:8, 264:20, 265:21, 266:2, 266:17, 268:19, 268:21, 269:11, 269:19, 270:8, 270:9, 270:15, 271:11, 271:14, 271:20, 274:16, 274:18, 277:12, 277:17, 277:18, 277:21, 278:10, 278:12, 279:6, 280:22, 281:2, 281:6, 281:19, 281:22, 282:10, 282:15
stepping 70:10
steve 284:16
steven 3:18
stick 252:13, 252:14
still 9:3, 60:6, 141:16, 141:19, 196:22, 217:15, 232:12
stops 267:7
storage 33:5, 33:17, 35:5, 55:3, 55:4
store 159:14, 163:4, 254:11, 254:20, 254:21, 256:20
stored 89:14, 133:22, 160:6, 160:10, 160:15, 160:16, 161:17, 162:1, 162:2, 162:7, 163:8, 163:9, 163:16, 164:2, 164:5, 171:15, 255:12, 256:1, 256:6, 257:15, 268:10, 269:13, 269:21, 276:18, 278:17, 281:5, 283:3, 283:6
stores 255:16, 255:22
storing 256:12, 256:16
straight 17:15, 26:5
strange 270:22
stream 80:10, 81:5, 116:1, 190:4, 245:1, 245:8
streamed 82:13
streaming 80:7, 81:10, 81:12, 81:14, 81:19, 82:2, 82:21, 83:3, 83:22, 84:7, 84:18, 84:19, 85:1, 85:7, 85:15, 85:20, 86:14, 86:15, 86:19, 87:10, 88:1, 88:10, 88:18, 88:22, 89:16, 89:18, 90:16, 94:4, 94:6, 94:9, 94:15, 94:22, 95:2, 95:15, 95:17, 95:19, 138:19
streams 82:4, 82:15
street 4:5
strictly 232:14
strike 122:21, 160:12, 194:7, 255:4, 260:18
structure 59:3, 69:16, 69:17, 227:1, 227:11, 227:21, 227:22, 228:1, 228:2, 253:9, 253:11, 257:1, 282:19, 283:9, 286:15, 293:2
structures 167:2, 169:13, 169:18, 227:9, 227:12, 228:4, 228:5
struggle 293:11, 295:15
struggled 257:13
struggling 55:22, 156:11, 287:12
student 148:12
sub 105:14, 105:20
sub-phrase 220:14, 220:18
subjective 35:12
submitted 10:13, 10:18, 10:20, 29:21
subscribed 308:8
subsection 222:2
subsequent 161:16, 161:22, 163:4, 189:22
substance 8:5, 57:2, 114:6, 182:12, 239:22, 285:4
successfully 46:7, 126:17, 128:12, 129:16, 153:4, 173:22, 186:5, 187:14, 187:21
sudick@mckoolsmi-th 3:17
sufficient 294:9
sufficiently 222:5

suggest
31:13, 225:2, 279:10, 279:11, 293:20, 294:3, 294:10
suggested
296:7
suggesting
214:11, 214:12
suggestion
289:3
suggests
192:22
suitable
33:4, 33:16
suite
3:5, 3:15, 4:5
summarize
34:5, 53:2, 62:11, 64:8, 76:13, 97:15, 256:14
summarizing
256:4
summary
51:9, 51:10, 87:20, 256:9
support
231:21, 233:6, 238:8
sure
5:17, 14:15, 23:11, 49:1, 52:16, 56:19, 77:3, 92:15, 95:10, 96:21, 98:20, 104:22, 116:9, 120:14, 129:10, 131:19, 170:4, 176:11, 182:4, 221:19, 253:2, 273:9, 288:18, 291:21, 303:2
surname
29:15, 30:3
sworn
5:2, 308:8, 310:8
synonym
153:14
synonymous
229:10, 229:14, 230:9, 298:7, 301:4
system
64:14, 87:14, 87:17, 87:21, 89:12, 145:15, 146:6, 255:15, 255:22, 282:12, 282:13
systems
27:21, 61:20, 62:4, 235:4

