



Planet Depos®
We Make It *Happen™*

Transcript of Hanspeter Pfister, Ph.D.

Date: February 16, 2024

Case: Realtek Semiconductor Corp. -v- ATI Technologies ULC (PTAB)

Planet Depos

Phone: 888.433.3767

Email: transcripts@planetdepos.com

www.planetdepos.com

WORLDWIDE COURT REPORTING & LITIGATION TECHNOLOGY

Transcript of Hanspeter Pfister, Ph.D.
Conducted on February 16, 2024

1 (1 to 4)

1	3
1 UNITED STATES PATENT AND TRADEMARK OFFICE	1 C O N T E N T S
2 BEFORE THE PATENT TRIAL AND APPEAL BOARD	2
3 - - - - -x	3 EXAMINATION OF HANSPETER PFISTER, Ph.D. PAGE
4 REALTEK SEMICONDUCTOR CORP.,	4 By Mr. Dokhanchy 5
5 Petitioner,	5 By Mr. Johnson 132
6 v.	6
7 ATI TECHNOLOGIES ULC,	7
8 Patent Owner.	8 Exhibits - none marked in the deposition
9 - - - - -x	9
10 Case No. IPR2023-00922	10
11	11
12 Videotaped deposition of Hanspeter Pfister, Ph.D.	12
13 Boston, Massachusetts	13
14 February 16, 2024	14
15 9:01 a.m.	15
16	16
17	17
18	18
19	19
20	20
21 Job No.: 524719	21
22 Pages: 1 - 135	22
23 Reported By: Alan H. Brock, RDR, CRR	23
24	24
25	25
2	4
1 A P P E A R A N C E S	1 February 16, 2024 9:01 a.m.
2 ON BEHALF OF PETITIONER:	2 P R O C E E D I N G S
3 BAKER BOTTS L.L.P.	3 THE VIDEOGRAPHER: Here begins Media No. 1
4 JEFFREY JOHNSON, ESQ.	4 in the videotaped deposition of Dr. Hanspeter Pfister,
5 910 Louisiana Street	5 in the matter of Realtek Semiconductor Corp. v. ATI
6 Houston, Texas 77002	6 Technologies ULC (PTAB), in the United States Patent
7 713.229.1234	7 and Trademark Office, Case No. IPR2023-00922. Today's
8 jeffrey.johnson@bakerbotts.com	8 date is February 16th, 2024. The time on the video
9	9 monitor is 9:02 a.m.
10 ON BEHALF OF PATENT OWNER:	10 The videographer today is Michael Safee,
11 MINTZ, LEVIN, COHN, FERRIS, GLOVSKY &	11 representing Planet Depos. This video deposition is
12 POPEO, P.C.	12 taking place at Orrick, Herrington & Sutcliffe LLP,
13 REZA DOKHANCHY, ESQ.	13 222 Berkeley Street, Suite 2000, Boston, Massachusetts
14 3580 Carmel Mountain Road, No. 300	14 02116.
15 San Diego, California 92130	15 Would counsel please voice-identify
16 858.314.1500	16 themselves and state whom they represent.
17 rdokhanchy@mintz.com	17 MR. DOKHANCHY: Reza Dokhanchy, of Mintz
18	18 Levin, on behalf of AMD/ATI.
19 ALSO PRESENT:	19 MR. JOHNSON: Jeffrey Johnson, Baker
20 Michael Safee, Videographer	20 Botts, on behalf of the Realtek defendants.
21	21 THE VIDEOGRAPHER: The court reporter
22	22 today is --
23	23 MR. JOHNSON: I'm sorry, Realtek
24	24 petitioners.
25	25 THE VIDEOGRAPHER: The court reporter

PLANET DEPOS
888.433.3767 | WWW.PLANETDEPOS.COM

ATI Ex. 2123
IPR2023-00922
Page 2 of 71

5 1 today is Alan Brock, representing Planet Depos. The 2 witness will now be sworn. 3 HANSPETER PFISTER, PH.D., 4 being first duly sworn or affirmed to testify to the 5 truth, the whole truth, and nothing but the truth, was 6 examined and testified as follows: 7 EXAMINATION 8 BY MR. DOKHANCHY: 9 Q. Good morning, Dr. Pfister. 10 A. Good morning. 11 Q. I'm going to hand you copies of Exhibit 1001, 12 which is the '454 patent, and 1003. And Exhibit 1003 13 is your declaration in this matter; is that correct? 14 A. Yes, that's correct. 15 Q. If you could go to Paragraph 169 of your 16 report, please: It begins on Page 81. This is where 17 you begin your analysis in Ground 2 of the '454 18 patent, claim 2; is that right? 19 A. This is with regards to particular prior art, 20 and it seems to be the combination of Amantides and 21 Kohn. 22 Q. Okay. And in this section that starts at 23 Paragraph 167, you're addressing the claimed unified 24 shader of the '454 patent; right? 25 A. That's correct, with regards to Amantides and	7 1 And so the i860 processor is basically the processor 2 in the Amantides system. So it does more, you know, 3 than just G/R processing. 4 Q. The thing you're calling the G/R unit, that's 5 a processor; right? 6 MR. JOHNSON: Objection, form. 7 A. Well, I call it a unit because I think that's 8 the term that Amantides uses. But if you could hand 9 me the Amantides reference, I can check that. 10 Q. Sure. I'm going to hand you Exhibit 1007, 11 which is the Amantides reference. 12 And I can point you to a particular place 13 if that would be helpful. 14 A. I found it. So in Amantides, Page 7 -- 15 sorry, Page -- well, which page should I use? I guess 16 I use the Bates Page 7? 17 Q. We can use 157. That's fine. 18 A. Page 157, on the bottom of the left column, 19 he describes these G/R units as the processors, or the 20 processor. So he uses the word "processor." You were 21 correct. 22 Q. So looking at Figure 3, there are multiple 23 G/R units. Those are, as we just saw, processors. 24 Each G/R unit is a processor; right? 25 A. That is correct.
6 1 Kohn. 2 Q. Okay. And if you continue onto the next 3 page, in Paragraph 169, your allegation is that 4 Amantides termed the unified shader unit the G/R unit. 5 Do you see that? 6 A. Yes, correct. 7 Q. And so your allegation is that the Amantides 8 reference discloses a unified shader in the form of 9 the G/R unit; right? 10 A. Yes, the G/R unit or units, plural. 11 Q. You didn't say units plural in your report. 12 A. That's correct. I would say it's either the 13 singular or it's the collection of. 14 Q. In your report you said singular; correct? 15 A. That is correct. I used the singular in the 16 report, but I may have, you know, missed that 17 particular point. But I think either one is fine. 18 Q. And your -- withdrawn. The Amantides 19 reference states that the G/R unit is implemented as 20 an Intel i860 processor; right? 21 A. That is correct. 22 Q. So the G/R -- 23 A. So just to be clear: You know, I'm referring 24 now to Figure 4 on Page 83 of my declaration, and 25 Figure 4 shows the actual system diagram of Amantides.	8 1 Q. And we see on the next page, 158, around the 2 middle of the page it says, "After exploring several 3 DSP microprocessors, we decided on the Intel i860"; 4 right? 5 A. Yes, that is correct. 6 Q. And so the authors are stating that the Intel 7 i860 is the processor selected as the G/R processor; 8 right? 9 A. Let me quickly check. I want to make sure 10 that we are still using the -- or Amantides is still 11 using the term "processor" in the same way. If you 12 don't mind, I'll quickly read that paragraph above. 13 Q. Yes. 14 A. I think it's a little ambiguous, just in the 15 sense that Amantides doesn't actually say that the 16 Intel i860 is just the G/R processor. The reason I 17 mention this is because, as I said earlier, in the 18 system diagram in Figure 4 the i860 appears as, you 19 know, the only processor within the graphics 20 processing unit, except for the host. So it 21 presumably will also have to do some other things. 22 And that's the only reason I bring this up. 23 Q. Okay. So looking at Figure 4, are you saying 24 that the G/R unit is coextensive with the i860, larger 25 than the i860, or smaller than the i860, or something

9 1 else? 2 MR. JOHNSON: Objection, form. 3 A. I don't know what you mean by "larger than," 4 "smaller than," whatever terms you used. 5 Q. Sure. So looking at Figure 4, what 6 components would you consider parts of the G/R unit? 7 A. Well, the G/R unit is clearly the i860, 8 within the i860, because that's the, you know, 9 one-to-one mapping to the previous diagram. 10 Q. Okay. So that's what I meant by "smaller 11 than": within. 12 What else, if anything, is in the i860 13 besides the G/R processor? 14 A. Well, I wouldn't say, you know, "is within." 15 I'm not sure what you mean by that term. 16 Q. Well, that's the word you used. So I'm using 17 your word. 18 A. Oh, sorry. I said the G/R unit, you know, is 19 basically within the i860. I mean, the i860 is -- is 20 the G/R unit. It may also function as, you know, sort 21 of-- 22 You know, it's a subtle distinction, 23 because the G/R unit actually can also do some other 24 processing. I was talking myself a little bit into a 25 corner here, because I'm thinking, besides the	11 1 Q. Sure. I'm just looking at the claim 2 language. We can talk about your mapping in a second. 3 But my question simply is, "claim 2 requires a 4 sequencer that maintains instructions; correct?" 5 A. Yes. 6 Q. And Paragraph 178 is where you discuss your 7 allegations of why Amantides and Kohn disclose the 8 claimed sequencer; is that correct? 9 MR. JOHNSON: Objection, form. 10 Q. I can just direct you to the first sentence. 11 A. Yeah, I see the paragraph. What I'm actually 12 saying is that both Amantides and Kohn disclose the 13 claimed sequencer. 14 Q. Okay. And in terms of what you identify, you 15 say at the bottom -- the end of Paragraph 178, "The 16 on-chip cache together with the memory unit comprise 17 the claimed sequencer and sequencer circuitry of the 18 '454 patent." Correct? 19 A. So I think in order to understand this, we 20 need to look at Kohn, because Kohn actually is 21 referenced here, if I'm not mistaken, Exhibit 1008. 22 Q. Well, let's just start with what you wrote. 23 You wrote, "The on-chip cache together with the memory 24 unit comprise the claimed sequencer and sequencer 25 circuitry of the '454 patent." Did I read that
10 1 geometry and the rendering processing, there is some 2 other processing happening. But that could be all 3 part -- I mean, that is actually in Amantides all part 4 of the G/R unit. 5 So, you know, let's step back. I'm going 6 to change -- basically, in my declaration, as I said, 7 I think, the G/R unit is the i860. 8 Actually, if you don't mind, I'll quickly 9 check my declaration. I want to make sure I have that 10 statement in here before I say I say that. 11 Yeah, actually, I do say that. 12 Q. Okay. 13 A. So I say in Paragraph 172 on Page 892, "The 14 claimed processor unit can be interpreted as each 15 G/R," and then in parentheses, you know, the i860 16 processor. 17 Q. And let's talk about claim 2 of the '454 18 patent. If you can open that up, please, or you may 19 just remember it. But there is a sequencer claimed in 20 the i860 -- excuse me. Withdrawn. Claim 2 requires a 21 sequencer that maintains instructions. Do you see 22 that? 23 A. So claim 2 has several subclaims. I think 24 we -- in my declaration I labeled them with A, B, C. 25 So I think you're referring to the third or the C?	12 1 correctly? 2 A. You read that correctly. 3 Q. So let's talk about what you cite here. So 4 you cite -- in this paragraph first you say, 5 "Amantides at 157 to 158." Do you see that? 6 A. Sorry, say that again. I'm sorry. 7 Q. I'm looking at the first citation in 8 Paragraph 178. 9 A. Oh, yeah. 10 Q. That's Amantides at Page 157 to 158; correct? 11 A. That's the citation. That's not all we just 12 talked about, just to be clear. 13 Q. That's fine. I just want to -- we're going 14 to go now look at that citation. 15 A. Okay, yeah. 16 Q. And then you have a quote saying, "As well, 17 program space needs to be larger," dot dot dot. Let's 18 go to that quote. Okay? 19 A. Okay. 20 Q. That's on Page 158 of Amantides. If you 21 could go there, please. So this is one of your 22 citations for why Amantides discloses the sequencer, 23 which, as we just talked about, needs to maintain 24 instructions. Correct? 25 A. The sequencer needs to maintain instructions,

13 1 correct. 2 Q. So where in this citation that you quoted, 3 Page 158 -- where do you see a disclosure of storing 4 instructions? 5 A. I'm actually trying to find it on 157-158. 6 Q. So the quote that you cited is about Line 8 7 on 158. It starts "As well." 8 A. Oh, yeah. Yeah, I think the relevant piece 9 of that quote, or of that, you know, section in 10 Amantides is that sentence at the end in my 11 declaration, Paragraph 178, where I'm quoting. And in 12 the sentence the quote says, "To get back to speed, we 13 wanted the processor to have an on-chip cache." So 14 that's what I'm referring to as the on-chip cache in 15 Amantides. 16 Q. Where does that say that what you're calling 17 the sequencer stores instructions? 18 A. If we look at Amantides, Page 158 -- also 19 where my quote starts in my declaration: "As well, 20 program space needs to be larger," et cetera. 21 "Program space would be understood by a POSITA to mean 22 memory for programs," and it's very clear that that 23 includes instructions. So it would be clear to a 24 POSITA that that program space which is stored in the 25 on-chip cache is part of that cache, meaning the	15 1 in the DRAM. 2 Q. Does Amantides say one way or another whether 3 the DRAM stores instructions? 4 MR. JOHNSON: Objection, form. 5 A. So, you know, you're directing me to this one 6 paragraph in Amantides. And so to answer your 7 question truthfully, I would have to look at, I think, 8 the rest of the reference. You know, I don't recall. 9 And I'd be happy to look through it if you like. 10 Q. Yeah, why don't do you that. It's only 11 discussed, I think, on that page, but you can feel 12 free to correct me and look for the word "DRAM" 13 throughout or whatever you want to look for. 14 But my question is: Does Amantides say 15 whether or not the DRAM stores instructions? 16 MR. JOHNSON: Objection, form. 17 A. So I think actually even within that 18 paragraph that you directed me to, you know, if you 19 read this carefully, Amantides points out that, 20 "Program space needs to be larger as each processor 21 does more work." Right? So he's pointing out the 22 fact that, as we're combining the G/R into this one 23 processor, it needs to do more work, and that means it 24 also needs more program space. And then he talks 25 about using DRAM.
14 1 instructions are stored in that cache. 2 Q. Okay. And it says originally, before 3 considering the on-chip cache, there was a possibility 4 of storing program space in SRAM or DRAM; is that 5 right? 6 A. Well, what Amantides actually points out is 7 that he or they wanted to eliminate the need for SRAM 8 and DRAM. So I guess you could say an alternative 9 design would have been to store the instructions in 10 SRAM and DRAM, but that's not -- 11 I'm sorry, in SRAM, sorry. I misquoted, I 12 think. What he actually says or they say is that "we 13 wanted to eliminate the need for SRAM, and use DRAM 14 instead." 15 And now I lost your question. I'm sorry. 16 Q. Well, my question is: Does the DRAM in 17 Amantides store instructions? 18 A. It may also store instructions, simply 19 because, you know, a cache is, if you will, kind of a 20 temporary storage and typically limited in space. So, 21 you know, it may not hold the whole program. And so 22 it is possible. 23 But Amantides does actually not disclose 24 that. I think for a POSITA it would seem reasonable 25 to say that it's possible that there are instructions	16 1 So I think it's clear from just inferring 2 this that he assumes that the program space will also 3 be stored in the DRAM, since he points out that it can 4 be large -- right? And a cache, as I mentioned 5 before, is typically a temporary storage and is 6 typically on or, you know -- I would say, yeah, 7 typically not large enough to hold all of the program 8 space. 9 So I think it's reasonable to say that, 10 yes, Amantides does say that the program space is 11 stored in the DRAM, because that's large. 12 Q. And you see in claim 2 there's a separate 13 requirement from the sequencer, and that is a general 14 purpose register block for maintaining data? Do you 15 see that? 16 A. Yes, the claim says that a sequencer is 17 coupled to a general purpose register block, and then 18 further down that, you know, that depicts a 19 calculation of operations on the selected data 20 maintained in the general purpose register block, is 21 part of that, yes. 22 Q. And could you actually look at the previous 23 page at the bottom. It says "a unified shader, 24 comprising a general purpose register block for 25 maintaining data"; right?

17 1 A. Yes, that is correct. 2 Q. And so that's a separate component from the 3 sequencer, which is for maintaining instructions; 4 right? 5 MR. JOHNSON: Objection, form. 6 A. The claim actually doesn't say "separate." 7 It says there's a general purpose register block for 8 maintaining data. "Data" is a very general term. It 9 could include instructions. 10 Q. Okay. So your read of the '454 patent is 11 that instructions can be considered to be data; is 12 that right? 13 MR. JOHNSON: Objection, form. 14 A. I didn't say that. I just pointed out that 15 in claim 2 -- you know, you have to go claim by claim, 16 and you directed me to claim 2. So in claim 2 there 17 is a sentence that says "a general purpose register 18 block for maintaining data." And so it doesn't say 19 what that data is in claim 2. So I think it's 20 undetermined, and it may include instructions. 21 Q. Okay. So when you were looking in the 22 Amantides reference for data and instructions, were 23 you using the interpretation that "instructions" can 24 satisfy the requirement for data? 25 A. I looked at each claim in turn, as you can	19 1 A. I think the claim doesn't really specify 2 that. Also, you know, you're mixing the claim and 3 Amantides; right? So could you please repeat the 4 question? Because you talked about the claim language 5 and then you mentioned DRAM, so I got a little 6 confused. 7 Q. Yeah. So assuming that the DRAM maintains 8 instructions -- 9 A. Yes. 10 Q. -- in your view, that would satisfy the 11 requirement for a sequencer to maintain instructions; 12 correct? 13 A. It doesn't say maintaining instructions. I'm 14 sorry. 15 Oh, sorry, the sequencer maintaining 16 instructions? You're switching from the register? 17 Q. I'm taking a little bit more piece by piece, 18 because it seemed like you were getting a little 19 confused. 20 So assuming that the DRAM stores 21 instructions, your view is that it can then be a 22 sequencer that maintains instructions; correct? 23 MR. JOHNSON: Objection, form. 24 A. So, you know, I want to reiterate what my 25 opinion is in my declaration, that the sequencer in
18 1 tell from my declaration. So I cannot answer this 2 question, because it really depends on the claim and 3 actually even on the part of the claim. And I 4 think -- 5 Yeah, if we can talk about the specific 6 claim, I'd be happy to answer that. 7 Q. My question was about claim 2. 8 A. Okay. 9 Q. So in claim 2, if one were to identify 10 maintaining instructions, would that satisfy the 11 requirement for maintaining data? 12 MR. JOHNSON: Objection, form. 13 A. I'm sorry, I didn't understand the question. 14 When you say -- what did you mean -- 15 Q. Sure. So we just talked about how your 16 opinion is that the DRAM stores instructions; right? 17 A. Possibly. 18 MR. JOHNSON: Objection, form. 19 A. Possibly. 20 Q. Let's assume that it does. Okay? Let's say 21 the DRAM does store instructions. Are you with me? 22 A. I'm assuming that, okay. 23 Q. Does that satisfy the claim requirement for a 24 general purpose register block maintaining data? 25 MR. JOHNSON: Objection, form.	20 1 Amantides is the on-chip cache together with the 2 memory management unit -- right? -- for claim 2. And 3 so for claim 2 my opinion is that the on-chip cache is 4 the one that maintains the instructions. 5 Q. Okay. By virtue of maintaining instructions, 6 does the on-chip cache also maintain data, as claimed 7 in claim 2 of the '454 patent? 8 MR. JOHNSON: Objection, form. 9 A. Well, the claim doesn't actually say 10 "maintain data." The sequencer only -- 11 Q. No, no, stop. "General purpose register 12 block for maintaining data." 13 A. Oh, I'm sorry. You're going back there. 14 Sorry. 15 Q. You said the on-chip cache stores 16 instructions; right? 17 A. No, I said the on-chip cache fulfills -- yes, 18 it does store instructions, correct. 19 Q. Okay. By storing instructions, does the 20 on-chip cache also store data? 21 MR. JOHNSON: Objection, form. 22 A. I'm actually quoting Kohn here. So I would 23 have to look at Kohn -- 24 Q. We're talking about Amantides, your opinion 25 in Paragraph 178 is Amantides. You say, "The on-chip