T

table
233:21, 234:3, 234:15, 234:17, 234:20
take
7:20, 8:2, 28:17, 39:6, 53:7, 54:12, 56:17, 61:9, 116:22, 148:11, 176:9, 182:2, 187:5, 195:3, 196:14, 220:14, 224:2, 228:9, 230:3, 239:16, 257:19, 257:22, 258:16, 258:17, 259:2, 259:17, 281:3, 284:17, 298:12, 300:15, 301:15, 305:9, 305:20, 305:21, 307:13
taken
2:18, 56:20, 61:11, 114:2, 182:5, 233:3, 239:18, 282:11, 284:21, 307:15
takes
54:16
taking
40:12, 40:20, 66:12, 300:3, 300:12
talk
8:6, 132:6, 140:8, 145:20, 151:17, 160:7, 172:21, 176:6, 180:14, 182:11, 185:12, 190:17, 202:16, 202:20, 219:22, 237:11, 237:13, 249:12, 249:15, 263:20, 281:14, 296:22
talked
42:13, 55:15, 95:8, 102:21, 114:17, 118:9, 174:6, 174:19, 184:18, 189:17, 195:19, 233:18, 237:12, 241:19, 254:14, 255:5, 264:4, 285:14, 297:12
talking
26:16, 76:17, 76:19, 86:6, 89:16, 102:3, 112:12, 148:2, 182:17, 206:17, 207:11, 231:1, 231:3, 232:12, 232:13, 235:16, 250:18, 251:1, 261:18, 268:9, 270:3, 285:7, 297:11
talks
29:12, 46:21, 103:8, 106:15, 169:20, 171:13, 188:13, 190:5, 190:13, 203:2, 261:14, 296:22
task
265:4
teach
31:13, 97:1, 211:13, 211:19, 225:2, 248:4, 255:3, 294:10
teaches
97:13, 206:7, 208:6, 211:8, 211:22, 220:5, 220:18, 222:6, 223:6, 224:19, 245:17, 245:22, 306:4, 306:6, 307:7
teaching
294:19
teachings
248:9
technical
14:3, 17:21, 93:17, 94:16, 114:19, 239:3
technically
18:8, 51:10, 95:12, 171:7, 171:22, 177:20, 242:19, 253:1
technologies
1:11, 2:13, 21:22, 22:10
technology
82:10, 245:18, 245:22
tell
215:5
tells
189:12, 280:17
temporal
125:6, 148:13, 149:12, 220:6, 222:8
temporal_reference
148:15, 148:18, 148:20, 148:22,

149:2, 149:15, 149:21, 150:10, 150:12, 151:4, 151:10, 151:13, 151:21, 152:6, 152:9, 152:13, 153:10, 154:10, 154:11, 154:16, 155:1, 155:6, 155:15, 155:20, 155:21, 156:2, 156:7, 156:9, 159:5, 159:10, 161:1, 164:5, 164:14, 164:15, 164:18, 164:22, 165:5, 165:6, 165:10, 200:11, 200:12, 201:13, 201:21, 202:2, 215:5, 215:11, 215:21, 216:6, 216:13, 216:15, 216:17, 217:4, 217:10, 217:22, 218:3, 218:7, 218:13, 218:21, 219:12, 227:15, 249:19, 250:3, 250:6, 250:13, 250:14, 251:9, 251:10, 251:14, 252:4, 252:12, 253:6, 253:11, 253:21, 254:2, 254:6, 254:9, 254:10, 287:21, 288:1, 288:2
ten
156:3
tend
86:14
tens
145:2
term
26:7, 57:18, 81:14, 84:6, 85:1, 85:14, 85:19, 92:14, 92:17, 107:15, 108:15, 110:18, 118:2, 118:4, 118:15, 118:17, 118:22, 119:2, 119:4, 119:5, 131:12, 153:12, 191:13, 213:19, 231:22, 232:9, 238:13, 288:18, 291:1, 291:3, 299:1, 299:3, 300:10, 300:20, 302:17
terminology
174:13, 175:3, 208:16, 208:18
terms
33:3, 34:13, 78:16, 94:17, 123:19, 130:22, 165:16, 178:12, 238:3, 238:6, 238:12, 239:7, 302:13, 302:15
testified
5:3, 74:11, 90:18, 113:9
testify
9:7
testifying
239:11
testimony
8:5, 57:2, 74:16, 114:6, 182:12, 239:22, 285:4, 310:10
testings
27:22
text
63:21, 94:6, 123:22, 188:21, 216:9
thank
5:14, 7:5, 10:12, 12:5, 25:2, 25:22, 27:9, 29:3, 34:3, 57:4, 96:1, 105:21, 114:8, 122:18, 182:10, 182:15, 188:19, 188:22, 214:4, 218:22, 234:10, 240:2, 244:10, 257:11, 258:10, 262:11, 273:10, 277:7, 289:21, 307:11, 307:20
thanks
83:10, 154:5, 306:22
that'd
98:20
themselves
141:9
theory
43:11, 46:3
thereafter
167:15
therefore
210:15
therein
21:19
thing
59:6, 73:20, 82:19, 103:12, 129:5, 134:22, 171:22, 172:2, 228:20, 229:6, 229:15, 230:4, 230:20, 231:10, 231:17, 231:19
things
23:19, 24:20, 44:3, 79:19, 115:5, 132:9, 145:19, 163:2, 163:3, 195:21, 223:13, 276:15, 280:20, 305:19, 306:19
thinking
41:10, 41:14, 41:17, 41:20, 89:2, 117:1, 133:19, 145:12, 258:19
third
190:22
thought
63:13, 63:15, 116:10, 120:8, 135:19, 267:22, 273:7
thousands
145:1, 145:2, 145:3
three
6:16, 17:13, 19:1, 26:9, 62:6, 98:4, 100:14, 136:6, 142:17, 143:7, 143:16, 144:6, 169:20, 170:9, 236:19, 238:7, 239:9, 240:14, 240:16, 240:18, 243:11, 243:13, 248:5, 290:9, 300:3
through
24:13, 26:10, 80:10, 81:5, 82:13, 95:8, 100:19, 120:8, 124:6, 126:4, 134:7, 134:11, 141:10, 141:12, 220:22, 221:2, 249:10, 249:12, 267:7, 267:19, 269:16, 270:21, 274:18, 275:6
throughout
103:10, 203:6, 249:16
ticker
224:4, 224:5
time
9:2, 15:3,