21 1 cache stores instructions." Right? 2 A. Well, to be clear, the paragraph starts by 3 saying, "Amantides and Kohn." So I'm talking about 4 both references. 5 Q. Okay. 6 A. And actually, the sentence we're talking 7 about quotes Kohn, which I'm assuming is Exhibit 1008. 8 Q. Right. Kohn -- the memory management unit is 9 what you're talking about -- right? -- in Kohn, 1008? 10 A. I'm not sure. I would have to look at the 11 reference. If you would hand it to me, I can take a 12 look. 13 Q. Let's just talk about the on-chip cache. 14 Which reference are you relying on for that? 15 A. You know, the on-chip cache is mentioned in 16 Amantides, but it's also mentioned in Kohn. And since 17 you're starting to ask me questions about data and 18 instructions, I think we will find more details in 19 Kohn. 20 Q. Okay, well, let's start with the previous 21 sentence: "Amantides features an on-chip cache for 22 storing program instructions." Right? That's your 23 opinion? 24 A. Yes, because it says -- 25 Q. I don't need an explanation.	23 1 Jeff. 2 Q. Dr. Pfister, I'm not asking about other 3 components. I'm talking about the on-chip cache that 4 your opinion states stores instructions. Okay? 5 A. Yes. 6 Q. Are you clear? 7 A. Yes. 8 Q. Does the on-chip cache therefore also 9 maintain data as required by claim 2, yes or no? 10 MR. JOHNSON: Objection, form. 11 A. I don't know. I have to look at, you know, 12 more details in Kohn to really answer that question, 13 because I'm assuming Kohn will describe it at some 14 point, what the on-chip cache actually stores. 15 Q. Okay. Can you tell me, looking at claim 2, 16 whether instructions are data? Yes or no? 17 MR. JOHNSON: Objection, form. 18 A. I'm not sure what you mean by that question, 19 whether instructions are data. 20 Q. Let's say I have an instruction. 21 A. Yeah. 22 Q. Give me an example of an instruction? 23 A. "At." 24 Q. At what? 25 A. Okay, at register 1 with register 2 and store
22 1 A. Okay, yes. 2 Q. That's your opinion. 3 A. Yes. 4 Q. That on-chip cache -- 5 A. Maybe among other things. 6 Q. Okay. Because the on-chip cache stores 7 instructions, does it also satisfy the requirement for 8 maintaining data in claim 2? 9 MR. JOHNSON: Objection, form. 10 A. Well, there are several other register blocks 11 in Amantides -- 12 Q. I'm not asking about other register blocks. 13 MR. JOHNSON: Y'all have got to stop 14 talking over one another. You let him finish his 15 sentences -- 16 MR. DOKHANCHY: I'll ask my questions 17 however I want. 18 MR. JOHNSON: If you interrupt him again. 19 I'll just keep stopping you. Let him finish his 20 answers. 21 MR. DOKHANCHY: He needs to answer the 22 actual question. 23 MR. JOHNSON: That's fine. But let him 24 finish, and then you can ask another question. 25 MR. DOKHANCHY: I'll do it however I want,	24 1 the results in register C instruction. 2 Q. Okay, that's the instruction. 3 A. Yeah. 4 Q. Is that data in claim 2? 5 MR. JOHNSON: Objection, form. 6 A. No, probably not, because it says in the 7 claim that the calculations, i.e., if you will, the, 8 you know, operations, the instructions, are on the 9 selected data maintained in the general purpose 10 register block. So I think the claim actually means 11 different data actually -- you know, vertex and pixel 12 data, or other types of data, that would be used for 13 the calculations. 14 Q. When claim 2 says "data," it means something 15 other than instructions; correct? 16 A. Yeah, I think, looking at the claim language, 17 that's a good interpretation. It says, you know, the 18 data is basically used for the operations, and data is 19 stored as a response to the, you know, operations. So 20 there seems to be in that claim a distinction, and -- 21 You know, but it doesn't actually say what 22 data is. So there's still a slight ambiguity. I 23 don't think it necessarily precludes it. But reading 24 the claim, I think you could assume that, yes, there 25 is a difference between -- that data is not

25 1 instructions necessarily. 2 Q. All right, let's go to claim element 2[a], 3 which is starting at Paragraph 170 of your report, 4 Page 82. 5 A. Sorry, we're going to my report? 6 Q. Yes, back to your report, Page 82, Paragraph 7 170. 8 A. Yes. 9 Q. This is regarding the, as it says in the 10 heading, the general purpose register block element of 11 claim 2 for the Amantides and/or Kohn references; 12 right? 13 A. Yes. 14 Q. One of the requirements of the general 15 purpose register block is that it maintains data -- 16 correct? -- in claim 2? 17 A. Yes. 18 Q. And you say -- well, first let me ask: What 19 are the items or -- what is the item or what are the 20 items that you claim are the general purpose register 21 block in Amantides? 22 A. So in this section -- sorry, I lost the 23 beginning. Here. In this section in my declaration 24 starting at Paragraph 170, I identify several units. 25 One is the DRAM, one is the input FIFO associated with	27 1 right? 2 A. Yes, I said that. 3 Q. And now we're talking about storing data. 4 Where does Amantides say that the DRAM stores data? 5 A. So in the section I quote here in Paragraph 6 170 in my declaration, Amantides says, "The new design 7 would require more memory for each processor. A lot 8 of memory is needed as textures and geometric models 9 are space intensive." So then he goes on talking 10 about DRAM. So that's part of the data that Amantides 11 mentions that is stored in the DRAM. 12 Q. Well, the part you skipped is about program 13 space, which you said is instructions; right? 14 A. Yes. 15 Q. So what it's talking about there is, as we 16 discussed earlier, potentially storing instructions in 17 DRAM; right? 18 A. Yes. 19 Q. Where does it say anything about storing data 20 in DRAM? 21 A. In the sentence before. So it says, "A lot 22 of memory is needed as textures and geometric models, 23 if they are stored locally, are space intensive." So 24 that's the data. And then he continues "as well," and 25 then instructions. So it's both.
26 1 each processor. And then there's also mention of an 2 on-chip cache. And I'm referring again to Kohn 3 here -- again, assuming that Exhibit 1008 is Kohn -- 4 showing the data cache. 5 So presumably the data cache in Kohn or 6 the on-chip cache in Amantides is this -- another 7 memory that comprises together the general purpose 8 register block. 9 Q. Do you allege that the bus that you mention 10 on Page 83 -- is that part of the general purpose 11 register block? 12 A. No. I think -- well.... 13 I don't think so. I think the bus is 14 basically used to move packets around. So, in other 15 words, it's not used for storage. 16 However, you know, it depends on the 17 actual implementation of the bus. I have certainly 18 seen some types of buses that include some local 19 storage. So there might actually be some registers, 20 potentially, associated with the bus. 21 So it's a little unclear where the bus 22 starts and where the FIFO ends, so to speak, if you 23 know what I mean. 24 Q. Okay. So let's talk about the DRAM. You 25 said earlier potentially the DRAM stores instructions;	28 1 Q. So the data that you're pointing to under 2 that theory is textures and geometric models; is that 3 right? 4 A. That's one example. 5 Q. Okay. What else is there? 6 A. I would have to look at Amantides, what else 7 they mention. 8 Q. Sure. Why don't you look at that and tell me 9 what types of data you contend are stored in DRAM. 10 A. By the way, did I mention -- if you don't 11 mind, I'm not going to answer your question, but I 12 would like to point out: I missed maybe something 13 earlier. I'm just realizing now. It also mentions 14 VRAM, which could also be considered part of the 15 register block. 16 Q. We'll get back to that question. 17 A. I just wanted to correct myself. 18 Q. Let me get your allegations correctly. So 19 your belief is that the components that could make up 20 the register block are the DRAM, potentially the bus, 21 the input FIFO, the VRAM, and the on-chip cache of 22 Amantides and possibly Kohn; is that right? 23 A. Correct, and possibly other registers that 24 are not mentioned here. But those are the ones that 25 he mentions, yes, correct.

29 1 Q. And are those generally general purpose 2 register blocks or only in combination? 3 MR. JOHNSON: Objection, form. 4 A. I hear what you're saying. I understand the 5 English. But I don't understand the meaning of what 6 you're saying; I'm sorry. 7 Q. Okay. My question is: Are you saying that a 8 DRAM is a general purpose register block and 9 separately the input FIFO is also a separate register 10 block, or are you combining all of the components and 11 calling that the register block of the claims? 12 A. I think it's individually and collectively. 13 Q. So let's get back to my question: What type 14 of data beyond textures and geometric models are 15 stored in DRAM? I'll ask the question again. 16 In Amantides, what type of data beyond 17 textures and geometric models are stored in the DRAM? 18 A. So I took a cursory reading, meaning I didn't 19 read every word in the short period of time. This is 20 a question I did not, you know, face when I was 21 studying this. 22 So by my cursory reading, Amantides does 23 mention graphic primitives earlier, just as general 24 data, that would be sent down to the graphics 25 processor. We could, you know, assume that geometric	31 1 So the geometry messages include geometry, 2 which is graphics primitives; I would say probably 3 also includes what he calls earlier graphical models. 4 Yeah, that's what I'm reading here. 5 Q. Is a geometry message an instruction? 6 MR. JOHNSON: Objection, form. 7 A. Are you referring to a particular passage in 8 my declaration or in Amantides or in a claim? What 9 are you referring to? 10 Q. So claim 2 requires instructions; right? 11 A. Yes. 12 Q. Is a -- withdrawn. Is a geometry message an 13 instruction in the meaning of claim 2? 14 MR. JOHNSON: Objection, form. 15 A. So a message, you know, is not the same thing 16 as an instruction. A message actually has a 17 particular format. It may include instructions, but 18 it certainly has other things with it. So just by 19 that observation alone, I would say no, messages, 20 geometry messages, are not instructions. 21 Q. And you said it may include instructions. 22 Are you able to say yes or no whether the Amantides 23 geometry messages include instructions? 24 A. Amantides doesn't -- 25 MR. JOHNSON: Objection, form.
30 1 models includes graphics primitives. So those would 2 then also be stored in the DRAM. 3 Yeah, I think those are the things he 4 mentions: textures, primitives, geometric models are 5 being stored in the DRAM. 6 Q. Amantides discusses geometry messages, which 7 it calls GMs. Are you familiar with that? 8 A. Yes. 9 Q. What do you understand that to be? 10 A. A message is something that goes on the 11 message bus, as you can see in Figure 4, so that is 12 something that is being sent to the different 13 processors. And so geometry -- is it "geometric" or 14 "geometry"? -- a geometric message, a GM -- well, let 15 me check. 16 Q. So the definitions are on Page 159, in the 17 top right side. Geometry message is GM and rendering 18 message is RM. Do you see that? 19 A. Yes. 20 Q. So what's your understanding of what a 21 geometry message is? 22 A. Well, it says here, "The geometry messages 23 come from the host, and their polygons are 24 transformed." So it certainly includes polygons. 25 Polygons are graphics primitives.	32 1 A. Amantides does not say that in the passage 2 I'm looking at on Page 159. I would have to look at 3 the rest of this to just make sure that he doesn't 4 mention it later. 5 Q. Same questions about rendering messages. 6 What is your understanding of what a rendering message 7 is? 8 MR. JOHNSON: Objection, form. 9 A. So the rendering messages seem to contain 10 polygons, at least. So, in other words, you know, it 11 seems that at least polygons are part of rendering 12 messages. I can't find a mention of other data right 13 now in Amantides. But, you know, it's possible. So, 14 for example, other graphics primitives could be part 15 of it; maybe, you know, maybe even textures -- I'm not 16 sure. So Amantides doesn't actually say those things. 17 But at least polygons. 18 Q. Okay. Are rendering messages instructions in 19 the meaning of claim 2? 20 A. The same answer as before. So a message is 21 more than an instruction. So a message has, you know, 22 other things in it. And it may contain instructions. 23 Again, my understanding is -- and I would 24 have to read Amantides again carefully. But my 25 understanding is that Amantides doesn't specify that,

33 1 so leaves that open. 2 Q. Let's look at Page 160 of Amantides. 3 Well, actually, let's start at the bottom 4 of 159, bottom right corner, the section entitled 5 Deadlock. Do you see that? 6 A. Yes. 7 Q. And this section and going on to the next 8 page is discussing how the authors of Amantides are 9 going to handle the deadlock problem; is that right? 10 A. That's my understanding, yes. 11 Q. And it starts on Page 160 to say that "The 12 worst possible scenario is when every input queue is 13 full of GMs." Is that right? 14 A. That's what it says there, yes. 15 Q. What do you understand the deadlock problem 16 to be that the authors are trying to avoid in 17 Amantides? 18 A. So in the first paragraph on Page 159, in the 19 section Deadlock, Amantides gives an example. 20 "Consider the following scenario," and basically 21 points out that the input FIFOs of all the 22 processors -- 23 And I'm paraphrasing. 24 Q. That's fine. 25 A. -- could be filled with GM packets. Then the	35 1 But certainly he says that the GM can be 2 converted into an RM, and in this example here -- so 3 this is an example -- right? -- a scenario, as he 4 calls it -- in this example, that is happening. So in 5 this example each processor converts the GM into an 6 RM. 7 Q. Does Amantides say that all RMs are generated 8 or converted from GMs, or does Amantides say RMs can 9 come from somewhere else? 10 A. I haven't considered that question before, so 11 I have to look at the reference for a little while to 12 find that out. 13 Q. Okay. Well, let's go back to -- or just up 14 on the same page, 159, where Amantides first 15 introduces the concept of GMs and RMs, in the third 16 full paragraph. Do you see that? 17 A. Yeah. 18 Q. It says, "The GMs come from the host, and 19 their polygons are transformed, clipped, and shaded, 20 and then sent out as RMs via the output FIFO to the 21 other processors." Correct? 22 A. Yes. 23 Q. So the GMs come from the host, and then 24 they're operated on and converted to RMs; right? 25 A. Correct.
34 1 processors convert those packets to RM and put them 2 into the output FIFO. So the output FIFOs are full as 3 well. So the scenario right now is the input FIFO is 4 full of GM packets. The output FIFOs are full of RM 5 packets. 6 So then the question is: If an input FIFO 7 has an RM, meaning a rendering message, it is 8 consumed, and that's good. But if the next packet 9 that is being sent is a GM, then there is no more room 10 in its output FIFO. So basically the processor is 11 stuck. And Amantides says, "If something like this 12 happens at several processors, deadlock occurs," 13 meaning several processors are stuck and there is no 14 progress in terms of rendering any more. 15 Q. The relationship between a GM and an RM is 16 that each processor converts a GM into an RM; is that 17 right? 18 MR. JOHNSON: Objection, form. 19 A. Well, you know, they're just messages; right? 20 So there is geometry messages and rendering messages. 21 And yes, the processor's function is, you 22 know, at least to convert messages from GM into RM, 23 but there's not a one-to-one mapping. I don't think 24 Amantides specifies that, that each GM needs to lead 25 to an RM.	36 1 Q. Do the RMs come from any other source besides 2 being converted from GMs in Amantides? 3 A. Amantides doesn't specify that, actually. So 4 it's not clear. 5 Q. Okay. Let's talk about the deadlock problem 6 again. So if you go to Page 160, at the top, this 7 first full paragraph is where Amantides discloses the 8 switching between working on GMs and working on RMs, 9 generally speaking. Is that right? 10 A. I think this paragraph describes kind of a 11 solution to the deadlock problem. Basically what he's 12 saying is that if you're switching between working on 13 RMs and GMs in the processors, then the processor will 14 always do useful work. So I believe that's what this 15 paragraph is directed to. And I'm not sure if that 16 answers your question. 17 Q. Yeah, I think that's fine. So the process 18 disclosed in Amantides is, as soon as one of these GMs 19 turns into an RM, it shuts down further introductions 20 of GMs from the host. Do you see that? 21 A. Yes, that's what he says. 22 Q. So I just want to talk about the different -- 23 A. Just to be clear, this is a scenario; right? 24 So this is an example. 25 Q. Okay. So I just want to talk about the

37 1 phases of operation that are described in this 2 paragraph. So in this paragraph on Page 160 of 3 Amantides the authors are saying first just GMs are 4 sent; right? 5 A. He's talking about the input queues; right? 6 Q. Correct. 7 A. So the GMs are in the input queue. He 8 doesn't actually say they're being sent. So they're 9 already in the input queue. 10 Q. Okay, well, we saw earlier GMs are sent from 11 the host; right? 12 A. Yes. 13 Q. And then they arrive at the input queue. 14 A. Yes. 15 Q. And at this phase there is not yet any RMs; 16 right? 17 A. Well, there might be RMs from previous GMs; 18 right? So -- 19 Q. I'm talking about the very outset of 20 operation, startup of the device. 21 A. Okay. 22 Q. So from the very first step, the first things 23 that are sent are just GMs; right? 24 MR. JOHNSON: Objection, form. 25 A. I think you're directing me to a different	39 1 We saw that on Page 159; right? 2 A. Yes. 3 Q. And there's no disclosure, we saw, or you 4 weren't able to point me to any disclosure, of the 5 host sending RMs. Fair? 6 A. Fair. 7 Q. So let's talking about Paragraph 160 now. 8 A. Sorry, which paragraph? 9 Q. Sorry, Page 160, not Paragraph 160. I 10 misspoke. 11 So the way that Amantides designs around 12 the deadlock problem -- and it says the way to handle 13 deadlock is to design it out in the first place. 14 Right? Do you see that on Page 159? 15 A. Yes. 16 Q. The way the authors chose to design around 17 the deadlock problem is that during operation, as soon 18 as one of the GMs is turned into an RM, it shuts down 19 introduction of GMs from the host to the input 20 buffers. Is that right? 21 A. Are you quoting? You seem to be reading. 22 What were you quoting from? 23 Q. I was reading Page 160, the second sentence. 24 So here's the scenario: The host is 25 sending GMs. The input queue has some GMs in it.
38 1 scenario now, because this deadlock scenario we've 2 been talking about assumes that the system is in full 3 operation, and I think only in that scenario could we 4 actually get a deadlock. 5 So this paragraph, this whole section 6 here, does actually not talk about what happens during 7 startup. 8 Q. Okay. Well, it says as soon as one of the 9 GMs turns into an RM. So by virtue of that sentence, 10 the first sentence before that triggering event 11 happens, none of the GMs have turned into an RM, 12 because it says as soon as one is furnished into an 13 RM; right? 14 MR. JOHNSON: Objection, form. 15 A. No, that's not quite correct. You know, 16 there could be previous GMs that have been turned into 17 RMs. He actually says, "As soon as one of these GMs," 18 and by "these" he means the GMs that are currently in 19 the input queue, and the scenario is that every input 20 queue is full of GMs. So we have all the input queues 21 are full of GMs, and as soon as one of these GMs is 22 turned into an RM. 23 Q. Okay. Well, let's talk about startup. So 24 we've seen that Amantides says that GMs are sent from 25 the host and they are operated on and turned into RMs.	40 1 Okay? Are you with me so far? 2 A. Yes. Just to be clear: That's not what he's 3 describing here; right? So he's describing a scenario 4 where every input queue is full of GMs. 5 Q. Okay. So I was going to get there. 6 A. Okay. 7 Q. Let me just from the start: During startup 8 or whatever -- withdrawn. 9 The system starts up, and the host begins 10 sending GMs to fill the input queues. Okay? 11 A. Okay. 12 Q. The worst possible scenario, as it says, is 13 when every input queue is full of GMs; right? 14 A. Yes. 15 Q. So as part of the solution to the deadlock 16 problem, what the authors say is "As soon as one of 17 the GMs turns into an RM," the system is going to do 18 something different: it is going to shut down further 19 introduction of GMs. Correct? 20 A. Well, he says the current GMs are still 21 slowly turning into RMs. So I don't know where you 22 saw the shutdown part. 23 Q. It's on the second sentence on Page 160. 24 Let me read it verbatim. 25 A. I'm sorry. You said -- I misunderstood. You

41 1 said "shut down further introduction of GMs from the 2 host." 3 Q. Yes. 4 A. I'm sorry, I thought you meant shutdown of 5 RMs. Sorry. I got it. 6 Q. So the solution to the deadlock starts, "As 7 soon as one of these GMs turns into an RM, the system 8 shuts down further introductions of GMs from the 9 host." Correct? 10 MR. JOHNSON: Objection, form. 11 A. Well, this is just the beginning of this 12 paragraph. Let me quickly check that this is.... 13 Can you ask your question again? 14 Q. Sure. So the first step in their solution to 15 the deadlock problem in Amantides is, quote, "As soon 16 as one of these GMs turns into an RM it shuts down 17 further introduction of GMs from the host." Right? 18 A. That's a scenario, yes. That's part of the 19 scenario, correct. 20 Q. And at that point -- withdrawn. The change 21 that it is describing is that the triggering event is 22 one of the GMs turns into an RM. Are you with me? 23 A. Yeah. 24 Q. And the result is that GMs are no longer 25 being sent from the host to the input FIFOs; right?	43 1 MR. JOHNSON: Objection, form. 2 A. He doesn't use the term "triggering event." 3 So I guess I have to ask you what you mean by 4 "triggering event." 5 Q. The event that happens that causes the system 6 to shut down introducing GMs into the input FIFOs and 7 instead, when space appears, only putting RMs in the 8 input FIFOs, that event is that one of the GMs turns 9 into an RM. Correct? 10 A. Yes, assuming that all of the input queues 11 are full of GMs. 12 Q. Okay. So other than -- 13 A. In this scenario. 14 Q. Yes. In this scenario on Page 160 at the 15 top, the two conditions are that the input queues -- 16 every input queue is full of GMs and one GM turns into 17 an RM. And when those two conditions are met, the 18 system goes from filling the input FIFOs with GMs to 19 filling the input FIFOs only with RMs. Right? 20 A. I wouldn't quite -- 21 MR. JOHNSON: Objection, form. 22 THE WITNESS: Sorry. 23 A. I wouldn't quite say filling the input FIFO 24 with RMs. I mean, what happens then, whenever space 25 appears in an input FIFO, then an RM is being
42 1 A. At that point, yes, in this scenario, yes. 2 Q. And during that time period, when space 3 appears at input FIFOs, it is the RMs that are 4 delivered to it; right? 5 A. Yes. 6 Q. So the decision to go from transmitting GMs 7 to input FIFOs to only transmitting RMs to input 8 FIFOs, that decision is only based on the triggering 9 event that one GM has turned into an RM; correct? 10 MR. JOHNSON: Objection, form. 11 A. Well, in this scenario, I think actually the 12 triggering event really is that the input queues are 13 full of GMs. Right? I think that's a triggering 14 event that has the host shut down further introduction 15 of GMs. 16 So if you want to say triggering event, 17 that to me will be the triggering event. 18 Q. But that wouldn't be enough. That wouldn't 19 actually trigger the change. It says the worst 20 scenario is when every input queue is full of GMs. 21 The triggering event is, as soon as one is turned into 22 an RM, it shuts down further introductions. So the 23 triggering event is one GM turns into an RM. 24 MR. JOHNSON: Objection, form. 25 Q. Right?	44 1 delivered to it. 2 Q. Are there any other events or analyses or 3 actions in the system that cause the system to go from 4 the state of transmitting GMs from the host to put 5 into the input FIFOs to the scenario you just 6 described, where when space appears in the input FIFOs 7 the RMs are delivered to it? 8 MR. JOHNSON: Objection, form. 9 A. When you say are there any other, you seem to 10 imply there was one. 11 Q. You said -- okay. 12 A. So are you referring to the thing we just 13 talked about? 14 Q. Yes. 15 A. So in this scenario that he's describing 16 here, no, that seems to be the -- you know, as soon as 17 GM -- as another GM turns into an RM, that will shut 18 down the -- that will trigger all of this. 19 Q. Okay. And then the next phase of operation 20 described in Paragraph 60 -- sorry, Page 60, first 21 paragraph, it starts with the sentence "Now." And it 22 says, "Now, when all the output FIFOs are empty and 23 the processors are working on the remaining RMs in 24 their input FIFOs can the host begin to deliver GMs 25 again." Do you see that text?

45 1 A. Yeah, I do see that text. There was a little 2 bit of text in between that you described. So he also 3 says, you know, we stopped our discussion whenever 4 space appears at an input FIFO, it's the RM. There's 5 going to be an RM that's delivered to it. Eventually, 6 all of the GMs will be turned into RMs and will be 7 delivered into their appropriate input FIFOs. So that 8 seems to be before the "now"; right? 9 So in this scenario there is also the time 10 that the GMs will -- it says, "All the GMs will be 11 turned into RMs and will be delivered to the input 12 FIFOs." 13 Q. Okay, yeah, that's fine. We'll talk about 14 that. 15 So in the phase where no GMs are permitted 16 to be sent to the input FIFOs and only RMs are 17 permitted to be sent to the input FIFOs, it describes 18 that eventually all the GMs are operated on and 19 they'll be turned into RMs. 20 A. Yes. 21 Q. That's what you're talking about. Okay. 22 Then there's a transition point to 23 explaining under what conditions GMs will be allowed 24 to be sent to the input FIFOs; right? 25 A. So when you say "an additional condition,"	47 1 to be permitted to be put into input FIFOs, is that 2 all the output FIFOs are empty, and as to the GMs that 3 are in the input FIFOs, the processors are working on 4 them. Right? 5 A. Yes. 6 Q. Sorry, I said that wrong. 7 A. I know the sentence you're referring to. 8 Q. So the thing that happens.... 9 Are you all set? 10 A. Yes. 11 Q. The thing that happens that permits the GMs 12 to be put in the input FIFOs again is that the output 13 FIFOs are empty and, as to the RMs that are in the 14 input FIFOs, the processors are working on those. 15 Correct? 16 A. Did you say the output FIFOs are empty? 17 Q. Yes. 18 A. Or did I hear you say the input FIFOs? 19 Q. No, I said the output FIFOs. 20 A. The output FIFOs, yes, correct. 21 Q. So it's that, and it's that there are RMs in 22 the input FIFOs, and the processors are working on 23 those; is that right? 24 A. Yes. 25 Q. Is there any other analysis or check or
46 1 can you please specific exactly what you're referring 2 to? 3 Q. That's what I want to get at. But I'm saying 4 there's something that happens and then eventually GMs 5 can be put in the input FIFO again. 6 A. By the host. 7 Q. By the host. Well, by anybody, because in 8 that period no GMs were allowed to be put in the input 9 FIFOs; right? 10 MR. JOHNSON: Objection, form. 11 Q. Neither by the host nor by anybody else. 12 MR. JOHNSON: Objection, form. 13 Q. It says, "Whenever space appears at any input 14 FIFO it is the RMs that are delivered to it," meaning 15 not GMs; right? 16 A. Yes, that's what it says, correct. 17 Q. And it goes on to explain -- and we'll get 18 into this, what it is -- but something else happens, 19 and then once again GMs are permitted to be put into 20 the input FIFOs; right? 21 A. Yes, by the host. 22 Q. Yes, by the post. It happens to be by the 23 host. 24 A. Yeah, yeah, yeah. 25 Q. And the thing that happens, that causes GMs	48 1 condition that causes the system to return to a phase 2 in which the host can begin delivering GMs to the 3 input FIFOs once again? 4 A. I mean, there may be, but in this scenario he 5 doesn't describe it. 6 Q. And you map in your report GMs to the claimed 7 vertex operations; is that correct? 8 A. Can you direct me to my report where I say 9 that? And when you say "claim," you're still talking 10 about claim 2, or what are we talking about? 11 Q. Well, I think your opinions on that point are 12 the same for all claims, but you can correct me if I'm 13 wrong. 14 You're aware that all of the claims 15 require vertex operations; right? 16 A. Correct, yes. 17 Q. And in Amantides -- you can turn to Paragraph 18 184 -- you use a slightly different word. You say 19 "vertex processing tasks." 20 A. Can you tell me where we are on the page? 21 Q. So 184, last sentence before the block quote. 22 A. Yes. 23 Q. You equate the vertex tasks on -- withdrawn. 24 You make a correspondence between the GMs 25 and the vertex processing tasks; right?

49 1 A. Yes. 2 Q. And similarly, the RMs you say are the pixel 3 processing tasks; right? 4 A. Yes. 5 Q. So your interpretation of the claims -- the 6 claim language doesn't say "vertex processing tasks," 7 technically. It says, for example, in claim 1, 8 "vertex operations." Right? 9 A. Yes. 10 Q. So in claim 2 it says "vertex calculation 11 operations." 12 A. Yes. Well, yes. 13 Q. When you say "vertex processing tasks," 14 you're referring to those operations; is that right? 15 A. Well, you know, the tasks really maps to the 16 messages here; right? And a message -- we talked 17 about this earlier -- is more than just an instruction 18 or, you know, a processing task -- sorry, a processing 19 calculation operation. So I don't think there's a 20 one-to-one mapping between the two. 21 Q. Okay. So let's just start more simply, then. 22 What do you claim are the vertex calculation 23 operations recited in claim 2? 24 A. Do you mean in Amantides specifically? 25 Q. Yes. What meets that limitation in	51 1 operations, "are carried out by executing instructions 2 on pixel and vertex data." Do you see that? 3 A. Yes. 4 Q. And is it correct that the pixel data is the 5 RM in Amantides? That's your allegation? 6 A. No. I mean, the RM is a message. Pixel data 7 is something else. 8 Q. Okay. And then is the vertex data the GMs in 9 your analysis? 10 A. No, again, GMs is a message, and the data -- 11 the vertex data is something else. 12 Q. If you go to previous Page 177 -- sorry, 13 Paragraph 177. 14 A. Yeah. 15 Q. You say, "Amantides also discloses that its 16 processor units perform geometry/vertex operations in 17 response to receiving 'geometry messages' with vertex 18 data (GMs)." Do you see that? 19 A. Yeah. 20 Q. So you just told me the GMs and the vertex 21 data are different. So what does this mean, then? 22 A. Well, it receives geometry messages. So the 23 GM in parentheses is really referring to the geometry 24 messages; right? So vertex data can be part of a 25 geometry message.
50 1 Amantides? 2 I can point you to what I think is the 3 closest. I don't think it's very clear in your 4 report, I'll be candid. But I think this is what 5 you're saying. So if you go to Page 89 -- well, go to 6 88. You'll see the Claim 2 Sequencer heading? 7 A. Yeah. 8 Q. And the sequencer element in claim 2 is where 9 we see the first appearance of "vertex calculation" 10 and "pixel calculation operations," if you look at 11 claim 2. 12 A. Okay, yes. 13 Q. So this is where I would expect to see your 14 allegation of what the vertex calculation and pixel 15 calculation operations are. And the closest thing I 16 can see is that you say at the top of Page 19, last 17 sentence of that paragraph -- 18 A. Sorry, you're back in Amantides; right? 19 Q. No, I'm in your report. 20 A. Paragraph 189? 21 Q. It's Paragraph 179, that starts on the bottom 22 of 88, and goes to the top of 89. 23 A. Yeah, okay. 24 Q. So you end the paragraph with, "These 25 operations," which are presumably the claimed	52 1 Q. Okay. 2 A. As we talked about earlier, Amantides talks 3 about, you know, the geometry data, and I think he 4 calls it something -- "geometric model data" or 5 something like that. So vertex data is part of that. 6 Q. And in terms of executing vertex calculation 7 operations, in Amantides you point to operating on the 8 GMs as meeting that; is that correct? 9 A. I don't think I say that -- unless you can 10 point me to it. 11 So here in this paragraph we were just 12 looking at, in Paragraph 177, what I'm saying is that 13 the rendering/pixel operations are performed in 14 response to receiving these messages, either with 15 geometry data -- i.e., you know -- that could include 16 vertex data. So I think it's more correct to say that 17 the geometry processing happens on the vertex data, 18 but the rendering messages -- sorry, the geometry 19 messages are actually different. It doesn't happen on 20 the geometry messages itself. 21 Q. You're saying it happens on the data within 22 the geometry message. Is that the distinction you're 23 talking about? 24 A. That's one way to put it, yes -- I mean, 25 being in this case the vertex data, yes.

53 1 Q. So let me just understand: So you say that 2 there's vertex data in the GMs. Is that right? 3 A. That's one example, yes, among other things. 4 There could be vertex data in the GMs. 5 Q. Where else is there vertex data? 6 A. Sorry, when you say "where else," are we 7 talking about the messages now, or we're talking about 8 anywhere in the processor, or.... 9 Q. Well, what I'm getting at is: What is your 10 allegation of where Amantides discloses executing 11 vertex calculation operations? 12 A. Well, first, I mean, the whole article is 13 directed towards, you know, computer graphics; right? 14 And he describes -- I should really say they 15 describe -- you know, basically different graphics 16 operations, including vertex operations. 17 So, you know, I'm just picking one 18 example. The front end -- I'm looking at Amantides, 19 Page 156, the paragraph on the lower right, just below 20 Figure 1. "The front end, which performs a series of 21 geometric operations, is typically implemented as a 22 pipeline of ALU," so that would be the processing -- 23 the processors, sorry -- or the processing units, 24 rather. "And then afterwards the transformed shaded 25 primitives are sent to the back end." So geometry	55 1 Q. Is that right? 2 A. Yeah. I mean, you know, that is fair. You 3 know, I'm using the term "tasks" here instead of 4 saying "messages." But yeah. 5 Q. But -- well, withdrawn. You here associate 6 vertex tasks with GMs and pixel tasks with RMs. Is 7 there anywhere in your report where you associate 8 vertex tasks with RMs? 9 A. I honestly don't know. I would have to 10 check. 11 Q. Well, my impression is that throughout your 12 report you associate vertex with GM and pixel with RM. 13 I just want to make sure that that's your 14 understanding. 15 A. Yeah, so vertex tasks or, you know -- I 16 probably should have used the term "message," really. 17 So vertex message -- I mean, vertex processing 18 messages are GMs, yeah. I think that's probably true, 19 yes. I think you are correct. 20 Q. And the pixel processing task or the pixel 21 processing message, those are RMs; right? 22 A. Yes. 23 Q. And you don't have any allegation that the 24 reverse is true, where vertex processing tasks are 25 mapped to RMs; right?
54 1 processing at least includes transforming and shading 2 primitives. So that's the type of processing that 3 would happen on geometry data. 4 Q. Okay. 5 A. Or, you know -- yeah. 6 Q. Let me help focus us a little bit. So in the 7 discussion of the deadlock problem on Page 160 -- if 8 you could turn to that. 9 A. Yes. 10 Q. When it's talking about GMs, does that relate 11 to vertex data and vertex operations, or something 12 else? 13 A. So geometry messages can include vertex data, 14 as we have just identified. So it can include that, 15 yes. 16 Q. Okay. 17 A. But, you know, the message is not processed, 18 just to be clear. I mean, the processing happens on 19 the vertex data. That's the geometry processing, as 20 referred to in the claim. 21 Q. I think we're kind of talking at different 22 levels. I'm speaking a little bit more high level, as 23 you do in Paragraph 184 of your report. You associate 24 vertex tasks with GMs and pixel tasks with RMs. 25 A. Okay.	56 1 A. I'm not sure, to be honest. Again, I would 2 have to read my declaration a little bit more 3 carefully. I haven't considered that question before. 4 I mean, I can't tell you, actually. I can't answer 5 that question. I mean, I'm happy to answer, but I 6 have to check first. 7 Q. Okay. Why is it that a geometry message has 8 any relationship with a vertex task? 9 MR. JOHNSON: Objection, form. 10 A. Well, I don't really say that it has -- I 11 mean, I'm basically calling them equal; right? So a 12 vertex task, I call it geometry message or vice versa. 13 But we know that they come from the host; 14 right? And we know that they include geometric 15 primitives, including vertices. So that's the 16 relationship. 17 Q. Okay. And why is a rendering message 18 equivalent to a pixel processing task? 19 A. Well, it's basically a message that goes and 20 is eventually, you know, converted into pixels. So it 21 might include some transformed geometry, and so 22 eventually this is going to be converted into pixels. 23 Q. Okay. The RM -- when you say "it" or "this," 24 are you referring to the RM being converted into 25 pixels?

57 1 A. No, no. Sorry. When I meant "converted to," 2 I actually meant the data within the RM. So the data 3 within the RM could be some transformed geometry, 4 which is then rasterized, let's say -- I'm just 5 picking an example -- but it can be rasterized by the 6 processor, and the rasterization leads to pixels. 7 Q. And then is that before or after the creation 8 of the RM? 9 A. So the RM is created first, sent to the 10 processor. The processor receives it, then does the 11 rasterization and then creates the pixels. So the 12 pixels are created after the receipt of the RM. So 13 that's why I call it a pixel processing task; right? 14 I don't say -- I don't say it has pixels in it. What 15 I meant is, it's a pixel processing task. I guess 16 that's explaining my use of the term "task" here. 17 Q. So is it the rasterization process that is 18 the pixel processing task in your example? 19 A. That's one example. That's one example that 20 can turn a geometry into pixels, yes. 21 Q. And would you consider that a vertex 22 processing task also? 23 A. No. I think a vertex processing task, you 24 know, generally speaking, is more a transformation, 25 for example. That's an example of a vertex processing	59 1 MR. DOKHANCHY: Why don't we take a 2 ten-minute break. 3 THE VIDEOGRAPHER: We are going off the 4 record. The time is 10:45 a.m. 5 (Recess taken.) 6 THE VIDEOGRAPHER: We are back on the 7 record. The time is 10:56 a.m. 8 Q. All right, Dr. Pfister, if you'd grab the 9 '454 patent, Exhibit 1001, turn to claim 11. Claim 11 10 requires "an instruction store and wherein the 11 processor unit of the unified shader performs the 12 vertex manipulation operations and pixel manipulation 13 operations at various degrees of completion based on 14 switching between instructions in the instruction 15 store." Do you see that? What does it mean to 16 perform operations at various degrees of completion to 17 you? 18 A. Yeah, I think it's a good question. I think 19 the patent isn't very clear about this, you know, so I 20 have to interpret this. 21 So one way to interpret this, as an 22 example, is, you know, you're starting an instruction 23 or, a, you know, vertex manipulation operation, as 24 they call it here. And, you know, you might -- you 25 might stop it, and then you might do another either
58 1 task. 2 Q. And would you consider a transformation to be 3 a pixel processing task? 4 A. When you say a transformation, what exactly 5 do you mean? 6 Q. Whatever you just meant in your last answer. 7 A. So what I meant is transforming by a matrix; 8 right? So you have a 3D vertex position, for example, 9 and you transform that 3D vertex position by 10 multiplying it by a transformation matrix, and that 11 gives you a transformed vertex. 12 Q. Is that a pixel processing task? 13 A. It is possible to, you know, multiply pixels 14 by a matrix to get, for example, a different color of 15 the pixel. So that is certainly possible. That's one 16 example, though. 17 Q. And are those -- that example, is that a 18 geometry message? 19 A. No, again, we're talking about, you know, 20 operations now. We're talking about the processing. 21 So we're talking about, you know, what's actually done 22 to the data; right? And a message is not that. A 23 message is not the actual processing, as I talked 24 about earlier. A message contains, for example, 25 vertex data, but it may contain other things as well.	60 1 vertex or pixel manipulation operation before you 2 resume that first operation. 3 So I think that's one way to interpret "at 4 various degrees of completion." 5 Q. Would it suffice to change vertex 6 manipulation operation and/or pixel manipulation 7 operation after the full completion of that operation? 8 Would that satisfy the various stages of completion 9 requirement? 10 MR. JOHNSON: Objection, form. 11 A. I think I need to ask you what you mean -- 12 what is meant by "full completion"; right? So, for 13 example, another way to interpret this might be that 14 there is some manipulation operation done, but then 15 maybe the data for that operation is still, you know, 16 somewhere. 17 Let's say an addition; right? So an 18 addition is finished, but maybe the result of the 19 addition isn't written back to memory yet, for 20 example. I'm just picking an example. 21 Q. Okay. 22 A. So does that mean the operation is completed 23 or not? I mean, I think by this definition -- by my 24 understanding, also -- it would be at some degree of 25 completion -- right? -- but it wouldn't be fully

61 1 completed. 2 So to come back to your question: I think 3 it depends what you mean by -- I forget the word you 4 used; that's why I couldn't finish. 5 Q. Let's just move on. I was just using the 6 claim language. 7 So let's presume there are definite 8 boundaries of when an operation begins and when an 9 operation ends. Let's take that as an assumption. 10 Okay? 11 A. Okay. 12 Q. So operation 1 has a defined boundary and 13 then operation 2 has a defined boundary. Does that 14 make sense so far? 15 A. Okay. 16 Q. Would it meet the claim language if the 17 switch between operation 1 and operation 2 was after 18 operation 1 was completed? 19 A. Probably not, because it says "various 20 degrees of completion." Again, the patent is 21 unfortunately not clear what that means; right? But I 22 think what they mean is it's something else than what 23 you just described; right? So it's not that they're 24 completed yet; it's that they're at some degree of 25 completion. So the term "degree of completion"	63 1 intuitively understand is, literally, as a pipeline, 2 you put data in, it's being processed in steps, and 3 then the results come out. As you push more data in, 4 more results come out. 5 Q. Let me hand you a copy of Exhibit 1008, which 6 is the Kohn reference. And in your report you cite to 7 Pages 22 to 23, so I'd like to go there. And because 8 you don't really have any specific explanation in your 9 report, I'd like you to point me to what you're 10 referring to in Kohn as supporting the notion that 11 Kohn discloses executing operations at various stages 12 of execution. 13 A. Yeah, actually, I do have an instance here, 14 and it's the sentence right after where we stopped 15 reading. So I say in my report, "For instance, Kohn 16 describes the execution of operations in parallel, 17 specifically in a 'dual-operation mode' within the 18 'pipelined modes,' where by both floating-point 19 operations and scalar operations could take advantage 20 of the pipelining capability," and referring back to 21 Kohn. 22 Q. So why does that disclose operation at 23 various stages of execution? 24 A. Well, on one hand, it's this dual operation. 25 So you have two types of operations. You have
62 1 probably means that the operation hasn't completely 2 finished yet, as you said. 3 Q. Okay. Can you turn to Paragraph 201 of your 4 report, Page 98. 5 A. Yes. 6 Q. You see the heading for c, Claim 11[b] 7 Instruction Store, and this is where you have some 8 allegations regarding the various stages of completion 9 requirement. Do you see that? 10 A. I see the Instruction Store title. 11 Q. And you see in around Line 5 you mention 12 "different operations being loaded into the pipeline 13 at various stages of execution"? 14 A. Yeah, and I think I'm referring here to Kohn. 15 And yes -- right, "at various stages of execution," 16 correct. 17 Q. And so you mentioned the pipeline. Can you 18 explain to me what a pipeline is in the context of 19 Kohn? 20 A. Yeah, I think he's using the term as one 21 skilled in the art would understand it, which is 22 operations in a processor are typically pipelined, 23 meaning that some things happen before other things 24 kind of in sequence, and they move down different 25 units in lockstep; right? So an easy way to kind of	64 1 floating-point operations and scalar operations 2 happening. So that's one part; right? And then the 3 other part is advantage of the pipelining capability. 4 Again, somebody skilled in the art would understand 5 that, you know, pipelining means you can actually push 6 operations down, but that doesn't mean the operation 7 is completed yet; right? And so, I don't know, it 8 could be that, I don't know, we just have finished an 9 addition but we haven't written back the result yet to 10 memory. So that would be part of the pipeline. 11 And then something else can happen -- for 12 example, a floating-point operation could happen -- 13 before the result is being written back. And so that 14 is how things would be at various stages of 15 operations. 16 Q. But if you look at the claim language, the 17 operations have to not just be at different various 18 degrees of completion; the processor unit needs to 19 switch between instructions such that it performs 20 vertex manipulation and pixel manipulation operations 21 at various stages of completion. 22 So my question is: Where is the switching 23 between operations that are at various degrees of 24 completion in Kohn? 25 A. So coming back to the quote we just looked at