19:14, 22:14, 28:17, 55:6, 63:6, 63:13, 63:16, 64:4, 77:4, 81:20, 83:16, 90:7, 90:22, 92:10, 113:21, 116:22, 166:3, 190:11, 213:3, 245:9, 261:20, 282:10, 291:19, 298:12, 301:15, 301:18, 305:9, 305:20
times
18:20, 109:2, 260:20, 261:3, 297:13
title
10:9, 27:6, 62:7, 147:5, 166:2, 204:13
tkendrick@perkinscoie
3:7
today
5:20, 9:7, 81:18, 88:17, 95:22, 108:18, 223:18, 224:10, 299:8, 305:4
today's
16:16, 17:19
together
196:1, 300:8
tolerances
49:4, 50:2
took
57:5
tools
190:6, 190:12
top
15:9, 49:10, 69:15, 126:22, 171:14, 185:6, 259:10, 285:15
total
6:3
touched
49:9
tough
17:14
towards
54:22
trademark
1:1, 2:1
transcoder
36:22, 37:4
transcoders
36:22
transcript
7:4, 289:10, 310:9
transform
42:1, 42:7, 42:9, 75:9, 78:3, 78:4, 78:12, 78:13, 78:15
transformation
262:2
transforming
37:8
transmission
33:5, 33:17, 35:5, 280:7
transmitted
85:5, 266:18, 271:18
treated
44:8
trial
1:2, 2:2
tried
129:15, 188:6, 189:9
tries
187:10
true
116:12, 151:5, 208:14, 243:13, 310:10
truthfully
9:7
try
7:20, 8:9, 11:14, 11:18, 15:22, 28:3, 113:14, 192:17, 253:3, 299:11
trying
44:21, 53:2, 62:11, 64:7, 76:13, 80:6, 117:2, 125:4, 130:17, 133:3, 176:2, 185:16, 188:11, 191:10, 264:16, 300:5, 300:15
tsui
4:8
tsui@alston
4:7
tuesday
19:13
turn
21:12, 69:7, 75:3, 89:21, 93:6, 98:7, 100:17, 101:1, 124:1, 130:12, 157:13, 172:10, 219:16, 221:16, 221:21, 246:15, 302:22, 305:22
turning
96:2, 210:10, 210:11, 243:11
turns
9:17
twice
275:19
two
6:2, 10:5, 10:13, 10:18, 19:6, 19:16, 32:10, 33:11, 34:6, 34:12, 34:14, 34:19, 35:3, 35:10, 35:16, 41:2, 60:13, 60:17, 69:2, 73:17, 77:20, 81:4, 100:14, 119:14, 142:8, 146:9, 160:3, 205:19, 208:18, 219:15, 225:10, 225:18, 230:9, 230:10, 233:2, 239:14, 240:10, 255:9, 274:13, 274:22, 275:4, 286:19, 290:9, 298:6, 301:17
two-minute
61:10
tx
3:16
tyler
3:10, 5:9
type
37:17, 69:18, 70:8, 71:1, 71:15, 71:16, 71:18, 72:9, 73:2, 74:2, 74:13, 129:11, 129:12, 136:12, 139:3, 149:9, 170:13, 170:19, 171:11, 171:18, 233:21, 234:3, 234:15, 303:18
types
37:13, 38:9, 87:3, 169:20, 170:9, 188:16, 255:9, 304:4
typical
64:2, 64:15, 76:15, 76:20, 77:6, 83:17, 169:4, 169:6, 169:10, 169:11, 169:12, 169:17, 172:6, 172:8, 214:15, 214:20
typically
82:11, 84:3,