65 1 in my report, referring to Paragraph 201, 2 approximately in the middle, this dual-operation mode 3 has two types of operations: floating-point 4 operations and scalar operations. So, in other words, 5 if we go back to the claim, the instructions, you 6 know, that the claim is referring to -- in this case, 7 one type of instruction might be a floating-point 8 instruction, and another type of instruction might be 9 a scalar instruction. So it could be that a 10 floating-point instruction is being processed in this 11 pipelining mode -- not finished yet; it's at a stage 12 of degree -- at a degree of completion. Right? And 13 then a scalar operation might start, or, you know, a 14 scalar instruction might be starting to execute. So 15 in that case the processor would point to the -- or 16 switch to the scalar instruction in the instruction 17 store. 18 So that's the switching between 19 instructions, whereas, you know, there is still a 20 degree of completion. So the first one may not have 21 finished yet -- as an example. 22 Q. So looking at Figure 7, is there some part of 23 this that you're calling -- I think you gave the 24 example of multiplication and adding. Are you 25 pointing to the two different blocks, multiplier unit	67 1 time" part is not required. But Kohn happens to do 2 this, these two operations. 3 And so basically, one is in some degree 4 of, let's say, of completion, whereas, you know, Kohn 5 then points to the other. And one instruction could 6 be a floating-point operation and the other one could 7 be a scalar operation. 8 Q. So in this example of floating-point 9 operations and scalar operations, are those the two 10 parallel paths or are those both -- are those the two 11 parallel paths? 12 A. When you say "parallel path," what are you 13 referring to? 14 Q. It says "operation with." Your report says, 15 "Kohn describes the execution of operations in 16 parallel, specifically in 'a dual-operation mode' 17 within the 'pipelined mode,' whereby both 18 floating-point operations and scalar operations could 19 'take advantage of the pipelining capability.'" 20 And earlier you said a pipeline is a 21 sequence of operations on a piece of data; right? 22 A. Fair. I mean, we would have to define what 23 "operations" are. But fair, yes, in the broadest 24 term, yes. 25 Q. So my question is -- so one pipeline would be
66 1 and adder unit? 2 A. I haven't pointed to anything in Figure 7. I 3 would have to find where he talks about dual-operation 4 mode within those two pages. I don't know if you know 5 where that is. But that's what I just described. So 6 I didn't point at anything in Figure 7. I would have 7 mentioned Figure 7 if I did -- in my declaration, I 8 mean. 9 Q. So you pointed to Pages 22 to 23. So that's 10 where I presume you are saying the dual-operation mode 11 is. 12 A. Yes. I mean, if we were able to search, I 13 would find it really quickly. But I would have to 14 read to actually find it right now. 15 Q. So to the best of your understanding right 16 now, what is the switching between instructions in the 17 instruction store in Kohn that you are relying on in 18 your report? 19 A. So I'm just giving one example; right? I'm 20 picking out one example, one instance, I say here "for 21 instance." So one would be this dual-operation mode, 22 which clearly implies that there are two operations 23 being performed, you know, in Kohn at the same time. 24 I want to be clear: That is not required 25 by the claim, by the way. You know, the "at the same	68 1 a serial sequence of operations; right? 2 A. It gets a little confusing. You know, people 3 say -- and I'm talking now about somebody skilled in 4 the art -- would say that pipelining is a form of 5 parallelism, because you actually have different data 6 going down the pipeline at the same time. So that 7 gets a little complicated. So that's why I wouldn't 8 call it a sequence in that sense. But, you know, if 9 you just focus on one data item, you're right: one 10 data item is being processed in a sequence of 11 operations. That is correct. 12 Q. Okay. So is that what the dual-operation 13 mode is, the form of parallelism that you just 14 described, whereby processing blocks are in series but 15 they have different data, and thereby achieving 16 parallelism? 17 A. Well, it's actually two things; right? I 18 think there's two types of parallelism, if you will. 19 There's the pipelined mode; so that's what we've been 20 talking about, in terms of pipelining. And as I 21 described earlier, different instructions can be at 22 various degrees of completion down the pipeline. So 23 that's one form of parallelism or one form of 24 pipelining. 25 And then the other that Kohn describes

69 1 here is this dual operation, which I understand to 2 mean that Kohn describes, and I describe this at the 3 end of the paragraph, that the processor has the 4 capacity to perform both vertex and pixel operations 5 simultaneously -- again, not required by the claim, 6 but certainly meets the claim language. 7 So these two operations, like in this case 8 a floating-point operation and a scalar operation, 9 seem to be happening at various degrees of completion 10 at the same time. 11 Q. Are those floating-point operations and 12 scalar operations in sequence with one another and 13 operating on different data, or is it some different 14 structure that is the dual-operation mode? 15 A. Good question. I would have to familiarize 16 myself a little bit in more detail. 17 I think what I describe and what I quote 18 here in the declaration meets the claim language. But 19 we can certainly talk about more specifics. 20 So what I'm not sure about, for example, 21 is does it have to be different data or could it be 22 the same data that is being operated on? I don't know 23 that. I would have to actually look at the 24 instruction set of the i86 -- the i860, rather; sorry. 25 Q. The switching between instructions in the	71 1 let's say. 2 A. Let's say data A. Let's say it's a vertex. 3 Let's say vertex A. 4 Q. Then what's the next step? 5 A. So the switching between instructions would 6 then be, let's say, the processor -- while -- you 7 know, the vertex operation on data A has not 8 completed; okay? That's still ongoing. 9 Now the processor starts let's say a 10 scalar operation on data B; right? 11 Q. Okay. 12 A. So that's the switching of instructions. And 13 both of these instructions are now in various degrees 14 of completion. 15 Q. And in your example are the vertex operation 16 on data A and the pixel operation on data B, are those 17 happening simultaneously or not? 18 A. Yes, that's the specifics of the i860 19 processors. As I also say in my declaration, it has 20 the capacity to perform both of these operations 21 simultaneously. Again, that is not required by the 22 claim, but it's certainly -- the processor is capable 23 of doing that. 24 Q. Okay, so the processor is doing both of those 25 operations at the same time. Then what changes so
70 1 instruction store, you're saying that is the -- that 2 is the fact that there's floating-point operations and 3 scalar operations; is that right? 4 MR. JOHNSON: Objection, form. 5 A. Not exactly. I mean, you have to -- again, 6 as somebody skilled in the art, you would understand 7 that, in order to have a floating-point operation or a 8 scalar operation, you'd have to at some point point to 9 instructions in the instruction store. So at some 10 point these instructions have to be pointed to and 11 have to be actually -- start execution. So that's.... 12 Q. But then what's the switch, is my question? 13 A. Right. So let's say at some point in time 14 we're pointing to the first instruction, and let's say 15 for argument's sake the first instruction in this 16 dual-operation mode is a floating-point operation. 17 Q. Okay. 18 A. And let's say that starts executing and goes 19 down the pipeline. So it's at some degree of 20 completion; right? It's not finished; it's at some 21 degree of completion. 22 Q. Can I stop you right there? Are you 23 referring to -- in your example can you explain what 24 data is being operated on? So you're saying that 25 floating-point operations are operating on data A,	72 1 that there is a switch between instructions? 2 MR. JOHNSON: Objection, form. 3 A. I'm not sure. I thought I just explained 4 that. Like we started executing one instruction, and 5 then we switched, you know, to the other instruction; 6 right? 7 Q. Well, but I believe you just said they're 8 happening simultaneously. So that's the distinction I 9 understand. If they're happening at the same time, 10 there's no switch. So what's the switch? 11 A. Yeah, so the switching between instructions 12 in the instruction store, that's what we're referring 13 to, that when -- so let's say, again, I'm starting 14 with my first vertex instruction; right? So that's my 15 first instruction. It starts to execute. It's at 16 some degree of completion, not finished. 17 Then I'm switching. That's the switching 18 I'm referring to, to a pixel operation on data B. So 19 that's the switching between instructions. But the 20 first operation is still at some degree of completion. 21 And by the way, so is the second. None of 22 them have completed, actually. 23 Q. So I see that you said there's a switch, but 24 earlier you said the operations on A and B are 25 happening simultaneously. So --

73 1 A. Yeah, I was a little bit vague there in my 2 terminology, which is common. Again, we have two 3 types of parallelism; right? One is when the 4 instructions execute -- that's this dual-operation 5 mode -- like what type of instructions I can execute. 6 And the other is the pipelining. 7 So when I said "simultaneously," what I 8 was referring to was really the pipelining aspect. 9 The first one is -- the first instruction is still 10 being executed while the second instruction starts 11 execution. 12 So because of the pipelining, they're both 13 executing simultaneously, at various degrees of 14 completion. 15 Q. Okay. So two processes going on 16 simultaneously at various stages of completion; right? 17 A. Yes. 18 Q. And then the switch is what? 19 A. It's just when I start a second one, that's 20 when I switch. So I go from the first one to the 21 second one. 22 Q. Well, so, that's the confusion. You're 23 saying you go from the first one to the second one. 24 But when you're saying they're operating 25 simultaneously, you're not leaving the first one. The	75 1 what happens in the i860 processor. But I do think 2 that does mean -- that does meet the claim language. 3 Q. Okay. So in the scenario where you're 4 operating on instruction 1 and you then fetch 5 instruction 2 and begin operating on instruction 2 6 while continuing to operate on instruction 1, that's 7 the switch that you're talking about. Is that right? 8 A. I think the switch happens when I'm -- in the 9 instruction fetch unit, when I'm going from the first 10 instruction, which we said was, let's say, a vertex 11 instruction, to fetching the second instruction. 12 That's when I'm switching from the instruction store. 13 So to be precise: The first instruction 14 is stored at memory location A in the instruction 15 store. The second instruction is stored at memory 16 location B in the instruction store. So when I'm 17 fetching from one to the -- sorry, when I'm switching 18 from one to the other in the fetch unit, I'm referring 19 to two different locations in the instruction store. 20 So that is the switching that I'm referring to. 21 Q. Okay. 22 A. The switching from one instruction to another 23 instruction in the instruction store. 24 Q. And switching what you are fetching is what 25 you're talking about, not what you're executing.
74 1 first one is still ongoing. 2 A. Yes. 3 Q. So what.... 4 A. So maybe let's -- maybe I need to take a step 5 back. 6 For somebody skilled in the art, they 7 would know that to execute an instruction has several 8 steps; right? One is instruction fetch, instruction 9 decode, instruction execution, and then writeback of 10 the results. So those are, largely speaking, the 11 steps that are followed by any processor. 12 So when we're talking about switching 13 between instructions in the instruction store, that 14 seems to me to refer to the instruction fetch, because 15 it's referencing in the instruction store. So we're 16 fetching a different instruction from the instruction 17 store. We're switching from one instruction to 18 executing another instruction. 19 Q. Okay, I think that's the disconnect. I'm 20 interpreting "switch" as meaning you switch from 21 instruction 1 to instruction 2, you no longer are 22 working on instruction 1. But it sounds like your 23 interpretation is you can still be working on 24 instruction 1. Is that right? 25 A. Well, it's not my interpretation. That's	76 1 A. Yes. 2 MR. JOHNSON: Objection, form. 3 A. You are correct. That's what I just 4 explained as an example; right? 5 Again, we're looking at the i860 as 6 meeting the claim language. You know, there could be 7 other embodiments of this, but that's one that I think 8 meets the claim language. 9 Q. If we could jump back to claim 2. 10 Sorry, on that last point: What is the 11 evidence in Kohn or Amantides that that switch could 12 happen when the operations are at various stages of 13 completion? 14 A. So I think that's -- you know, again, coming 15 as a person of skill in the art, it's understood that 16 "taking advantage of the pipelining capability," I 17 think is really the key phrase here. So as I 18 described earlier, you know, as the first instruction 19 is executing in this pipelining -- right? -- in this 20 pipelining fashion, so it's not finished, it's at some 21 degree of completion, we are switching to the second 22 instruction. And so that's -- 23 Q. Is there anything that would make sure that 24 the fetch happens when the first instruction is not 25 complete, or are you just saying over time it should

77 1 happen? 2 MR. JOHNSON: Objection, form. 3 A. I don't quite understand where you're going 4 with the question. I don't know what you mean. 5 Q. So is there anything in the Amantides system 6 that would say, "Do that fetch of the second 7 instruction before the execution of instruction 1 8 finishes completion"? 9 A. So, you know, the i860 instructions are 10 executed as quickly as possible -- right? -- because 11 it's a high-performance processor. So there is always 12 an instruction being fetched as quickly as possible, 13 unless there is some kind of a stall in the pipeline. 14 So as long as there is no stall, instructions are 15 executed before other instructions are finished. At 16 least it's fair to assume that, if we're just talking 17 about two instructions that come right after each 18 other, it is fair to assume that the first instruction 19 will not be completed by the time the second 20 instruction is being -- is being switched to in the 21 instruction store. 22 Q. Okay. But you're not saying there's anything 23 in Amantides to ensure that the second fetch happens 24 before the first fetch -- before the first instruction 25 execution finishes. You're not saying that, are you?	79 1 consistent. Maybe I was a little bit more loose and 2 high-level in my language earlier, and now I'm being a 3 little bit more precise. I'm not sure. 4 Q. Okay. That's fine. We can move on. So 5 let's go to claim 2, and column 7. You'll see that 6 "the processor unit executes instructions that 7 generate a pixel color in response to selected data 8 from the general purpose register block." Do you see 9 that? 10 A. Yes. 11 Q. What do you understand to be the process of 12 selecting data? 13 A. So are you asking me just reading the claim 14 language, or are you referring to a particular prior 15 art? 16 Q. Just in the claim language. What is required 17 to select data in claim 2? 18 Let me start with a simpler question. 19 Does the selected data have to be in the general 20 purpose register block when it is selected? 21 A. Yeah, it looks that way. It looks like the 22 claim says selected data from the general purpose 23 register block. So that selected data at some point 24 was selected from the general purpose register block. 25 Q. And let's look at claims 7 and 10. You see
78 1 MR. JOHNSON: Objection, form. 2 A. No. I mean, unless, you know -- again, there 3 could be -- I don't know the instruction set off the 4 top of my head of the i860 processor. You know, there 5 could be a particular feature where maybe the 6 processor would ensure that, but I'm not familiar with 7 that. 8 But I think in general, you know, that's 9 actually the whole point of taking advantage of the 10 pipelining capability, that you can start with the 11 second instruction while the first instruction is 12 still being executed. 13 Q. I thought that you said the advantage of the 14 pipelining was so that different processors can work 15 on different blocks of data and introduce a sense of 16 parallelism. Isn't that the major advantage of 17 pipelining? 18 MR. JOHNSON: Objection, form. 19 A. You're getting me back into a different 20 context again, something we talked about I don't know 21 how many questions before. You know, we've been going 22 down into more and more detail. So what I'm talking 23 about now is sort of a detailed explanation. 24 You'd have to refresh my context again 25 about what I said earlier. I think it's all	80 1 that in 7 there is "the shader of claim 5, further 2 including a selection circuit operative to provide 3 information to the store in response to a control 4 signal"? Do you see that? 5 A. I see that, yeah. 6 Q. And then claim 10 builds on claim 7 and says 7 that the arbiter -- or "an arbiter" provides the 8 control signal; is that right? 9 A. Yes, that's what it says, correct. 10 Q. And let's turn to Paragraph 194 of your 11 declaration, Page 95. So this is your section on 12 dependent claims 7 and 10 in Amantides and Kohn, and I 13 want to actually point you to Page 97, and my question 14 is about the arbiter of claim 10. And if you could 15 point me to what you say in Amantides is the arbiter, 16 please. 17 A. Yeah, Amantides describes this, you know, at 18 the level where a POSITA would understand what the 19 arbiter is. 20 So he basically -- and I'm referring to 21 Paragraph 196 in my declaration, starting on Page 96. 22 "Amantides refers to assigning low-priority and 23 high-priority packets to the time slots of the bus as 24 'arbitration.'" So there is circuitry that, you know, 25 puts these packets onto the bus.

81 1 And then further describing now on Page 2 97, "As well, there is a fair scheduling policy in 3 which a second packet from a FIFO cannot get onto the 4 bus until other FIFOs are given a chance." So that's 5 that. 6 And then, "'Running at 10 megahertz, 2 7 time slots are required for arbitration.'" 8 So he's basically describing a process 9 whereas packets have to be on the bus not at the same 10 time slot, they have to be on the bus at two different 11 time slots, which physically makes sense. And that 12 process, as a POSITA would understand, you know, 13 requires some circuitry that allows that to happen. 14 So that circuitry is what I point to as 15 the arbiter. He calls it the arbitration -- sorry, I 16 call it the arbitration at the processor, but behind 17 the processor is also some circuitry that's required 18 for making that happen. 19 Q. Okay. So you are saying a POSITA would know 20 that that arbitration would have to be performed by 21 logic circuits not described in Amantides; correct? 22 A. Correct. 23 Q. And that's the arbiter; right? 24 A. That logic circuit, yes. 25 Q. Does that arbiter perform the action of	83 1 work. 2 Q. Okay. 3 A. Or how this could work. 4 Q. So your view is that the selection circuits 5 decides which packets get put onto the bus and the 6 arbitration is for time-slot organization? Is that 7 right? 8 A. I think that's one embodiment, yes. 9 Q. Do you describe any other embodiment in your 10 report? 11 A. In my report I don't, no. Looking at this 12 paragraph, this section here, in this section I don't. 13 I mean, actually, I might as well refer 14 back to the overview of Amantides, just to make sure, 15 if you'd like me -- 16 Q. I don't know if we have time to go hunting. 17 I think we can just move on. 18 MR. DOKHANCHY: Let's take a quick, 19 five-minute break. 20 THE VIDEOGRAPHER: We are going off the 21 record. The time is 11:40 a.m. 22 (Recess taken.) 23 THE VIDEOGRAPHER: We are back on the 24 record. The time is 11:44 a.m. 25 Q. All right. Dr. Pfister, you recall the
82 1 ensuring that low-priority packets get onto the bus 2 only if no high-priority packets are waiting to be 3 sent? 4 A. Well, that's actually the selection circuit 5 that's referred to in the claim. 6 Q. So that concept of ensuring low-priority 7 packets get onto the bus only if no high-priority 8 packets are waiting is not part of the, quote, 9 "arbitration." Is that fair? 10 A. Yeah, Amantides, you know, refers to the, you 11 know, assigning the packets to the time slots on the 12 bus as arbitration, you know. And the selection of 13 the packets either as low-priority packets or high- 14 priority ones, that is a different process that is 15 done by the selection. 16 Q. I guess what I'm saying is: Is it the 17 unshown arbiter that has that final say of only 18 low-priority packets getting onto the bus in the event 19 that there are no high-priority packets, or does that 20 happen from some other entity? 21 A. I think it's the selection circuit that 22 determines that and not the arbitration circuit. 23 That's the way I interpret Amantides. And again, he's 24 vague about the specifics. But I think that's one 25 embodiment where, you know, where this would actually	84 1 requirement in claim 11 of the '454 patent to switch 2 operations at various degrees of completion? Do you 3 recall that? 4 A. Yes. 5 Q. Could you go to your report at Paragraph 256. 6 This is going to be in Ground 3, no longer Ground 2. 7 And that will be Page 125. 8 A. So now we switched to Selzer -- 9 Q. And Fiske, yes. 10 A. Okay. 11 Q. So this is -- you see on the previous page 12 the heading 9b, The Processor Unit? And this is where 13 you describe your allegations regarding the various 14 degrees of completion requirement; is that right? And 15 I'll point you to the sentence in Paragraph 256 that 16 says, "Fiske describes this process being done in a 17 fine-granted" -- I think you mean "fine-grained" -- 18 "manner that can switch among different instructions 19 of a thread, and at a time when the entire thread is 20 not completed." Do you see that? 21 A. I see that, yes. 22 Q. And is that your rationale for why Ground 3 23 discloses the performing-operations-at-various- 24 stages-of-completion requirement? 25 A. Yes, that's one of them.