85:3, 86:3, 86:7, 145:1, 189:4, 189:6
typo 167:21, 289:9
typographical 167:21

U

uh-huh 8:15
uh-uh 8:15
um 17:7, 134:21
um-hmm 101:16, 226:18
unable 199:1, 216:3, 216:14
unavailable 132:18
unclear 218:10
uncompressed 33:2, 33:7, 33:15, 35:13
under 5:19, 47:9, 75:5, 82:6, 133:2, 140:20, 141:15, 193:3, 194:5, 194:11, 197:13, 200:4, 200:18, 201:9, 204:12, 215:20, 216:4, 222:2, 233:20, 240:4, 241:11
underlying 82:10
underneath 27:7, 215:4
understand 5:19, 8:10, 12:9, 12:11, 12:22, 19:16, 25:12, 25:18, 25:19, 27:5, 28:11, 49:19, 50:12, 76:11, 82:21, 86:13, 88:1, 88:6, 91:2, 93:22, 110:2, 111:2, 112:21, 154:12, 156:8, 156:12, 175:4, 183:17, 188:2, 194:20, 198:5, 202:9, 202:13, 204:5, 204:8, 215:20, 221:6, 228:14, 229:3, 229:17, 230:14, 231:20, 231:22, 246:8, 246:11, 247:14, 252:18, 253:18, 257:21, 258:16, 259:3, 261:4, 264:16, 269:7, 270:3, 274:3, 274:13, 286:20, 288:4, 290:20, 293:17, 303:9, 304:10
understanding 9:14, 11:2, 11:8, 11:11, 20:1, 26:8, 26:12, 29:8, 29:13, 31:6, 49:2, 83:13, 88:2, 90:5, 90:7, 91:17, 93:19, 119:15, 150:4, 163:7, 193:17, 220:21, 232:8, 238:18, 282:9, 303:17
understood 8:22, 9:5, 11:4, 11:13, 17:14, 23:21, 35:15, 59:20, 68:4, 73:21, 113:16, 113:20, 124:12, 135:6, 154:9, 218:9, 244:10
unfair 300:16
unfamiliar 64:5
unintentional 104:3, 186:4, 187:19, 188:14, 189:6
unique 285:17, 285:20, 286:1
unit 69:12, 69:15, 69:18, 69:20, 71:1, 71:3, 71:15, 71:16, 71:18, 72:9, 73:2, 74:1, 74:13, 233:16, 233:17, 233:21, 234:3, 234:15, 236:2, 237:3, 282:5, 284:6, 303:3, 303:5, 303:10, 303:11, 303:16, 303:18, 304:4
unit_type 69:22
united 1:1, 2:1
units 70:8
unless 9:2, 28:1, 36:7, 60:9, 94:9, 272:2, 298:11
unlike 216:12
unlikely 126:17, 127:19
unnecessary 280:21
unpack 16:22, 19:3
unsigned 161:4
unsure 139:8, 139:13
until 56:21, 61:12, 114:3, 182:6, 188:8, 193:13, 239:19, 242:5, 259:6, 277:12, 284:22, 307:16
unusual 171:22
use 26:6, 28:22, 36:6, 56:13, 79:13, 84:6, 85:14, 86:14, 93:15, 125:20, 126:13, 163:5, 165:3, 171:2, 175:2, 175:21, 178:4, 186:11, 190:13, 191:4, 191:18, 196:14, 205:9, 208:2, 231:12, 258:22, 259:1, 260:5, 260:11, 284:13
user 52:18, 79:22, 80:6, 81:3, 82:6, 82:15, 82:17, 83:21, 83:22, 84:2, 128:1, 128:22, 137:7, 138:13, 140:10, 142:7, 142:11, 144:11, 184:22, 187:9, 187:22, 188:6, 192:13, 197:11, 198:13, 244:22, 245:5, 245:6, 263:2, 263:3, 263:9, 263:16,

265:18, 276:5, 276:17, 276:20, 277:17, 290:1, 290:3, 292:6, 292:12, 292:16, 292:19, 292:20, 302:6, 302:16
user's
64:20, 266:18, 268:8, 269:14, 270:1, 270:11, 271:17, 284:2
users
262:15
uses
85:22, 94:9, 195:12, 208:16, 288:18
using
79:7, 84:1, 85:19, 125:15, 125:18, 166:10, 166:14, 172:7, 172:8, 172:15, 172:21, 173:4, 173:10, 173:17, 174:1, 174:20, 177:17, 178:13, 180:7, 180:22, 210:6, 211:14, 213:9, 213:18, 258:9, 260:17, 269:1, 284:2, 302:5, 302:7
usual
172:2
usually
138:20

V

vague
131:12
validity
12:7, 13:1, 121:5, 202:6
values
16:7, 46:15, 50:19, 51:15, 53:6, 54:11, 65:11, 65:18, 66:4, 66:18, 66:19, 66:21, 66:22, 71:10, 109:13, 109:16, 110:18, 111:8, 111:13, 112:15, 116:3, 116:5, 116:14, 117:5, 117:10, 117:12, 117:15, 118:3, 118:12, 118:20, 119:1, 119:9, 119:11, 120:3, 121:11, 122:3, 122:10, 122:13, 122:21, 123:1, 123:5, 123:17, 215:21, 216:6, 258:17, 307:4, 307:6, 307:8
variable
40:6, 40:8, 161:7, 258:21
variety
44:18
various
55:2
vector
273:3, 273:17, 274:6
vectors
174:16, 176:15, 178:18, 179:10, 180:3, 181:4, 207:22, 208:8, 209:1, 209:16, 210:1, 210:6, 210:14, 210:21, 212:11, 212:15, 212:20, 269:2
versa
39:20
verses
298:20
version
9:19, 236:8, 237:2, 237:6, 237:9, 237:17, 237:19, 291:21
versions
7:15
vice
39:20
videos
82:8, 82:12
view
82:8, 85:8, 85:9, 184:11, 191:16, 191:20, 205:19, 231:5, 231:20, 265:14, 296:12, 298:16, 299:17
viewing
80:1, 81:1, 81:3
visual
139:6
voice
77:10

W

want
13:15, 13:19, 15:4, 23:18, 49:16, 56:17, 59:6, 59:14, 59:16, 63:10, 69:7, 70:12, 95:16, 98:19, 104:20, 109:3, 113:14, 154:5, 174:11, 195:10, 197:1, 202:17, 202:20, 221:20, 236:13, 239:15, 257:9, 273:9, 275:11, 275:17, 276:13, 284:16, 289:10, 298:12, 301:15, 302:22, 305:22
wanted
49:14
watching
138:19
way
10:10, 46:4, 46:5, 59:3, 76:9, 76:11, 85:21, 86:14, 86:16, 112:1, 128:10, 129:18, 131:1, 133:15, 133:16, 158:8, 162:15, 162:17, 163:1, 163:3, 163:13, 163:18, 163:21, 168:20, 172:6, 172:8, 175:19, 175:21, 175:22, 177:18, 178:1, 178:5, 180:9, 180:13, 180:16, 181:22, 183:9, 191:10, 191:12, 206:14, 208:12, 212:4, 215:22, 216:3, 221:8, 229:1, 230:7, 231:19, 245:17, 245:22, 248:2, 252:14, 257:3, 267:7, 267:19, 270:18, 271:19, 278:10, 296:6, 305:1, 310:15
ways
43:12, 44:18, 141:7, 180:1, 191:21
we'll
7:20, 11:14, 14:15, 16:22, 29:20, 221:21
we're
5:15, 5:17, 14:15, 33:12, 36:7, 82:1, 108:17, 112:12, 163:12, 250:18,