85 1 Q. Are there any others that you identify in 2 your report? 3 A. I'm referring back to the discussion I have 4 in previous claims, where there's a selection which 5 should be used for the operations. So, you know, by 6 reference, I would like to include that in my answer. 7 I'm not recalling if I mention anything there. But I 8 would assume that I would have put it here if I did. 9 But just in case I didn't, I want to 10 include those references as well. 11 Q. So your citation here is just "id." So I'm 12 not able to tell what you're referring to. Can you 13 find for me where or tell me if you know where Fiske 14 describes switching among different instructions of a 15 thread when the entire thread is not completed? 16 A. If you don't mind, could I get the reference? 17 Q. Yes. 18 A. That would help. 19 Q. I'm going to hand you a copy of Exhibit 1010, 20 which is the Fiske reference. 21 Actually, before we dig into it, let's 22 just look at your report for one more second. 23 So the claim language is that the 24 processor unit performs vertex manipulation operations 25 and pixel manipulation operations at various degrees	87 1 instruction is at various stages of completion. 2 A. I'll be perfectly honest: You know, Fiske is 3 a long reference. I have no idea where to start. So 4 I'm happy to read it, but, you know, that's going to 5 take time. If that's okay with you, I'm happy to do 6 that. 7 Q. Well, you have your report, so feel free to 8 look at your report. But -- and if you don't know 9 where it is, that's fine. But this is your opinion, 10 that the claims are invalid over Fiske. So you should 11 be able to tell me where Fiske discloses switching 12 when an instruction is at various stages of 13 completion. 14 A. Okay, I'm going to start looking at Fiske. 15 Q. Okay. And again, we don't have all day, so 16 if you can't find it, then -- 17 A. I apologize. I should have included the 18 reference here, which, again, a typo. So I didn't 19 actually specify the reference in my report. I 20 apologize for that. 21 So just as a partial answer -- I mean, I 22 found one mention here of what's going on in Fiske. 23 I'm looking at Fiske Page 212, Section 2.3, when he's 24 talking about context switching. And he says what's 25 happening in context switching is that "11 cycles,
86 1 of completion based on switching between instructions 2 in the instruction store. And your statement in your 3 report says switching among threads at a time when the 4 entire thread is not completed. 5 So what is the relationship between 6 threads and operations? 7 A. Well, a POSITA would know that a thread 8 includes instructions; right? So the thread is made 9 up of one or more instructions. So those are the 10 operations I referred to. 11 Q. Are the instructions the operations or are 12 the threads the operations? 13 A. So I'm actually saying here that Fiske 14 describes switching among different instructions, so 15 the instructions are the operations. 16 Q. Okay. And then when it says "the entire 17 thread is not completed," what is the relevance of 18 that to performing operations when -- excuse me, at 19 various stages of completion? 20 A. Yeah, I would have to look at Fiske at this 21 point, because I need to see what I'm actually 22 referring to. 23 Q. Okay. Let's look at Fiske, the instructions 24 being the operations. And please point me to where 25 Fiske describes switching when the -- when an	88 1 it's used to drain the processor pipeline (five 2 cycles), execute a trap handler which saves state, 3 chooses the next context to execute using a round 4 robin scheme (six cycles), duplicating instruction 5 pointers," et cetera, et cetera. 6 So here Fiske describes a processor where 7 they're switching from one context to another, which 8 takes many cycles, and also talking about duplicating 9 instruction pointers, so meaning pointing to different 10 instructions in the instruction store. And knowing as 11 somebody skilled in the art, even without reading the 12 whole of Fiske, I know that they have a pipeline 13 processor. 14 So it's, I think, fair to assume that in a 15 context switch there will be instructions at various 16 degrees of completion -- or, I should say, vertex and 17 pixel manipulation operations. 18 Q. Can you perform a context switch at a time 19 when an individual instruction is incomplete? 20 A. Well, again, as somebody skilled in the art, 21 I would assume a context switch happens because an 22 instruction couldn't complete or maybe a thread 23 completed. So we're switching from, you know, one 24 context to another. 25 So in the case where the instruction

89 1 couldn't complete because it's waiting for a memory 2 reference or something like that, that instruction 3 would be at some degree of completion; right? It 4 wouldn't be complete yet. Yet we're switching to a 5 different context. 6 Does that answer your question? 7 Q. Sort of. In that instance is it -- is the 8 instruction in -- 9 So let me take a step back. Let's give an 10 example of an instruction, so something like calculate 11 new vertex point, something like that. Does that make 12 any sense as an instruction? 13 A. Not really. 14 Q. So why don't you give me an example. 15 A. An instruction is maybe more simple. Let's 16 just say "add," adding two values together and making 17 the result. 18 Q. Add two values. 19 A. Yes. 20 Q. So is there an intermediate point of that -- 21 of executing that instruction during which a context 22 switch could happen? 23 A. Yes. So, for example, the operation is at, 24 you know, the value of this memory location to the 25 value of that memory location and store the result in	91 1 you know, switched to in the instruction store. So 2 that includes things that are being switched. 3 But it could also switch some state 4 registers or, you know, some other state, et cetera. 5 Q. So that's what I was getting -- sorry, I 6 didn't mean to cut you off. That's what I was asking. 7 So the context is state data, essentially; is that 8 right? 9 A. I mean, that's another term that's being 10 often used. Again, these terms are not 100 percent 11 precise, because every processor implements the 12 context or the state differently. 13 Q. Okay. 14 A. And even people sometimes make a distinction 15 between context and state. So state could be part of 16 the context, but the context could also sometimes 17 include like other data. 18 So I'm saying it's not -- these terms are 19 not clearly defined. And right now we're talking 20 about Fiske, and I don't think, if I'm recollecting 21 this correctly, that Fiske gets down to that level of 22 detail to answer that question exactly. 23 Q. Okay. Let's jump back to claim 2. In column 24 6 at the bottom we see -- we talked about this 25 before -- but a general purpose register block is
90 1 the register. Okay, so that's the instruction. 2 When I'm starting to execute that 3 instruction, the first thing I need to do is I need to 4 fetch the values from those two memory locations. It 5 could be, and it's very likely, when I execute this 6 instruction the first time that that memory location 7 is off chip, it's in the DRAM, and so I have to 8 actually wait for that data to arrive. 9 So it is possible that at that point I'm 10 going to switch context and wait till that data 11 arrives. So that instruction is kind of on hold, and, 12 you know, it's put in some kind of a scoreboard so 13 that I can keep track of it, whereas, you know -- at 14 some degree of completion, whereas I'm executing the 15 context switch, which takes a long time, so I'm doing 16 something else and then I'm executing other 17 instructions at the same time. 18 Did that make sense? 19 Q. I think so. I think I understand. 20 And then the switch concept in a context 21 switch, what is being switched? 22 A. Good question. It's processor-dependent. 23 So, you know, a lot of things are being switched. I'd 24 say that certainly some data is being switched out of 25 the registers and maybe a new instruction is being,	92 1 claimed, and that general purpose register block is 2 for maintaining data. Right? 3 A. Yes. 4 Q. What's your interpretation of the term 5 "block" in the general purpose register block? 6 A. So now we're just talking about the claim 7 language and we're not specifically talking about the 8 prior art? 9 Q. That's correct. 10 A. I think it's basically any memory or memories 11 that make up a general purpose register. 12 Q. Is there any requirement on what can or 13 cannot be a single block? 14 A. I don't think the patent is that specific. I 15 think the patent is very broad. So I guess there's no 16 specific requirement as long as it's memory and 17 capable of storing data. 18 Q. So in your experience as an expert in this 19 field, have you come across register blocks before? 20 A. Yes. 21 Q. And can you give me an example of what sorts 22 of register blocks you've come across in your 23 experience? 24 A. Sure. A very classic one, which we've 25 already seen before, is a FIFO, first-in/first-out

93 1 type of memory. That register block or that, if you 2 will, memory block has a particular function -- right? 3 It's first in, first out. So that's one example. 4 Q. And let's say you take that FIFO that's on, 5 say, the i860 processor: Would you consider that 6 memory and something outside the i860 processor to be 7 part of a single block? 8 A. You know, we're talking about general purpose 9 register block; right? "Register," again, to somebody 10 skilled in the art probably means something that's on 11 the processor, because the things referred to outside, 12 like the DRAM, is typically referred to as the DRAM. 13 Now, I know that by including the DRAM in 14 the register block, because some people, again, use 15 that language loosely -- so the patent is not 16 specific. So I'm interpreting it in the broadest 17 possible way. 18 Q. What about, you know, in your ordinary usage, 19 would you consider a FIFO in a processor and a 20 separate DRAM to be part of the same block? 21 A. Are you asking me as me personally or as a 22 POSITA? 23 Q. As a POSITA. 24 A. As a POSITA, I think some people do, yes. 25 Actually, I have seen, you know, people use it that	95 1 this case, yes. 2 Q. So you don't think there's anything needed in 3 terms of interconnection or physical proximity in 4 order for two memories to be considered part of the 5 same single block. Is that fair? 6 A. Well, it gets a little tricky because, as you 7 know, the claim language does have "coupled to" 8 sometimes. 9 Q. But that's outside the register block. 10 A. Yes. 11 Q. I'm talking about -- 12 A. So just the register block itself, yeah, I 13 don't think it has to be next to each other, as long 14 as it's somehow coupled through the whole system. I 15 mean, the processor ultimately has to be able to 16 access the data in that -- wherever the register block 17 is; right? So it has to be coupled to the processor 18 at some point. I mean, it has to be coupled -- not at 19 some point. It has to be coupled. 20 Q. What about a FIFO within a processor and 21 hard-disk storage via an I/O connection outside of a 22 computer? Would you consider the hard-disk storage 23 and the FIFO in the processor to be part of the same 24 register block? 25 MR. JOHNSON: Objection, form.
94 1 way, because they might just consider all of that a 2 register block. 3 Q. But what makes those two things that are 4 disparate part of the same block? 5 MR. JOHNSON: Objection, form. 6 A. You know, I think what we're really talking 7 about is the functionality of this. So this is a 8 register block or a memory. Maybe they should have 9 used the term "memory." I don't know. But it's 10 basically something that can store general purpose 11 data; right? So I think that's how people use it 12 sometimes. So again, that would basically be the 13 broadest possible interpretation. 14 So it's any kind of memory that can store 15 general purpose data. And so if you look at it that 16 way, it includes things that are also off chip, like 17 the DRAM. 18 Q. So in your view, could the general purpose 19 register block be any combination of memories wherever 20 they're located as long as each of them stores general 21 purpose data? 22 A. Yes, I think that's -- I mean, some people 23 would interpret it that way. 24 Q. Would you personally interpret it that way? 25 A. I do so in my declaration, yes. I mean, in	96 1 A. So now you're getting into hypotheticals; 2 right? I mean, I haven't considered that question in 3 this case because neither the patent nor any of the 4 prior art refer to outside disk space in any form. 5 So first of all, I am not pointing to any 6 outside disk space in my declaration, so that just was 7 never a question I was considering. 8 And secondly, I probably would draw the 9 boundary there, because at that point you're talking 10 about a different medium. Now you're talking about a 11 magnetic medium, whereas before we had electric 12 circuits that are coupled to each other. 13 So I probably would draw the line. I'd 14 probably say no, that's not part of the register 15 block. But that's a hypothetical. I'm giving you my 16 answer as a POSITA on the spot here. 17 Q. If you could turn to Page the 114 and 18 Paragraph 229 of your report. What is your allegation 19 of what in Selzer is the claimed general purpose 20 register block? 21 A. So just referring to Selzer, that there are 22 programmable memories and FIFOs within the rendering 23 processors and the master module, and both of those 24 include general register blocks in the form of 25 programmable memories and FIFOs. So the programmable

97 1 memories and FIFOs is -- inside those is what I'm 2 pointing to. 3 Q. Okay. 4 A. And actually, if we look on the figure on the 5 next page, you can see there is programmable memory 6 and FIFOs being labeled in the figure. 7 Q. Okay. 8 A. Now, may I add something? 9 Q. Maybe. I'm asking about your report right 10 now, and I'm not asking about any new opinions you may 11 have. So if you're going to introduce a new opinion, 12 I would say let's focus on your report right now. 13 So the programmable memories and the 14 FIFOs: So if you're looking at Figure -- I believe 15 this is Figure 4 -- 16 A. It's Figure 4.4. I turned it 90 degrees, I 17 believe, because it's a very large figure. 18 Q. Let me hand you Exhibit 1009, which is 19 Selzer. 20 So Figure 4.4 of Selzer describes FIFOs, 21 and then there's something called PROG memory. And 22 you're saying that stands for programmable memory; is 23 that right? 24 A. Yes. And I may add, since it is actually 25 part of my opinion: You know, I say, "Each of the	99 1 included in what you're calling the general purpose 2 register block? 3 A. Yes, the rendering processor and the master 4 module. So I referred to both in my declaration. 5 Q. Are you aware of where Selzer states that the 6 programmable memory stores data as the general purpose 7 register block must do? 8 A. I don't recall if Selzer explicitly says 9 that. But, I mean, that's understood by a POSITA. I 10 mean, that's the purpose of the memory, is to store 11 data. 12 I think the term "memory" implies a place 13 where you store data. That's -- a POSITA -- that's -- 14 it's not even a POSITA. I think that's just what 15 memory in an electric circuit does. That's the 16 function of it. 17 Q. Right. But earlier we talked about how claim 18 2 recites data and also instructions. And so what I'm 19 getting at is: Does Selzer ever say the program 20 memory stores data rather than instruction-type 21 information? 22 A. I don't recall. I would have to look through 23 Selzer to see what they actually store. I mean, it 24 is -- 25 Yeah, I don't recall, to answer your
98 1 rendering processors includes general register 2 blocks," et cetera. 3 Q. Where are you looking? 4 A. I'm sorry, I'm going back to my report, 5 Paragraph 229. 6 I think what I'm implying here is that the 7 rendering processors, which are the, you know, 8 different DSPs here -- the DSPs themselves, you know, 9 might actually include FIFOs and some memory as well. 10 I mean, that's what a POSITA would understand. 11 So I think I'm kind of including those as 12 well in the general purpose register block. 13 Q. Okay. So on programmable memory, your 14 assertion is that's what's indicated by "PROG" memory. 15 It's not, for example, program memory. Is that fair? 16 A. Yes, that's fair. 17 Q. And your assertion is that the programmable 18 memories plus the explicitly disclosed FIFOs plus 19 potentially FIFOs within the DSP combined are the 20 general purpose register block; is that right? 21 A. Yeah, and plus possibly other registers 22 within the DSP -- other memories within the DSP as 23 well. 24 Q. Are the FIFOs and programmable memories 25 within the master module at the top of Figure 4.4	100 1 question. I would have to look through it. 2 Q. If you turn to Paragraph 236 of your report. 3 This is regarding the sequencer in claim 2. Right? 4 A. Yes. 5 Q. And what is the sequencer, in your opinion? 6 A. It's the master module. 7 Q. Anything else, or just the master module? 8 A. Well, I'm actually combining -- I'm combining 9 Fiske with Selzer. So Paragraph 237 -- this also -- 10 you know, if we combine the two, it would teach a 11 sequencer where the -- I think this should be the 12 master module combining with Fiske's scheduling 13 module. 14 Q. Okay. So within Selzer you point to the 15 master module as the sequencer; right? 16 MR. JOHNSON: Objection, form. 17 A. I'm, you know, pointing to both Selzer and 18 Fiske for the claimed sequencer. So it's the 19 combination of the two. 20 Q. In the combination of Selzer and Fiske, is 21 there anything other than, within the Selzer 22 reference, the master module that you're pointing to 23 for the sequencer? 24 A. I'm sorry, can you repeat the question? 25 Q. Sure. So you said that you're pointing to

101 1 the combination of Selzer and Fiske for the sequencer; 2 right? 3 A. Yes. 4 Q. Within Selzer are you pointing to anything 5 other than the master module? 6 A. For the sequencer? 7 Q. Correct. 8 A. No, I don't, at least not in this section 9 here we're looking at in my report, starting on 10 Paragraph 236. 11 Now, I don't recall if I describe anything 12 else when I -- in my overview description of Selzer. 13 But I believe the -- this one is pretty clear that the 14 master module in Selzer is the only thing I'm pointing 15 to. 16 Q. Okay. And the master module -- withdrawn. 17 The sequencer of claim 2, among other 18 things, must maintain instructions operative to cause 19 the processor unit to execute calculations; correct? 20 A. Correct. 21 Q. And you cite in your report that one of the 22 ways that is met is that the master module in Selzer 23 accesses the graphics data in the system and fetches 24 the next primitive or task automatically; right? 25 A. Can you refer me to the sentence where I say	103 1 finishes a last task. 2 Q. So in that instance the geometry module 3 accesses the geometry bus and fetches the next 4 primitive or task automatically. Correct? 5 A. In that case, yes. 6 Q. Okay. And what does "automatically" mean in 7 that sentence? 8 A. It seems to imply that -- and I'm 9 interpreting as a POSITA here. But it seems to imply 10 that the geometry module has some circuitry that lets 11 it know that it finished the last task; right? And so 12 once that circuitry or maybe instructions -- I'm not 13 sure -- but once that mechanism, that, you know, 14 circuitry, has detected that it finished the last 15 task, then it will fetch the next primitive or 16 instruction -- the next primitive or task or 17 instruction. 18 So the "automatically" implies -- or 19 refers to it has detected that it finished the last 20 task, and now it goes to the next task. So it 21 finished processing the last task and now it's the 22 next one. 23 Q. And it says that the fetching is done 24 automatically; correct? 25 A. Yes, that's what it says. It says "fetches
102 1 that? 2 Q. It's the second sentence in Paragraph 236. 3 A. Yes. 4 Q. Why does that indicate to you that the master 5 module maintains instructions operative to cause the 6 processor unit to execute calculations? 7 A. Well, the term "task" implies some set of 8 operations, and operations in a processor require 9 instructions. So it's requiring that it fetches 10 instructions, by this description here. 11 Q. And what does "automatically" mean in this 12 context? 13 MR. JOHNSON: Objection, form. 14 A. So I'm looking at my citation in Selzer, and 15 I'm looking at Page 43, Section 4.4, the second 16 paragraph. That's where I have the citation from. 17 So in this example here -- and it seems to 18 be an example -- the master module traverses graphics 19 data structures and puts them into the data FIFO. So 20 that's the data FIFO here in the master module, as we 21 can see in the figure. 22 And then if a geometry module has finished 23 a last task, it accesses the geometry bus and fetches 24 the next primitive or task automatically. So this 25 seems to be one example, like when a geometry module	104 1 automatically," yes. 2 Q. And from where does the geometry module fetch 3 the next task automatically? 4 A. From the FIFO. So it's the previous sentence 5 says that the master module has put, you know, the 6 primitives into the data FIFO. 7 Q. So that's primitives. Where does it fetch 8 the tasks from? 9 A. Well, you know, I'm assuming it's also from 10 the FIFO, because that's -- that's very likely where 11 it's from. 12 Q. And when you say "the FIFO," looking at 13 Figure 4.4, which FIFO? 14 A. So he's referring to Figure 4.5. 15 No, I'm sorry. I guess we have to look at 16 Figure 4.4. 17 I mean, if I look at the Figure 4.4, it 18 will be the FIFO within the master module. 19 Q. And where does it say that the tasks are 20 stored in the master module? 21 A. Well, you know, if it's fetching the task 22 from the FIFO and the FIFO's in the master module, it 23 stands to reason that the instruction is there. 24 Q. Right. What I'm saying is where this 25 reference says that the instructions or the tasks are