271:20, 293:16, 297:11, 307:19
we've
56:16, 123:14, 153:11, 160:1, 160:2, 164:16, 180:1, 277:16, 278:4, 284:17, 290:22
weather
8:6
web
61:2
week
19:4, 19:11, 19:17
weird
56:10
welcome
56:22, 114:4, 182:8, 234:11, 239:20, 262:12, 285:2
well-known
84:4, 169:17
west
4:5
whatever
109:17, 118:8, 138:13, 196:2, 299:12
whatsoever
260:11
whenever
36:6
whereas
211:16, 213:21
whereupon
56:20, 61:11, 114:2, 182:5, 239:18, 284:21, 307:15, 307:22
whether
13:4, 18:8, 52:12, 57:6, 69:5, 71:2, 94:8, 95:16, 113:2, 113:3, 116:11, 116:14, 117:4, 117:14, 119:4, 119:20, 119:21, 120:8, 120:9, 121:12, 121:17, 122:2, 123:15, 131:21, 135:20, 139:10, 152:11, 166:7, 180:21, 187:16, 198:9, 213:4, 215:6, 216:4, 216:14, 219:18, 220:17, 221:4, 223:5, 223:19, 224:1, 224:11, 224:19, 225:13, 232:16, 235:20, 247:4, 254:19, 265:19, 297:17, 298:6
whilst
52:1, 52:6
whole
41:9, 90:9, 103:12, 111:22, 140:7, 141:7, 171:2, 241:21
widely
69:2
wider
34:2
width
53:16
wish
62:18, 147:9
within
42:2, 42:6, 49:4, 50:2, 68:14, 94:13, 108:4, 119:7, 133:13, 149:10, 173:13, 175:15, 177:2, 177:14, 179:15, 185:21, 186:1, 220:8, 222:11, 242:9
without
13:3, 60:3, 89:1, 116:11, 128:17, 131:18, 132:18, 133:4, 139:6, 166:10, 166:14, 169:22, 170:15, 171:5, 173:10, 174:1, 174:16, 174:20, 175:9, 176:15, 176:21, 177:11, 178:13, 178:18, 179:10, 180:3, 180:6, 180:22, 181:4, 198:16, 199:2, 207:21, 208:8, 209:1, 209:15, 210:1, 210:6, 210:13, 210:21, 211:14, 212:10, 212:14, 212:19, 213:9, 255:10, 255:16
witness
92:6, 239:11, 307:12, 309:3, 310:7, 310:11
witnesses
304:20
wonder
113:21
word
75:16, 77:1, 77:9, 78:9, 82:14, 105:16, 165:3, 186:11, 191:18, 212:8, 224:6, 229:4, 230:15, 231:12
words
14:11, 44:20, 47:3, 90:10, 93:10, 93:15, 94:3, 94:5, 94:9, 96:15, 117:20, 118:6, 118:7, 123:21, 147:16, 162:13, 177:17, 177:18, 185:3, 192:4, 196:3, 205:9, 252:13, 252:19, 300:4, 300:8, 300:12, 300:16
work
20:15, 22:22, 35:8, 60:12, 71:5, 144:12, 144:18, 145:8, 184:2, 196:20, 196:21, 253:3
worked
23:13, 67:5, 73:14, 74:19, 91:1, 270:21
working
17:13, 19:12, 19:13, 132:15, 160:19
works
148:12, 196:19, 284:19, 288:15, 289:6, 289:19
wouldn't
41:18, 68:12, 172:7, 191:18, 235:13, 245:11, 275:1, 278:1, 295:10
write
51:16, 52:8, 52:10, 87:2, 138:14, 168:2, 168:12
writing
52:19, 63:18, 76:3, 271:1
written
10:5, 22:14, 73:15, 73:17, 249:1, 249:2, 249:3, 249:7
wrong
234:5, 248:2
wrote
10:5, 10:11, 11:3, 11:5,

51:9, 60:13, 60:14, 62:5, 75:20, 76:10, 168:9, 237:14, 252:14, 262:22, 263:5, 271:10

Y

y'all
19:20
yagasaki
30:3, 30:8, 30:10, 30:16, 30:18, 30:19, 30:20, 32:6
years
22:13, 22:18, 23:14, 32:16, 35:3, 46:1, 62:5, 63:17, 67:5, 73:16, 75:20, 76:3, 90:21, 91:2, 92:1, 236:19, 238:7, 239:9
yep
134:5
yesterday
7:19, 19:18
york
1:17, 2:20, 5:3, 310:5
yourself
23:22, 32:11, 100:5, 101:2
youtube
84:1

Z

zero
280:1
zoom
18:21, 19:5

.

.2
44:4, 204:13
.20
154:20
.264
9:20, 10:6, 10:8, 10:9, 60:12, 60:13, 60:15, 60:19, 60:20, 60:21, 61:3, 66:2, 66:13, 66:15, 66:17, 67:3, 67:6, 67:12, 67:14, 68:10, 68:21, 69:2, 69:5, 69:11, 69:20, 70:1, 70:3, 70:12, 70:14, 70:21, 71:4, 71:16, 71:22, 72:3, 72:16, 73:6, 73:11, 73:15, 74:19, 75:4, 75:7, 76:4, 76:15, 76:19, 76:21, 76:22, 77:22, 79:1, 201:9, 201:12, 201:15, 202:1, 233:8, 233:14, 234:4, 234:6, 235:9, 235:10, 236:2, 236:20, 236:22, 237:7, 237:9, 237:15, 237:20, 303:5, 303:10, 303:21, 304:3, 304:15, 305:2, 305:14, 305:18
.3
44:9, 153:1
.4
44:9, 44:12
.5
153:1, 204:13

0

0
105:7, 107:13, 126:22
0.4
162:11
0.5
161:9
000
280:15, 306:13
001
285:19, 286:7
002
285:19
00691
1:9, 2:9
0321
6:7
05
2:21
09:10 am
56:21
09:22 am
56:21

1

1,023
155:5, 155:18
1.1
47:9
10
13:1, 13:12, 28:7, 28:13, 28:19, 28:20, 29:6, 29:11, 68:22, 143:13, 161:4, 221:17, 265:7
100
235:18
1001
6:6, 13:16, 84:13
1003
6:7
1005
6:9, 26:17, 26:21, 27:4, 47:1, 53:10, 149:7, 196:7
1006
6:12, 29:13, 29:14, 29:21, 236:4, 255:5
101
285:19
1015
30:10, 30:14
102
228:12, 232:4, 232:7, 233:1, 238:17
103
228:22, 230:8, 232:4, 232:7, 233:1
105
235:6
106
69:8, 69:10, 69:16, 69:19, 233:18, 233:20, 234:18, 236:5
10:35 am
114:3
10:45 am
114:3
11
13:2, 13:12, 26:10, 28:7, 28:13, 28:21, 29:6, 29:11, 91:13, 107:10, 108:12, 188:18, 189:3, 265:7
111
232:11, 262:1, 266:19, 269:13, 270:15, 271:18, 272:7, 272:19, 273:17
11172
6:10, 26:22, 27:6, 27:12, 29:2, 47:8, 47:11, 49:13, 158:15
112
240:3, 240:5
113
240:12

114
242:2
11452
3:5
119
241:3
12
53:12, 55:1
120
193:7, 241:15, 241:19
1200
3:15
1201
4:5
121
243:12, 243:22, 272:12, 273:1, 273:14
122
32:3, 232:21
123
32:3
125
245:13
129
246:18, 249:9
12:08 pm
182:6
12:47 pm
182:6
13
13:20, 92:16
130
195:22
1301
2:19
131
272:6, 273:18
133
246:22, 247:1
137
249:10
14
62:5, 63:17, 124:3
140
284:6
141
250:12, 252:5, 252:15
15
47:1, 62:5, 63:17, 75:20, 76:3, 86:7, 86:18, 124:3, 162:10, 172:11, 282:3, 284:4, 291:11
150
284:5, 284:13
151
276:13
154
265:1
16
161:8, 203:2
166
265:8
17
21:12, 23:5, 23:10, 261:13
171
292:18, 293:7, 294:13
172
265:14
174
307:1
176
302:3, 302:5
178
302:1, 302:11
18
24:3, 47:9, 214:17, 216:6, 216:8, 216:10, 272:20, 273:5
180
306:8, 307:7
184
306:16
186
31:5
19
84:21, 146:11, 146:16, 185:19, 186:9, 187:6, 188:5, 195:6, 199:5, 200:3, 215:15, 218:8, 219:2, 219:5, 219:14, 225:20, 226:2, 247:5, 247:11, 247:21, 248:5, 248:15, 248:20, 249:11, 249:14, 287:14, 288:3, 289:14
1998
291:18

2

2
6:10, 27:6, 27:12, 29:2
2.4
44:4, 44:9, 44:12, 171:8
2.7
171:8
20
50:14, 189:4, 273:12
200
155:5
2001
6:17, 10:22, 11:15, 91:16
2002
236:11
2003
10:11, 60:15, 237:1, 237:14
2004
6:21, 11:6, 11:18, 21:13
2005
7:3, 236:7, 237:1, 237:4, 246:1
2006
234:13, 236:5, 236:6
2010
10:5, 60:18, 61:3, 61:14, 77:14
2024
1:9, 2:9
2025
1:18, 2:21
21
168:5, 221:21, 222:3
22
262:9, 262:16, 262:21, 263:1, 263:10, 264:5, 275:3, 281:22, 282:15
23
13:21, 14:18, 90:1, 236:11, 275:2
2351
6:12
24
15:10
25
153:15
26
185:6, 185:17, 186:10, 201:17, 288:21
27
1:18, 2:21
29
146:14, 146:16, 271:16
2:02 pm
239:19
2:16 pm
239:19