105 1 in the master FIFO. 2 A. I think we can just refer to the section I 3 just pointed to. Let me see something else. 4 So going all the way back in Selzer to 5 Page 40, Paragraph 4.3.1, "The master module is the 6 system supervisor, handles the communication to the 7 host processor, and is responsible for startup and 8 synchronizing activities." 9 So by "synchronizing activities" and 10 "startup," it also needs to handle instructions. 11 Q. Let's go to the description of the master 12 module. So the next section, I believe it is, 13 4.3.1.1. It says that the master module -- this is 14 the second paragraph -- "traverses the graphics data 15 structure and feeds graphics data to a special FIFO 16 for delivering to the appropriate processors." Right? 17 A. Yeah. 18 Q. And then on the next page, Section 4.4, it 19 says in the second paragraph -- 20 A. Sorry, page? 21 Q. This is Page 43. 22 A. Yeah. 23 Q. Section 4.4, the second full paragraph, "The 24 master module traverses the graphics data structure 25 and puts the high-order primitives, like splines,	107 1 point has to receive those instructions and 2 temporarily store them somewhere. 3 So I think just from a systems point of 4 view, a POSITA would understand that has to be the 5 case. 6 Q. In your report, Paragraph 236, you wrote, 7 "The master module in Selzer accesses the graphics 8 data in the system and 'fetches the next primitive or 9 task automatically,'" and you cited Page 43. Correct? 10 A. Yes. 11 Q. In the quote that we looked at, it's actually 12 the geometry bus that fetches the next primitive or 13 task automatically; right? 14 A. It's the geometry module, yes. 15 Q. You wrote in your report "the master module" 16 does that. So what you wrote is wrong? 17 A. I'm sorry, that's a typo. 18 Q. Okay. 19 A. I was clearly citing -- not successfully, 20 apparently. I was citing this particular sentence, 21 because this is the only sentence on this page that 22 "fetches the next primitive or task automatically." 23 So I mistyped. 24 MR. DOKHANCHY: Why don't we break for 25 lunch.
106 1 polygons, meshes, and triangles, into the data FIFO." 2 Right? 3 A. Yes, I read that, yes. 4 Q. So in those two instances the master module 5 is putting graphics data into the FIFO; correct? 6 A. Yes, but not exclusively. So that's one 7 thing it puts. But if you read the next sentence, 8 where you just ended, if a geometry module has 9 finished the last task, it accesses the FIFO, 10 basically, the geometry bus, and fetches the next 11 primitive or task. So it's referring to task, and 12 "task" would include instructions. 13 So it's more than just the data stored in 14 the FIFO. 15 Q. So other than that quotation on Page 40 that 16 you said indicates to you that the instructions are 17 stored in the master module, is there anything that 18 explicitly says the instructions are stored in the 19 master module? 20 A. I mean, a POSITA would understand that has to 21 be the case, because the master module is the one 22 connected to the host. So, you know, instructions 23 have to come from the host, and they have to be stored 24 somewhere on the graphics processor. So that would 25 imply the graphics -- sorry, the master module at some	108 1 THE VIDEOGRAPHER: We are going off the 2 record. The time is 12:32 p.m. 3 (Recess taken.) 4 THE VIDEOGRAPHER: We are back on the 5 record. Time is 1:31 p.m. 6 Q. All right, Dr. Pfister, if you could look at 7 claim 1 of the '454 patent. You see the requirement 8 around column 6, line 55, it says, "Performing vertex 9 operations on the vertex data by a processor within 10 the unified shader unless the general purpose register 11 block does not have enough available space therein to 12 store incoming vertex data." Do you see that? 13 A. Yes. 14 Q. Can you go to Paragraph 140 of your report, 15 on Page 68. We're now going to be looking at Ground 16 1, the Lindholm references. 17 A. Sorry, 140? 18 Q. Yeah, so Page 67 begins your analysis of 19 claim 1, and then on to Page 68 -- or we can start on 20 Page 67. 21 A. So this is Lindholm; is that correct? 22 Q. That's right, this is Lindholm. So my 23 question is: What is the vertex data that you are 24 pointing to within Lindholm, the claimed vertex data? 25 A. So I see vertex data mentioned a couple times

109 1 here. Are you asking me what specifically is vertex 2 data? 3 Q. Yeah, what in the Lindholm references are you 4 considering to be the vertex data of the claims? 5 A. Oh. Can I look at the Lindholm references? 6 Q. Yes. I am handing you Exhibits 1005 and 7 1006, which are the two Lindholm references. 8 So if you can look at your report, 9 Paragraph 140, you say -- this is six lines from the 10 bottom -- "Similarly, the Thread Control Unit receives 11 vertex data from the Vertex Input Buffer and pixel 12 data from the Pixel Input Buffer," citing Exhibit 1005 13 at 550 to 6.3. So let's start there. If you could go 14 to that part of Exhibit 1005 and let me know what 15 items you are considering to be vertex data. 16 A. Yeah, at some point Lindholm uses the term 17 "sample," which could be either vertex or pixel data. 18 I think we have to look a little bit 19 earlier in Lindholm, and I'm looking at my report 20 here. I'm actually saying in Paragraph 140, "As 21 discussed earlier under claim 2," yada, yada, you 22 "receive vertex data." So I'm actually looking back 23 in my report under claim 2. And there, for example, 24 in my report, on Paragraph 90, I say, you know, "Each 25 execution pipeline can perform vertex operations such	111 1 storage, then it's full. So that would be in the 2 claim language it doesn't have enough available space 3 to store the incoming vertex data. 4 I think what I'm saying here is that, you 5 know, it's clear to a POSITA that if some memory is 6 full, you can't actually store any more incoming data 7 in it. 8 Q. Where does Lindholm specifically discuss 9 performing vertex operations and then evaluating 10 whether there's room for vertex data, in which case 11 ceasing vertex operations? 12 A. Well, the claim doesn't say the word 13 "evaluating." You know, it says "unless the register 14 block does not have enough available space." So there 15 is no "evaluating" here. It just says that if there 16 is not space in the memory, you can't store anything 17 more in it. That's what it requires. 18 Q. Okay. And your evidence for that being met 19 is your statement at the top of Page 69 that a POSITA 20 would know that that would happen; correct? 21 MR. JOHNSON: Objection, form. 22 A. I mean, that's one piece of evidence. You 23 know, I think a POSITA would know that if you can't 24 store data in the general purpose register block -- 25 yeah, if it's -- you know, the thread control unit
110 1 as transforming vertices objects based on another 2 coordinate system," referencing Lindholm at 4:42. 3 So clearly "vertices" is vertex data. 4 Q. So vertices and samples. Is there anything 5 else that you're considering to be vertex data? 6 A. Well, I'm referring to Page 50, Paragraph 92 7 in my report, quoting Lindholm, Exhibit 1005, saying 8 that -- and now he's using the term "sample." "Such 9 as intermediate data generated during processing of 10 the sample." So presumably there could be some 11 intermediate data that is associated with vertex data. 12 Q. And where are your allegations regarding why 13 Lindholm meets the claim 1 requirement to perform 14 vertex operations unless the general purpose register 15 block does not have enough available space therein to 16 store incoming vertex data? 17 A. So without reading, you know, the Lindholm 18 references back to back, which I'd be happy to do but 19 I'm assuming we don't have time for: In my 20 declaration in Paragraph 140 I'm basically saying that 21 it would be clear to a POSITA that the thread control 22 unit could not send data to the execution units or 23 PCUs if the data could not be stored in the general 24 purpose register block. And that statement is 25 definitely true. If the register block doesn't allow	112 1 wouldn't be able to send more data if the general 2 purpose register block is full. 3 Q. Okay. Does Lindholm say -- when I say 4 "Lindholm," I mean Exhibit 1005. Does Lindholm ever 5 say that the system performs vertex operations until 6 the general purpose register block does not have 7 enough available space to store incoming vertex data 8 specifically? 9 MR. JOHNSON: Objection, form. 10 A. So, you know, the paragraph we're looking at 11 right now, what I'm referring to is the thread control 12 unit would send data to the execution units; right? 13 So that's the performing vertex operations on the 14 vertex data. So if that cannot be done because it 15 can't be stored in the general purpose register block 16 that doesn't have enough available space. 17 Q. Is there any discussion either in your 18 opinions of what a POSITA would know or the reference 19 itself that would differentiate between having enough 20 available space to store incoming vertex data in 21 contrast with having enough available space to store 22 incoming pixel data? 23 MR. JOHNSON: Objection, form. 24 A. So I didn't consider that question because I 25 don't think that's required by the claim. And so to

113 1 answer your question, I don't know. I would have to 2 look through Lindholm, Exhibit 1005, and presumably 3 Exhibit 1006. But I think that question doesn't 4 relate to the claim. 5 Q. I understand you think it doesn't relate to 6 the claim. But is there any disclosure in either of 7 the Lindholm references of the system considering 8 there's enough room for pixel data but not enough room 9 for vertex data in the general purpose register block? 10 A. Yes, actually. In my declaration, Paragraph 11 140, I say including space available in the general 12 purpose register block such that vertex operations 13 could be given priority as long as there is sufficient 14 resources, including space in the general purpose 15 register block, for vertex processing, and, when there 16 is not, prioritizing pixel operations. 17 So Lindholm does say that, you know, as 18 long as there's space for the vertex operations, you 19 can continue with that, but otherwise prioritize pixel 20 operations. 21 Q. Well, hold on. What you quoted was you 22 saying that. I'm talking about Lindholm. Where does 23 Lindholm say that? 24 A. I'm referring back to claim limitations 3(a) 25 and 3(c). So let me look at those.	115 1 A. Yeah, I think I have the citations at the end 2 of that, in Paragraph 107. 3 Q. Okay. So where does Lindholm say that it 4 knows or determines that there's enough room for pixel 5 calculation data but not enough room for vertex data? 6 MR. JOHNSON: Objection, form. 7 A. I believe it's column 7, lines 6 to 36. Let 8 me see. 9 Q. Sorry, what was that, column 7 what? 10 A. So I'm actually looking at my report, 11 Paragraph 109, Page 58, where I quote Lindholm at, you 12 know, about the middle of that paragraph. 13 "Dynamically modifying a thread allocation priority 14 for pixel samples based on the amount of sample data," 15 et cetera. 16 All right, let me read Lindholm. 17 Okay, I found something here. Lindholm, 18 column 7 -- Exhibit 1005, column 7, "dynamically 19 modifying" -- line 11. "Dynamically modifying a 20 thread allocation priority for vertex samples based on 21 demand of sample data in vertex input buffer permits 22 vertex input buffer to drain faster and fill vertex 23 output buffer." 24 So here he seems to refer to, you know, 25 the amount of storage available.
114 1 So Lindholm talks about priorities for 2 processing. 3 Q. Where are you reading from? 4 A. Sorry, I'm at Page 57 of my declaration, 5 Paragraph 107. So the second sentence, "This priority 6 is determined for each sample type (vertex or pixel) 7 that may be fixed, programmable, or dynamic," and then 8 referring back to Lindholm. 9 So that priority may include -- and I'm 10 referring back to Page 56 in my declaration -- you 11 know, "it may be dynamic, based on the vertex or pixel 12 workload or the number of resources available, such as 13 'space available in the store,' i.e., the Register 14 File and the Vertex/Pixel Input Buffers." 15 Q. I'm sorry, where were you just reading from? 16 A. I was just reading from the end of Paragraph 17 106 in my declaration, last sentence. 18 Q. So again, this is all you saying this. 19 A. No, I'm referring to Lindholm, and I have 20 quotes here from Lindholm. 21 Q. You don't have any citations. 22 A. No, I do, on Page 57. 23 Q. I was talking about 56, that you just read. 24 There's no citation on Page 56, Paragraph 106, that 25 you were just pointing me to; right?	116 1 Q. Where is that? 2 A. Well, based on the amount of sample data in 3 vertex input buffer and fill vertex output buffer, and 4 I skipped a bunch of words in between. 5 Q. Right. So that's talking about how much 6 sample data you have in the buffer. The claim says 7 "operate on vertex operations until there's no more 8 room for incoming vertex data." How does what you 9 read relate to determining if there's enough space for 10 more vertex data? 11 A. Well, you know, it's a little hard to read 12 this language in Lindholm. But the way I read it is 13 that he's looking -- you know, based on the amount of 14 sample data in vertex input buffer permits vertex 15 input buffer to drain faster and fill vertex output 16 buffer faster or drain slower and fill vertex output 17 buffer slower. So it seems to monitor the amount of 18 storage available in vertex output buffer, and based 19 on that, the processing proceeds. 20 So it's certainly related to the storage 21 of vertex data. It's monitoring the storage of vertex 22 data to determine how fast the processing can proceed. 23 Q. Well, what it says is that you modify a 24 priority based on the amount of data in the input 25 buffer; right? One example of that is, if you have a

117 1 lot of vertex data to go through, you should give it a 2 higher priority because you have a lot of data. 3 Right? 4 MR. JOHNSON: Objection, form. 5 Q. I'm asking if that's -- 6 A. Are you quoting Lindholm? I'm not sure I'm 7 reading that here. 8 Q. I'm summarizing what you -- line 11 of column 9 7. When it says "dynamically modifying a thread 10 allocation priority for vertex samples based on the 11 amount of sample data in vertex buffer 220," so that 12 part. What that's saying is, you modify the priority 13 based on the amount of data in the input buffer; 14 right? 15 MR. JOHNSON: Objection, form. 16 A. Yeah, but the sentence continues, and, you 17 know, he's now also talking about monitoring -- or 18 basically, you know -- you know, drain faster and 19 fill -- sorry, I'm missing a word here. 20 Oh, yeah, fill pixel input buffer faster 21 or drain slower. Okay, the part I'm referring to is 22 that part, "fill vertex output buffer slower," so I'm 23 skipping a couple of words. 24 Q. That's the consequence of the modification of 25 the priority. That's different from what I'm saying.	119 1 says "dynamically modifying a thread allocation based 2 on the amount of sample data in the input buffer 3 permits the input buffer to drain faster and fill the 4 output buffer." It doesn't say monitor the output 5 buffer; right? 6 A. True, but, you know, coming back to the 7 claim: It's just requiring that some general purpose 8 register block does not have enough available space; 9 right? It doesn't actually say the term "monitor" -- 10 there is no monitoring here. It says if you don't 11 have space in your memory, you can't store the 12 incoming vertex data. 13 And so, you know, one part of that memory 14 is the vertex input buffer. If it doesn't have space, 15 you can't store it in there. 16 And so the thread allocation priority is 17 modified, yes, in this case based on the vertex input 18 buffer, but that's okay, because that meets the claim 19 language. 20 Q. But the information that it's based on is 21 already in the input buffer in Lindholm, and in the 22 claims it's an evaluation of is there room to store 23 unstored vertex data, whereas in Lindholm the vertex 24 data is already in the buffer, so there's no 25 evaluation of space. Do you understand what I'm
118 1 What I'm saying is what the decision to modify the 2 priority is based on. When it says what the dynamic 3 modification of the priority is based on, line 12, 4 it's based on the amount of sample data in the input 5 buffer; it's not based on anything about the output 6 buffer. Right? 7 MR. JOHNSON: Objection, form. 8 A. Well, here, if we go back to the claim -- 9 Q. Let's stay on Lindholm, because -- 10 A. I think it's -- go ahead. 11 Q. Let's figure out what it does in Lindholm; 12 right? We're on Paragraph 7. Before we get confused 13 about anything else: What Lindholm teaches modifying 14 the priority based on is the amount of data in the 15 input buffer or, a few lines down, the amount of data 16 in the pixel input buffer. Those are the two things 17 that modifying the priority is based on; correct? 18 MR. JOHNSON: Objection, form. 19 A. I think it's unfortunately not very clear 20 from his language here, because he also says, you 21 know, drain slower and fill pixel output buffer 22 slower. So obviously there is something that monitors 23 the pixel output buffer, too, I think. 24 Q. Well, it says that's a consequence of the 25 change. Where are you reading in a monitoring? It	120 1 saying? 2 MR. JOHNSON: Objection, form. 3 A. I actually did not quite understand what you 4 were saying, no. 5 Q. So the claims say perform vertex operations 6 until or unless the general purpose register block 7 does not have space to store incoming vertex data. 8 A. Correct. 9 Q. Meaning unstored vertex data. Right? 10 A. Yeah, yes, obviously. So there's some 11 unstored vertex data that it doesn't have space for. 12 Q. Right. And the consideration in Lindholm is 13 limited to how much data is already in the input 14 buffer, not whether more can be stored. Do you see 15 the distinction I'm drawing? 16 A. I don't see the distinction. I mean, what's 17 the difference between monitoring, you know, what's in 18 there and not having enough space? 19 Q. I think that -- so in Lindholm it's, hey, 20 there's a lot of data in this input buffer, we should 21 give that a high priority; and in the claims it's do I 22 have room for more vertex data in the buffer. Those 23 two things don't seem the same to me, and if they seem 24 the same to you, can you explain why they're the same? 25 MR. JOHNSON: Objection, form.

121 1 A. Well, Lindholm says it's modifying the 2 priority for vertex samples based on basically the 3 amount of data that's in the input buffer. So that 4 means, in my mind, it means that if the input buffer 5 is full, it can't store any more vertex data there. I 6 mean, why would it monitor it otherwise? 7 Q. Why would it monitor otherwise besides what? 8 A. Besides making sure that it has space to 9 store more vertex data. 10 Q. Well, I'm not here to testify, but if it 11 helps you move along the discussion, I would think one 12 obvious way is that it's not full but it has more 13 vertex work to do. Wouldn't that be a reason to 14 provide a priority to vertex data? 15 A. I don't think that's what this particular 16 section here reads to. I mean, I read it differently 17 than you do. 18 I also want to point out, you know, we 19 focused -- I found, you know, one particular instance 20 of this in Lindholm. You know, I'd be happy to look 21 for more. This is just one example. And so there 22 probably -- I know there are others, but Lindholm has 23 many, many sort of modifications of how to change the 24 thread priorities. 25 Q. Okay. Let's switch to the performing	123 1 helps you with the context. So in claim 11 that 2 phrase appears in the instruction store limitation. 3 It says "instruction store wherein the processor unit 4 performs the operations at various stages of 5 completion." So it's either processor unit or 6 instruction store. You don't actually say in your 7 report where you're finding that limitation met of 8 various degrees of completion. So you can point me to 9 wherever you want. Paragraph 114 is the only place I 10 found that talks about various stages of completion, 11 but feel free to look at your report. 12 A. Yeah, I think I'm referring back -- you're 13 correct. I'm referring back to the discussion of 14 claim 2, elements E, D, and 3, element A, C. So I 15 guess we can start where you pointed me, to claim 3. 16 So I think what I'm saying -- first of 17 all, what I'm seeing in my declaration, Page 60, 18 Paragraph 114 -- I'm saying Lindholm, you know, 19 switched between unfinished pixel threads and 20 unfinished vertex threads are disclosed in the 21 Lindholm patents. So that's one method in which you 22 can switch between vertex and pixel operations. 23 So I think here I'm taking the view that 24 you have vertex threads that are not finished and you 25 have pixel threads that are not finished, and so
122 1 operations at various degrees of completion concept. 2 And I think I can direct you to Page 60, Paragraph 3 114, of your report. And in this paragraph you see 4 that it ends, "This is one method through which the 5 Lindholm patents describe the ability to freely 6 alternate between vertex and pixel operations at 7 various stages of completion." Do you see that text? 8 A. Yes. 9 Q. And then you have some citations that I'd 10 like to look at and also some explanation. But 11 basically the question is -- because it's not clear to 12 me from your explanation why the switching is done at 13 various degrees of completion, could you please 14 explain to me your theory about why the switching of 15 instructions is done at various stages of completion 16 of the operations? 17 A. Sorry, which claim are we talking about right 18 now? 19 Q. 11. 20 A. This is, I think, we're in the section for 21 claim 3 in my -- 22 Q. Yeah, I think your discussion of 11 refers 23 back to claim 3. At least this is where I saw the 24 phrase "various stages of completion." But we can go 25 to claim 11 and then come back, if that makes you --	124 1 you're switching between unfinished pixel and vertex 2 threads. And that's the switching that the claim 3 refers to. 4 Q. And so here you just have some citations of, 5 you know, about four paragraphs of text. So maybe you 6 can help explain that. What are your theories of what 7 the required operations are, and what shows that 8 they're switched between at various stages of 9 completion? So maybe we can start with what are the 10 operations. 11 A. Well, I just, you know -- I say, "Generally 12 speaking, here are some examples." 13 I mean, you know, Lindholm -- maybe I'll 14 start at a high level. Lindholm talks about pixel and 15 vertex processing using multithreading. So there are 16 pixel threads and vertex threads. And each of these 17 are comprised of instructions. 18 So, you know, there could be some 19 operations happening from a pixel thread and some 20 operations happening from a vertex thread. 21 Now, you know, in Lindholm I'm taking the 22 view that the pixel operations are basically 23 unfinished in the pixel thread when you're switching 24 to a vertex thread. 25 So I don't know if that high-level --

125 1 Q. Well, it does, but earlier when we talked 2 about threads and instructions and which ones are the 3 operations of claim 11, you said it's not the threads, 4 it's the instructions that are the operations. Do you 5 have a different view now? 6 MR. JOHNSON: Objection, form. 7 A. Earlier we talked about a different 8 combination; right? And I forget if it was Amantides, 9 I believe, and Kohn where we had that discussion. Now 10 we're talking about Lindholm. So it's two different 11 references. 12 Q. Is there a difference between the threads of 13 Amantides and Kohn and the threads of Lindholm that 14 would impact whether the threads are considered 15 operations in claim 11? 16 A. You know, we can go back to the discussion of 17 Amantides. I have to refresh my memory. But I think 18 we looked very specifically at one example of, you 19 know, how an instruction could be not completed while 20 another instruction starts. That doesn't mean that's 21 the only way that operations on pixels and vertices 22 may -- sorry, what's the claim language? 23 Q. "Performed" I think is the word -- 24 A. May be switched -- or may be in degrees of 25 completion.	127 1 one or more samples using at least one program, where 2 each sample may be processed in an order that is 3 independent of the order in which the samples were 4 received." 5 So what he's describing is the general 6 concept of multithreading; right? So you can process 7 a sample while you're, you know, processing another 8 sample, and you can also switch from processing a 9 sample to processing another sample. 10 Q. Well, that's the part I don't see. So it 11 says multithreading and simultaneously process one or 12 more samples. Where does it say before finishing one 13 of the threads switch to a different thread? 14 A. I think that's not what the claim requires; 15 right? So it just -- the claim requires vertex and 16 pixel manipulation operations at various degrees of 17 completion based on switching between the 18 instructions. And I think that's what Lindholm 19 describes. 20 Q. So I think this is the same issue as the, I 21 think it was -- I don't remember if it was Kohn or 22 Fiske -- where we talked about when they're being 23 performed simultaneously, what's the switch? Do you 24 remember that discussion that we had earlier? 25 A. Was that the one where we talked about
126 1 Q. So for Amantides your theory is that the 2 threads are switched between at various stages of 3 completion, not the constituent instructions. Is that 4 right? 5 A. I think that's correct. I mean, that's one 6 way. I'm not saying that's the only way, but that's 7 one way, that the threads are not -- the pixel threads 8 are not completed, or the pixel thread, singular, is 9 not completed while a vertex thread starts. 10 Q. Okay. 11 A. And I would say that matches with, you know, 12 this broad term of vertex manipulation operations. 13 Q. So let's go through the subsequent paragraphs 14 in your report, which go up to, it looks like, 15 Paragraph 119. And can you help me understand how the 16 Lindholm '913 patent, which is what you point to, 17 switches at various stages of completion of the 18 operations? 19 A. So in Paragraph 117 I am quoting Lindholm 20 Exhibit 1006, and where Lindholm says, "Instruction 21 Scheduler determines which instructions can be 22 scheduled, choosing amongst instructions that process 23 different samples," keeping in mind that samples can 24 be vertex samples or pixel samples. "In this manner 25 multi (sic) Processing Unit can simultaneously process	128 1 fetching instructions -- 2 Q. Yes. 3 A. Yes. 4 Q. Is that your same theory here as to Lindholm, 5 that the instruction switch is the fetch of the 6 instruction? 7 MR. JOHNSON: Objection, form. 8 A. So the multiprocessor has multiple threads in 9 flight; right? And so on one hand there is multiple 10 instructions being executed simultaneously. 11 Q. Right. 12 A. And on the other hand, yes, it is true that 13 the unit could switch from, you know, executing one 14 instruction to executing another instruction. I think 15 that's what Lindholm says. 16 Q. And when you say "switch," you mean 17 instruction 1 is running; while continuing to run 18 instruction 1, go fetch instruction 2 and start also 19 processing instruction 2. Is that right? 20 A. Yeah, I think, you know, as I'm saying here 21 in Paragraph 119 -- 118, sorry, Page 62 of my 22 declaration, "Each execution unit may have multiple 23 threads functioning concurrently (at least at 24 different stages of its pipeline)." So I think that 25 is....

129 1 Q. Does Lindholm ever say something to the 2 effect of, "Execute instruction 1. At some point when 3 instruction 1 is not complete, stop processing 4 instruction 1 and instead process instruction 2?" 5 MR. JOHNSON: Objection, form. 6 A. I mean, first of all, I don't think that's 7 what the claim requires, that there is no, like, 8 requirement in the claim that there is sort of a, I 9 guess, monitor of some sort, as you seem to imply -- 10 because you seem to imply there is like something that 11 monitors when one instruction, you know, like, can 12 switch. 13 Q. Well, let's take that concept out, because 14 that's not really what I was trying to get at. 15 So whether or not it's monitored, my 16 question is: Is there something in Lindholm that 17 discusses executing on instruction 1; at some point 18 before that instruction is complete, ceasing to 19 perform that instruction and, instead of that, perform 20 instruction 2? 21 MR. JOHNSON: Objection, form. 22 A. Yeah, I think the sentence that I'm quoting, 23 again, on Page 62, on the top, "Each multithread 24 processing unit can simultaneously process one or more 25 samples." Right? So, you know, there is some, I	131 1 instructions can be scheduled, choosing amongst 2 instructions that process different samples." I think 3 that's the one I just quoted here; right? 4 Q. Yeah, I think so. 5 A. Again, I think that the question you posed -- 6 I'm struggling answering it because that's not one I 7 considered because it wasn't required by the claim. 8 So I think to answer your question I would have to 9 look through the Lindholm reference more carefully. 10 I would assume, just as a POSITA, and 11 knowing about multithread processing, that Lindholm 12 does discuss at some point, I guess, you know, as you 13 put it -- well, I forget which way you put it -- 14 either overlapping the execution of instructions or 15 finishing one instruction before the next. I'm 16 forgetting which one you're asking about. 17 Q. Okay. 18 MR. DOKHANCHY: I pass the witness to any 19 redirect. 20 MR. JOHNSON: Let's go off the record. 21 THE VIDEOGRAPHER: We are going off the 22 record. The time is 2:26 p.m. 23 (Recess taken.) 24 THE VIDEOGRAPHER: We're back on the 25 record. The time is 2:32 p.m.
130 1 think, overlapping of processing going on here. 2 That's the way I read it. 3 Q. Right, I think that's what I'm getting at: 4 Is there a discussion of a non-overlapping processing 5 in Lindholm, where in order to execute the second 6 instruction, you cease performing the first 7 instruction and you substitute it with instruction 2? 8 MR. JOHNSON: Objection, form. 9 A. I don't think that's what the claim requires. 10 Q. Okay. Well, regardless of that, does 11 Lindholm discuss it? 12 A. Could you repeat what your theory is, or the 13 question? 14 Q. My question is: Does Lindholm discuss 15 executing instruction 1 and, at some point before it 16 finishes, ceasing to execute instruction 1 and instead 17 executing instruction 2? 18 MR. JOHNSON: Objection, form. 19 A. So I'm going into Lindholm, Exhibit 2006, at 20 column 13, line 24. 21 Q. Okay. This is the part quoted in your 22 declaration? 23 A. Yeah, that's correct. I'm quoting that just 24 above Paragraph 118. So in that paragraph Lindholm 25 says, "Instruction scheduler determines which	132 1 EXAMINATION 2 BY MR. JOHNSON: 3 Q. Good afternoon, Dr. Pfister. 4 A. Good afternoon. 5 Q. Do you have Exhibit 1003 in front of you? 6 A. Yes. 7 Q. And that's your declaration; is that correct? 8 A. Yes, correct. 9 Q. And do you recall earlier today -- well, 10 first of all, turn to Page 125 of your declaration, 11 Paragraph 256. 12 A. Yes. 13 Q. So I'm just going to use this paragraph as an 14 example. But do you recall earlier today at times 15 counselor asked you for citations to various pieces of 16 prior art because they seemed to be ambiguous or a 17 typo in your report? 18 A. Yeah, I think we had at times struggled with 19 that, yes. 20 Q. And, for example, in Paragraph 256 he 21 questioned what the "ids." were referring to; is that 22 correct? 23 A. Yes. 24 Q. And so, for example, with respect to 25 Paragraph 256, do you agree with me that this is a

133 1 discussion about the Fiske reference? 2 MR. DOKHANCHY: Objection, form. 3 A. Yes. 4 MR. JOHNSON: What's your objection? 5 MR. DOKHANCHY: Form. 6 A. So we're in Paragraph 256. I'm discussing 7 Fiske. It's -- you know, the whole paragraph is 8 directed at Fiske. 9 Q. Okay. And do you also have a discussion of 10 that reference at other places in your report? 11 A. Yes. There's a whole section of my report 12 directed towards combinations with Fiske, yes. 13 Q. So if you turn to Page 105, for example, 14 starting on Paragraph 213, and going through Paragraph 15 220 on 109, are those paragraphs directed to your 16 opinions and summary of the Fiske reference? 17 A. Yes, specifically the section you're pointing 18 me to is my description of Fiske, in my understanding, 19 yes. So that whole section you quoted is directed 20 exclusively at Fiske. 21 Q. And so do you recall -- if you look at those 22 paragraphs, does that refresh your recollection as to 23 whether or not you cited to any portions of Fiske that 24 discuss the limitation of switching between 25 instructions?	135 1 CERTIFICATE OF COURT REPORTER 2 3 I, Alan H. Brock, Registered Diplomate 4 Reporter and Notary Public, the officer before whom 5 the foregoing deposition was taken, do hereby certify 6 that the foregoing transcript is a true and correct 7 record of the testimony given; that said testimony was 8 taken by me stenographically and thereafter reduced to 9 typewriting under my supervision; that reading and 10 signing was not requested; and that I am neither 11 counsel for nor related to nor employed by any of the 12 parties to this case and have no interest, financial 13 or otherwise, in its outcome. 14 IN WITNESS WHEREOF, I have hereunto set my 15 hand and affixed my notarial seal this 17th day of 16 February, 2024. 17 My commission expires March 25, 2027. 18 19 20  21 _____ 22 Alan H. Brock, RDR, CRR 23 24 25
134 1 A. Yeah, the whole -- a lot of the section here 2 discusses, you know, the different levels of thread 3 prioritization in Fiske, how Fiske can switch between 4 threads, and hence how different instructions of 5 different types can be executed at different times. 6 Q. Okay. So my question is just more 7 high-level: If I want to understand your opinions 8 related to Fiske, I would also need to read those 9 background sections. Correct? 10 MR. DOKHANCHY: Objection, form. 11 A. Yeah, I think at the time -- and, you know, I 12 don't remember if it was in this context, but at times 13 I did refer back to my earlier -- my description of, 14 you know, certain prior art. And I may have said in 15 this context that, yes, I also discuss Fiske in a more 16 high level at the beginning of that whole section. So 17 I think I did mention that at some points. 18 MR. JOHNSON: I don't have any other 19 questions. 20 MR. DOKHANCHY: Nothing from me. We can 21 wrap it up. 22 THE VIDEOGRAPHER: This marks the end of 23 the deposition of Dr. Hanspeter Pfister. We're going 24 off the record at 2:38 p.m. 25 (The deposition concluded at 2:38 p.m.)	

A			
ability	56:4, 57:2,	afterwards	50:14, 51:5,
122:5	58:21, 63:13,	53:24	53:10, 55:23,
able	64:5, 66:14,	again	96:18
31:22, 39:4,	68:5, 68:17,	12:6, 22:18,	allegations
66:12, 85:12,	69:23, 70:11,	26:2, 26:3,	11:7, 28:18,
87:11, 95:15,	72:22, 78:9,	29:15, 32:23,	62:8, 84:13,
112:1	80:13, 82:4,	32:24, 36:6,	110:12
above	82:25, 83:13,	41:13, 44:25,	allege
8:12, 130:24	85:21, 86:13,	46:5, 46:19,	26:9
access	86:21, 87:19,	47:12, 48:3,	allocation
95:16	90:8, 93:25,	51:10, 56:1,	115:13, 115:20,
accesses	97:4, 97:24,	58:19, 61:20,	117:10, 119:1,
101:23, 102:23,	98:9, 99:23,	64:4, 69:5,	119:16
103:3, 106:9,	100:8, 107:11,	70:5, 71:21,	allow
107:7	109:20, 109:22,	72:13, 73:2,	110:25
achieving	111:6, 113:10,	76:5, 76:14,	allowed
68:15	115:10, 119:9,	78:2, 78:20,	45:23, 46:8
across	120:3, 123:6	78:24, 82:23,	allows
92:19, 92:22	add	87:15, 87:18,	81:13
action	89:16, 89:18,	88:20, 91:10,	alone
81:25	97:8, 97:24	93:9, 93:14,	31:19
actions	adder	94:12, 114:18,	along
44:3	66:1	129:23, 131:5	121:11
activities	adding	agree	already
105:8, 105:9	65:24, 89:16	132:25	37:9, 92:25,
actual	addition	ahead	119:21, 119:24,
6:25, 22:22,	60:17, 60:18,	118:10	120:13
26:17, 58:23	60:19, 64:9	alan	also
actually	additional	1:23, 5:1,	2:19, 8:21,
8:15, 9:23,	45:25	135:3, 135:22	9:20, 9:23,
10:3, 10:8,	addressing	all	13:18, 14:18,
10:11, 11:11,	5:23	10:2, 10:3,	15:24, 16:2,
11:20, 13:5,	advantage	12:11, 16:7,	19:2, 20:6,
14:6, 14:12,	63:19, 64:3,	25:2, 29:10,	20:20, 21:16,
14:23, 15:17,	67:19, 76:16,	33:21, 35:7,	22:7, 23:8,
16:22, 17:6,	78:9, 78:13,	38:20, 43:10,	26:1, 28:13,
18:3, 20:9,	78:16	44:18, 44:22,	28:14, 29:9,
20:22, 21:6,	affirmed	45:6, 45:10,	30:2, 31:3,
23:14, 24:10,	5:4	45:18, 47:2,	45:2, 45:9,
24:11, 24:21,	affixed	47:9, 48:12,	51:15, 57:22,
26:19, 31:16,	135:15	48:14, 59:8,	60:24, 71:19,
32:16, 33:3,	after	78:25, 83:25,	81:17, 88:8,
36:3, 37:8,	8:2, 57:7,	87:15, 94:1,	91:3, 91:16,
38:4, 38:6,	57:12, 60:7,	96:5, 105:4,	94:16, 99:18,
38:17, 42:11,	61:17, 63:14,	108:6, 114:18,	100:9, 104:9,
42:19, 52:19,	77:17	115:16, 123:17,	105:10, 117:17,
	afternoon	129:6, 132:10	118:20, 121:18,
	132:3, 132:4	allegation	122:10, 127:8,
		6:3, 6:7,	

128:18, 133:9, 134:8, 134:15 alternate 122:6 alternative 14:8 alu 53:22 always 36:14, 77:11 ambiguity 24:22 ambiguous 8:14, 132:16 and 4:18 among 22:5, 53:3, 84:18, 85:14, 86:3, 86:14, 101:17 amongst 126:22, 131:1 amount 115:14, 115:25, 116:2, 116:13, 116:17, 116:24, 117:11, 117:13, 118:4, 118:14, 118:15, 119:2, 121:3 analyses 44:2 analysis 5:17, 47:25, 51:9, 108:18 another 15:2, 22:14, 22:24, 26:6, 44:17, 59:25, 60:13, 65:8, 69:12, 74:18, 75:22, 88:7, 88:24, 91:9, 110:1, 125:20, 127:7, 127:9, 128:14 answer 15:6, 18:1,	18:6, 22:21, 23:12, 28:11, 32:20, 56:4, 56:5, 58:6, 85:6, 87:21, 89:6, 91:22, 96:16, 99:25, 113:1, 131:8 answering 131:6 answers 22:20, 36:16 any 34:14, 36:1, 37:15, 39:4, 44:2, 44:9, 46:13, 47:25, 55:23, 56:8, 63:8, 74:11, 83:9, 85:1, 89:12, 92:10, 92:12, 94:14, 94:19, 96:3, 96:4, 96:5, 97:10, 111:6, 112:17, 113:6, 114:21, 121:5, 131:18, 133:23, 134:18, 135:11 anybody 46:7, 46:11 anything 9:12, 27:19, 66:2, 66:6, 76:23, 77:5, 77:22, 85:7, 95:2, 100:7, 100:21, 101:4, 101:11, 106:17, 110:4, 111:16, 118:5, 118:13 anywhere 53:8, 55:7 apologize 87:17, 87:20 apparently 107:20 appeal 1:2	appearance 50:9 appears 8:18, 42:3, 43:7, 43:25, 44:6, 45:4, 46:13, 123:2 appropriate 45:7, 105:16 approximately 65:2 arbiter 80:7, 80:14, 80:15, 80:19, 81:15, 81:23, 81:25, 82:17 arbitration 80:24, 81:7, 81:15, 81:16, 81:20, 82:9, 82:12, 82:22, 83:6 argument's 70:15 around 8:1, 26:14, 39:11, 39:16, 62:11, 108:8 arrive 37:13, 90:8 arrives 90:11 art 5:19, 62:21, 64:4, 68:4, 70:6, 74:6, 76:15, 79:15, 88:11, 88:20, 92:8, 93:10, 96:4, 132:16, 134:14 article 53:12 asked 132:15 asking 22:12, 23:2, 79:13, 91:6,	93:21, 97:9, 97:10, 109:1, 117:5, 131:16 aspect 73:8 assertion 98:14, 98:17 assigning 80:22, 82:11 associate 54:23, 55:5, 55:7, 55:12 associated 25:25, 26:20, 110:11 assume 18:20, 24:24, 29:25, 77:16, 77:18, 85:8, 88:14, 88:21, 131:10 assumes 16:2, 38:2 assuming 18:22, 19:7, 19:20, 21:7, 23:13, 26:3, 43:10, 104:9, 110:19 assumption 61:9 ati 1:7, 4:5, 4:18 authors 8:6, 33:8, 33:16, 37:3, 39:16, 40:16 automatically 101:24, 102:11, 102:24, 103:4, 103:6, 103:18, 103:24, 104:1, 104:3, 107:9, 107:13, 107:22 available 108:11, 110:15, 111:2, 111:14, 112:7, 112:16,
--	---	--	--

112:20, 112:21, 113:11, 114:12, 114:13, 115:25, 116:18, 119:8 avoid 33:16 aware 48:14, 99:5 B b 71:10 back 10:5, 13:12, 20:13, 25:6, 28:16, 29:13, 35:13, 50:18, 53:25, 59:6, 60:19, 61:2, 63:20, 64:9, 64:13, 64:25, 65:5, 74:5, 76:9, 78:19, 83:14, 83:23, 85:3, 89:9, 91:23, 98:4, 105:4, 108:4, 109:22, 110:18, 113:24, 114:8, 114:10, 118:8, 119:6, 122:23, 122:25, 123:12, 123:13, 125:16, 131:24, 134:13 background 134:9 baker 2:3, 4:19 based 42:8, 59:13, 86:1, 110:1, 114:11, 115:14, 115:20, 116:2, 116:13, 116:18, 116:24, 117:10, 117:13, 118:2, 118:3, 118:4, 118:5, 118:14,	118:17, 119:1, 119:17, 119:20, 121:2, 127:17 basically 7:1, 9:19, 10:6, 24:18, 26:14, 33:20, 34:10, 36:11, 53:15, 56:11, 56:19, 67:3, 80:20, 81:8, 92:10, 94:10, 94:12, 106:10, 110:20, 117:18, 121:2, 122:11, 124:22 bates 7:16 because 7:7, 8:17, 9:8, 9:23, 9:25, 11:20, 14:19, 16:11, 18:2, 19:4, 19:18, 21:24, 22:6, 23:13, 24:6, 38:1, 38:12, 46:7, 61:19, 63:7, 68:5, 73:12, 74:14, 77:10, 86:21, 88:21, 89:1, 91:11, 93:11, 93:14, 94:1, 95:6, 96:3, 96:9, 97:17, 104:10, 106:21, 107:21, 112:14, 112:24, 117:2, 118:9, 118:20, 119:18, 122:11, 129:10, 129:13, 131:6, 131:7, 132:16 been 14:9, 38:2, 38:16, 68:19, 78:21	before 1:2, 10:10, 14:2, 16:5, 27:21, 32:20, 35:10, 38:10, 45:8, 48:21, 56:3, 57:7, 60:1, 62:23, 64:13, 77:7, 77:15, 77:24, 78:21, 85:21, 91:25, 92:19, 92:25, 96:11, 118:12, 127:12, 129:18, 130:15, 131:15, 135:4 begin 5:17, 44:24, 48:2, 75:5 beginning 25:23, 41:11, 134:16 begins 4:3, 5:16, 40:9, 61:8, 108:18 behalf 2:2, 2:10, 4:18, 4:20 behind 81:16 being 5:4, 30:5, 30:12, 34:9, 36:2, 37:8, 41:25, 43:25, 52:25, 56:24, 62:12, 63:2, 64:13, 65:10, 66:23, 68:10, 69:22, 70:24, 73:10, 77:12, 77:20, 78:12, 79:2, 84:16, 86:24, 90:21, 90:23, 90:24, 90:25, 91:2, 91:9, 97:6,	111:18, 127:22, 128:10 belief 28:19 believe 36:14, 72:7, 97:14, 97:17, 101:13, 105:12, 115:7, 125:9 below 53:19 berkeley 4:13 besides 9:13, 9:25, 36:1, 121:7, 121:8 best 66:15 between 24:25, 34:15, 36:8, 36:12, 45:2, 48:24, 49:20, 59:14, 61:17, 64:19, 64:23, 65:18, 66:16, 69:25, 71:5, 72:1, 72:11, 72:19, 74:13, 86:1, 86:5, 91:15, 112:19, 116:4, 120:17, 122:6, 123:19, 123:22, 124:1, 124:8, 125:12, 126:2, 127:17, 133:24, 134:3 beyond 29:14, 29:16 bit 9:24, 19:17, 45:2, 54:6, 54:22, 56:2, 69:16, 73:1, 79:1, 79:3, 109:18 block 16:14, 16:17,
--	--	--	---