3

3.1
61:16, 61:18
30
22:13, 22:18, 23:13, 32:16, 83:18, 84:4,

86:7, 261:20, 261:21, 262:7, 263:15, 264:6, 264:20, 266:2, 266:17, 270:8, 271:11, 271:14, 277:18, 277:21, 280:4, 280:5, 280:9, 281:2, 281:6, 281:19
300
3:5, 3:15
30309
4:6
31
271:16
32
261:18, 274:16, 274:18
33
172:10, 190:20, 259:10
34
277:12, 278:6, 278:10, 278:12, 279:6, 279:7
35
264:8, 265:21, 266:16
351
89:7
36
53:10, 55:17, 146:19, 147:3, 147:5, 270:9
37
226:15
38
25:3, 25:6, 25:15, 29:10, 107:10, 149:5, 149:6, 150:8, 268:19, 268:21, 269:11, 269:19, 270:15, 271:20
39
171:9
3:20 pm
284:22
3:26 pm
284:22

4

4
158:15
4.3
75:3, 75:6
4.4
77:14, 77:15
4.5
77:14
41
49:13, 53:13, 69:8, 79:18, 80:4, 80:9, 80:13, 97:17, 97:18, 99:19, 100:17, 191:15
42
97:4, 103:19, 275:12
43
97:15, 104:6, 104:22, 106:15, 165:18, 167:1, 172:14, 173:8, 174:12, 174:14, 176:12, 178:16, 207:16, 208:22, 209:11, 212:6
44
161:2, 164:17, 217:18, 249:22, 250:1
45
124:3
46
124:13, 124:21
48
100:19, 125:7, 125:9, 125:22, 126:10, 126:21, 134:11, 134:16, 135:13, 286:10
49
49:9, 130:19, 133:2, 158:13, 171:12, 275:11, 275:13, 275:22
4900
4:5
4:02 pm
307:16
4:03 pm
307:16, 308:1

5

5,786,858
30:17
50
130:1, 130:8, 130:9, 130:13, 135:20
51
134:3, 134:15, 135:8, 136:5, 137:4, 137:6, 139:10, 140:1, 140:11, 141:2, 141:13, 294:7
52
98:9, 98:18, 98:21, 135:21, 142:3
54
140:19, 141:15, 141:20, 142:14, 143:22, 144:4, 144:8, 144:17, 144:21, 145:5
55
157:14, 268:21
56
98:9, 98:18
566945
1:23
57
50:13, 51:9, 51:21, 53:2, 86:6, 148:1, 159:1, 234:12, 234:13, 275:21
58
167:10, 256:12
59
170:2, 282:22
5a
259:8, 279:2, 285:12, 285:18, 285:22, 286:5, 286:16, 290:8, 290:20, 291:5, 293:2, 293:15, 307:3
5b
285:12, 293:3

6

6,912,351
29:6, 29:16
60
100:20, 269:16
600
68:2
62
99:21, 283:12
63
202:21, 203:5, 269:16
64
176:5
66
30:7, 257:12
69
176:5
691
6:12

7

7-1
234:15
73
185:4, 185:6, 185:11, 234:14
74
192:9, 193:3, 194:5, 194:11, 194:14, 197:13, 197:20, 200:4, 200:18
75201
3:16
79
83:9, 83:12,

84:7, 85:15, 86:17, 87:14, 89:6, 89:19, 95:14

8

8
2:21
80
255:14, 256:2, 256:7
805
6:7
84
204:12, 205:7
85
169:14, 206:13
86
166:1, 169:15, 174:10, 174:18, 178:15, 180:6, 208:15, 211:5, 214:14, 215:12, 217:8
87
152:20, 152:22, 153:13, 154:19, 156:3, 165:9, 169:15, 215:9, 215:14, 216:1, 217:6, 217:15, 218:1, 218:4, 226:2, 287:14
8a
15:2, 15:7, 183:1, 183:5, 184:9, 219:19, 220:19, 221:4, 221:9, 221:14, 223:7, 224:20, 241:5, 241:8, 265:6, 277:1
8b
15:16, 224:20, 246:16, 246:17, 247:2, 247:6, 247:14, 247:20, 248:13, 248:20, 265:7, 290:1, 290:3, 292:1, 292:10, 294:11, 294:16, 297:12, 300:10
8c
16:8, 109:11, 110:3, 112:22, 115:12, 116:1, 116:2, 116:15, 117:11, 117:16, 118:3, 118:11, 122:4, 123:11
8d
16:13, 114:10, 115:17, 117:18, 306:1

9

91
257:9, 258:11
92130
3:6
94
262:10, 262:14
95
263:20, 266:6, 268:11
97
147:3, 246:2
99
225:8, 232:4
9:28 am
61:12
9:29 am
61:12