16:20, 16:24, 17:7, 17:18, 18:24, 20:12, 24:10, 25:10, 25:15, 25:21, 26:8, 26:11, 28:15, 28:20, 29:8, 29:10, 29:11, 48:21, 79:8, 79:20, 79:23, 79:24, 91:25, 92:1, 92:5, 92:13, 93:1, 93:2, 93:7, 93:9, 93:14, 93:20, 94:2, 94:4, 94:8, 94:19, 95:5, 95:9, 95:12, 95:16, 95:24, 96:15, 96:20, 98:12, 98:20, 99:2, 99:7, 108:11, 110:15, 110:24, 110:25, 111:14, 111:24, 112:2, 112:6, 112:15, 113:9, 113:12, 113:15, 119:8, 120:6 blocks 22:10, 22:12, 29:2, 65:25, 68:14, 78:15, 92:19, 92:22, 96:24, 98:2 board 1:2 boston 1:13, 4:13 both 11:12, 21:4, 27:25, 63:18, 67:10, 67:17, 69:4, 71:13, 71:20, 71:24, 73:12, 96:23,	99:4, 100:17 bottom 7:18, 11:15, 16:23, 33:3, 33:4, 50:21, 91:24, 109:10 botts 2:3, 4:20 boundaries 61:8 boundary 61:12, 61:13, 96:9 break 59:2, 83:19, 107:24 bring 8:22 broad 92:15, 126:12 broadest 67:23, 93:16, 94:13 brock 1:23, 5:1, 135:3, 135:22 buffer 109:11, 109:12, 115:21, 115:22, 115:23, 116:3, 116:6, 116:14, 116:15, 116:16, 116:17, 116:18, 116:25, 117:11, 117:13, 117:20, 117:22, 118:5, 118:6, 118:15, 118:16, 118:21, 118:23, 119:2, 119:3, 119:4, 119:5, 119:14, 119:18, 119:21, 119:24, 120:14, 120:20, 120:22, 121:3, 121:4 buffers 39:20, 114:14 builds 80:6	bunch 116:4 bus 26:9, 26:13, 26:17, 26:20, 26:21, 28:20, 30:11, 80:23, 80:25, 81:4, 81:9, 81:10, 82:1, 82:7, 82:12, 82:18, 83:5, 102:23, 103:3, 106:10, 107:12 buses 26:18 C cache 11:16, 11:23, 13:13, 13:14, 13:25, 14:1, 14:3, 14:19, 16:4, 20:1, 20:3, 20:6, 20:15, 20:17, 20:20, 21:1, 21:13, 21:15, 21:21, 22:4, 22:6, 23:3, 23:8, 23:14, 26:2, 26:4, 26:5, 26:6, 28:21 calculate 89:10 calculation 16:19, 49:10, 49:19, 49:22, 50:9, 50:10, 50:14, 50:15, 52:6, 53:11, 115:5 calculations 24:7, 24:13, 101:19, 102:6 california 2:15	call 7:7, 56:12, 57:13, 59:24, 68:8, 81:16 called 97:21 calling 7:4, 13:16, 29:11, 56:11, 65:23, 99:1 calls 30:7, 31:3, 35:4, 52:4, 81:15 can't 32:12, 56:4, 87:16, 111:6, 111:16, 111:23, 112:15, 119:11, 119:15, 121:5 candid 50:4 cannot 18:1, 81:3, 92:13, 112:14 capability 63:20, 64:3, 67:19, 76:16, 78:10 capable 71:22, 92:17 capacity 69:4, 71:20 carefully 15:19, 32:24, 56:3, 131:9 carmel 2:14 carried 51:1 case 1:10, 4:7, 52:25, 65:6, 65:15, 69:7, 85:9, 88:25, 95:1, 96:3, 103:5, 106:21, 107:5, 111:10,
---	---	---	--

119:17, 135:12 cause 44:3, 101:18, 102:5 causes 43:5, 46:25, 48:1 cease 130:6 ceasing 111:11, 129:18, 130:16 certain 134:14 certainly 26:17, 30:24, 31:18, 35:1, 58:15, 69:6, 69:19, 71:22, 90:24, 116:20 certificate 135:1 certify 135:5 cetera 13:20, 88:5, 91:4, 98:2, 115:15 chance 81:4 change 10:6, 41:20, 42:19, 60:5, 118:25, 121:23 changes 71:25 check 7:9, 8:9, 10:9, 30:15, 41:12, 47:25, 55:10, 56:6 chip 90:7, 94:16 chooses 88:3 choosing 126:22, 131:1 chose 39:16	circuit 80:2, 81:24, 82:4, 82:21, 82:22, 99:15 circuitry 11:17, 11:25, 80:24, 81:13, 81:14, 81:17, 103:10, 103:12, 103:14 circuits 81:21, 83:4, 96:12 citation 12:7, 12:11, 12:14, 13:2, 85:11, 102:14, 102:16, 114:24 citations 12:22, 114:21, 115:1, 122:9, 124:4, 132:15 cite 12:3, 12:4, 63:6, 101:21 cited 13:6, 107:9, 133:23 citing 107:19, 107:20, 109:12 claimed 5:23, 10:14, 10:19, 11:8, 11:13, 11:17, 11:24, 20:6, 48:6, 50:25, 92:1, 96:19, 100:18, 108:24 claims 29:11, 48:12, 48:14, 49:5, 79:25, 80:12, 85:4, 87:10, 109:4, 119:22, 120:5, 120:21 classic 92:24	clear 6:23, 12:12, 13:22, 13:23, 16:1, 21:2, 23:6, 36:4, 36:23, 40:2, 50:3, 54:18, 59:19, 61:21, 66:24, 101:13, 110:21, 111:5, 118:19, 122:11 clearly 9:7, 66:22, 91:19, 107:19, 110:3 clipped 35:19 closest 50:3, 50:15 coextensive 8:24 cohn 2:11 collection 6:13 collectively 29:12 color 58:14, 79:7 column 7:18, 79:5, 91:23, 108:8, 115:7, 115:9, 115:18, 117:8, 130:20 com 2:8, 2:17 combination 5:20, 29:2, 94:19, 100:19, 100:20, 101:1, 125:8 combinations 133:12 combine 100:10 combined 98:19	combining 15:22, 29:10, 100:8, 100:12 come 30:23, 35:9, 35:18, 35:23, 36:1, 56:13, 61:2, 63:3, 63:4, 77:17, 92:19, 92:22, 106:23, 122:25 coming 64:25, 76:14, 119:6 commission 135:17 common 73:2 communication 105:6 complete 76:25, 88:22, 89:1, 89:4, 129:3, 129:18 completed 60:22, 61:1, 61:18, 61:24, 64:7, 71:8, 72:22, 77:19, 84:20, 85:15, 86:4, 86:17, 88:23, 125:19, 126:8, 126:9 completely 62:1 completion 59:13, 59:16, 60:4, 60:7, 60:8, 60:25, 61:20, 61:25, 62:8, 64:18, 64:21, 64:24, 65:12, 65:20, 67:4, 68:22, 69:9, 70:20, 70:21, 71:14, 72:16, 72:20, 73:14, 73:16,
--	--	--	---

76:13, 76:21, 77:8, 84:2, 84:14, 86:1, 86:19, 87:1, 87:13, 88:16, 89:3, 90:14, 122:1, 122:7, 122:13, 122:15, 122:24, 123:5, 123:8, 123:10, 124:9, 125:25, 126:3, 126:17, 127:17 completion" 60:12 complicated 68:7 component 17:2 components 9:6, 23:3, 28:19, 29:10 comprise 11:16, 11:24 comprised 124:17 comprises 26:7 comprising 16:24 computer 53:13, 95:22 concept 35:15, 82:6, 90:20, 122:1, 127:6, 129:13 concluded 134:25 concurrently 128:23 condition 45:25, 48:1 conditions 43:15, 43:17, 45:23 confused 19:6, 19:19, 118:12	confusing 68:2 confusion 73:22 connected 106:22 connection 95:21 consequence 117:24, 118:24 consider 9:6, 33:20, 57:21, 58:2, 93:5, 93:19, 94:1, 95:22, 112:24 consideration 120:12 considered 17:11, 28:14, 35:10, 56:3, 95:4, 96:2, 125:14, 131:7 considering 14:3, 96:7, 109:4, 109:15, 110:5, 113:7 consistent 79:1 constituent 126:3 consumed 34:8 contain 32:9, 32:22, 58:25 contains 58:24 contend 28:9 context 62:18, 78:20, 78:24, 87:24, 87:25, 88:3, 88:7, 88:15, 88:18, 88:21, 88:24, 89:5, 89:21, 90:10,	90:15, 90:20, 91:7, 91:12, 91:15, 91:16, 102:12, 123:1, 134:12, 134:15 continue 6:2, 113:19 continues 27:24, 117:16 continuing 75:6, 128:17 contrast 112:21 control 80:3, 80:8, 109:10, 110:21, 111:25, 112:11 convert 34:1, 34:22 converted 35:2, 35:8, 35:24, 36:2, 56:20, 56:22, 56:24, 57:1 converts 34:16, 35:5 coordinate 110:2 copies 5:11 copy 63:5, 85:19 corner 9:25, 33:4 corp 1:4, 4:5 correct 5:13, 5:14, 5:25, 6:6, 6:12, 6:14, 6:15, 6:21, 7:21, 7:25, 8:5, 11:4, 11:8, 11:18, 12:10, 12:24, 13:1, 15:12, 17:1, 19:12, 19:22, 20:18, 24:15, 25:16,	28:17, 28:23, 28:25, 35:21, 35:25, 37:6, 38:15, 40:19, 41:9, 41:19, 42:9, 43:9, 46:16, 47:15, 47:20, 48:7, 48:12, 48:16, 51:4, 52:8, 52:16, 55:19, 62:16, 68:11, 76:3, 80:9, 81:21, 81:22, 92:9, 101:7, 101:19, 101:20, 103:4, 103:24, 106:5, 107:9, 108:21, 111:20, 118:17, 120:8, 123:13, 126:5, 130:23, 132:7, 132:8, 132:22, 134:9, 135:6 correctly 12:1, 12:2, 28:18, 91:21 correspondence 48:24 could 5:15, 7:8, 10:2, 12:21, 14:8, 16:22, 17:9, 19:3, 24:24, 28:14, 28:19, 29:25, 32:14, 33:25, 38:3, 38:16, 52:15, 53:4, 54:8, 57:3, 63:19, 64:8, 64:12, 65:9, 67:5, 67:6, 67:18, 69:21, 76:6, 76:9, 76:11, 78:3, 78:5, 80:14, 83:3, 84:5,
---	--	--	--

85:16, 89:22, 90:5, 91:3, 91:15, 91:16, 94:18, 96:17, 108:6, 109:13, 109:17, 110:10, 110:22, 110:23, 113:13, 122:13, 124:18, 125:19, 128:13, 130:12 couldn't 61:4, 88:22, 89:1 counsel 4:15, 135:11 counselor 132:15 couple 108:25, 117:23 coupled 16:17, 95:7, 95:14, 95:17, 95:18, 95:19, 96:12 court 4:21, 4:25, 135:1 created 57:9, 57:12 creates 57:11 creation 57:7 crr 1:23, 135:22 current 40:20 currently 38:18 cursor 29:18, 29:22 cut 91:6 cycles 87:25, 88:2, 88:4, 88:8 D data" 16:25	date 4:8 day 87:15, 135:15 deadlock 33:5, 33:9, 33:15, 33:19, 34:12, 36:5, 36:11, 38:1, 38:4, 39:12, 39:13, 39:17, 40:15, 41:6, 41:15, 54:7 decided 8:3 decides 83:5 decision 42:6, 42:8, 118:1 declaration 5:13, 6:24, 10:6, 10:9, 10:24, 13:11, 13:19, 18:1, 19:25, 25:23, 27:6, 31:8, 56:2, 66:7, 69:18, 71:19, 80:11, 80:21, 94:25, 96:6, 99:4, 110:20, 113:10, 114:4, 114:10, 114:17, 123:17, 128:22, 130:22, 132:7, 132:10 decode 74:9 defendants 4:20 define 67:22 defined 61:12, 61:13, 91:19 definite 61:7	definitely 110:25 definition 60:23 definitions 30:16 degree 60:24, 61:24, 61:25, 65:12, 65:20, 67:3, 70:19, 70:21, 72:16, 72:20, 76:21, 89:3, 90:14 degrees 59:13, 59:16, 60:4, 61:20, 64:18, 64:23, 68:22, 69:9, 71:13, 73:13, 84:2, 84:14, 85:25, 88:16, 97:16, 122:1, 122:13, 123:8, 125:24, 127:16 deliver 44:24 delivered 42:4, 44:1, 44:7, 45:5, 45:7, 45:11, 46:14 delivering 48:2, 105:16 demand 115:21 dependent 80:12 depends 18:2, 26:16, 61:3 depicts 16:18 depos 4:11, 5:1 deposition 1:12, 3:8, 4:4, 4:11, 134:23,	134:25, 135:5 describe 23:13, 48:5, 53:15, 69:2, 69:17, 83:9, 84:13, 101:11, 122:5 described 37:1, 44:6, 44:20, 45:2, 61:23, 66:5, 68:14, 68:21, 76:18, 81:21 describes 7:19, 36:10, 45:17, 53:14, 63:16, 67:15, 68:25, 69:2, 80:17, 84:16, 85:14, 86:14, 86:25, 88:6, 97:20, 127:19 describing 40:3, 41:21, 44:15, 81:1, 81:8, 127:5 description 101:12, 102:10, 105:11, 133:18, 134:13 design 14:9, 27:6, 39:13, 39:16 designs 39:11 detail 69:16, 78:22, 91:22 detailed 78:23 details 21:18, 23:12 detected 103:14, 103:19 determine 116:22 determined 114:6
--	--	---	---

determines 82:22, 115:4, 126:21, 130:25 determining 116:9 device 37:20 diagram 6:25, 8:18, 9:9 diego 2:15 difference 24:25, 120:17, 125:12 different 24:11, 30:12, 36:22, 37:25, 40:18, 48:18, 51:21, 52:19, 53:15, 54:21, 58:14, 62:12, 62:24, 64:17, 65:25, 68:5, 68:15, 68:21, 69:13, 69:21, 74:16, 75:19, 78:14, 78:15, 78:19, 81:10, 82:14, 84:18, 85:14, 86:14, 88:9, 89:5, 96:10, 98:8, 117:25, 125:5, 125:7, 125:10, 126:23, 127:13, 128:24, 131:2, 134:2, 134:4, 134:5 differentiate 112:19 differently 91:12, 121:16 dig 85:21 diplomate 135:3 direct 11:10, 48:8,	122:2 directed 15:18, 17:16, 36:15, 53:13, 133:8, 133:12, 133:15, 133:19 directing 15:5, 37:25 disclose 11:7, 11:12, 14:23, 63:22 disclosed 36:18, 98:18, 123:20 discloses 6:8, 12:22, 36:7, 51:15, 53:10, 63:11, 84:23, 87:11 disclosure 13:3, 39:3, 39:4, 113:6 disconnect 74:19 discuss 11:6, 111:8, 130:11, 130:14, 131:12, 133:24, 134:15 discussed 15:11, 27:16, 109:21 discusses 30:6, 129:17, 134:2 discussing 33:8, 133:6 discussion 45:3, 54:7, 85:3, 112:17, 121:11, 122:22, 123:13, 125:9, 125:16, 127:24, 130:4, 133:1, 133:9 disk 96:4, 96:6 disparate 94:4	distinction 9:22, 24:20, 52:22, 72:8, 91:14, 120:15, 120:16 doing 71:23, 71:24, 90:15 dokhanchy 2:13, 3:4, 4:17, 5:8, 22:16, 22:21, 22:25, 59:1, 83:18, 107:24, 131:18, 133:2, 133:5, 134:10, 134:20 done 58:21, 60:14, 82:15, 84:16, 103:23, 112:14, 122:12, 122:15 dot 12:17 down 16:18, 29:24, 36:19, 39:18, 40:18, 41:1, 41:8, 41:16, 42:14, 42:22, 43:6, 44:18, 62:24, 64:6, 68:6, 68:22, 70:19, 78:22, 91:21, 118:15 dr 4:4, 5:9, 23:2, 59:8, 83:25, 108:6, 132:3, 134:23 drain 88:1, 115:22, 116:15, 116:16, 117:18, 117:21, 118:21, 119:3 dram 14:4, 14:8, 14:10, 14:13,	14:16, 15:1, 15:3, 15:12, 15:15, 15:25, 16:3, 16:11, 18:16, 18:21, 19:5, 19:7, 19:20, 25:25, 26:24, 26:25, 27:4, 27:10, 27:11, 27:17, 27:20, 28:9, 28:20, 29:8, 29:15, 29:17, 30:2, 30:5, 90:7, 93:12, 93:13, 93:20, 94:17 draw 96:8, 96:13 drawing 120:15 dsp 8:3, 98:19, 98:22 dsps 98:8 dual 63:24, 69:1 dual-operation 63:17, 65:2, 66:3, 66:10, 66:21, 67:16, 68:12, 69:14, 70:16, 73:4 duly 5:4 duplicating 88:4, 88:8 during 38:6, 39:17, 40:7, 42:2, 89:21, 110:9 dynamic 114:7, 114:11, 118:2 dynamically 115:13, 115:18, 115:19, 117:9,
---	---	--	---

119:1	28:5, 28:6, 35:9, 46:11, 46:18, 51:7, 51:11, 53:5, 53:6, 54:12, 61:22, 64:11, 90:16, 100:7, 101:12, 105:3, 110:5, 118:13	entitled 33:4 entity 82:20 equal 56:11 equate 48:23 equivalent 56:18 esq 2:4, 2:13 essentially 91:7 et 13:20, 88:5, 91:4, 98:2, 115:15 evaluating 111:9, 111:13, 111:15 evaluation 119:22, 119:25 even 15:17, 18:3, 32:15, 88:11, 91:14, 99:14 event 38:10, 41:21, 42:9, 42:12, 42:14, 42:16, 42:17, 42:21, 42:23, 43:2, 43:4, 43:5, 43:8, 82:18 events 44:2 eventually 45:5, 45:18, 46:4, 56:20, 56:22 ever 99:19, 112:4, 129:1 every 29:19, 33:12, 38:19, 40:4, 40:13, 42:20,	43:16, 91:11 evidence 76:11, 111:18, 111:22 exactly 46:1, 58:4, 70:5, 91:22 examination 3:3, 5:7, 132:1 examined 5:6 example 23:22, 28:4, 32:14, 33:19, 35:2, 35:3, 35:4, 35:5, 36:24, 49:7, 53:3, 53:18, 57:5, 57:18, 57:19, 57:25, 58:8, 58:14, 58:16, 58:17, 58:24, 59:22, 60:13, 60:20, 64:12, 65:21, 65:24, 66:19, 66:20, 67:8, 69:20, 70:23, 71:15, 76:4, 89:10, 89:14, 89:23, 92:21, 93:3, 98:15, 102:17, 102:18, 102:25, 109:23, 116:25, 121:21, 125:18, 132:14, 132:20, 132:24, 133:13 examples 124:12 except 8:20 exclusively 106:6, 133:20 excuse 10:20, 86:18 execute 65:14, 72:15,
E	embodiment 82:25, 83:8, 83:9 embodiments 76:7 employed 135:11 empty 44:22, 47:2, 47:13, 47:16 end 11:15, 13:10, 50:24, 53:18, 53:20, 53:25, 69:3, 114:16, 115:1, 134:22 ended 106:8 ends 26:22, 61:9, 122:4 english 29:5 enough 16:7, 42:18, 108:11, 110:15, 111:2, 111:14, 112:7, 112:16, 112:19, 112:21, 113:8, 115:4, 115:5, 116:9, 119:8, 120:18 ensure 77:23, 78:6 ensuring 82:1, 82:6 entire 84:19, 85:15, 86:4, 86:16		
each 7:24, 10:14, 15:20, 17:25, 26:1, 27:7, 34:16, 34:24, 35:5, 77:17, 94:20, 95:13, 96:12, 97:25, 109:24, 114:6, 124:16, 127:2, 128:22, 129:23 earlier 8:17, 26:25, 27:16, 28:13, 29:23, 31:3, 37:10, 49:17, 52:2, 58:24, 67:20, 68:21, 72:24, 76:18, 78:25, 79:2, 99:17, 109:19, 109:21, 125:1, 125:7, 127:24, 132:9, 132:14, 134:13 easy 62:25 effect 129:2 either 6:12, 6:17, 52:14, 59:25, 82:13, 109:17, 112:17, 113:6, 123:5, 131:14 electric 96:11, 99:15 element 25:2, 25:10, 50:8, 123:14 elements 123:14 eliminate 14:7, 14:13 else 9:1, 9:12,			

73:4, 73:5, 74:7, 88:2, 88:3, 90:2, 90:5, 101:19, 102:6, 129:2, 130:5, 130:16 executed 73:10, 77:10, 77:15, 78:12, 128:10, 134:5 executes 79:6 executing 51:1, 52:6, 53:10, 63:11, 70:18, 72:4, 73:13, 74:18, 75:25, 76:19, 89:21, 90:14, 90:16, 128:13, 128:14, 129:17, 130:15, 130:17 execution 62:13, 62:15, 63:12, 63:16, 63:23, 67:15, 70:11, 73:11, 74:9, 77:7, 77:25, 109:25, 110:22, 112:12, 128:22, 131:14 exhibit 5:11, 5:12, 7:10, 11:21, 21:7, 26:3, 59:9, 63:5, 85:19, 97:18, 109:12, 109:14, 110:7, 112:4, 113:2, 113:3, 115:18, 126:20, 130:19, 132:5 exhibits 3:8, 109:6 expect 50:13 experience 92:18, 92:23	expert 92:18 expires 135:17 explain 46:17, 62:18, 70:23, 120:24, 122:14, 124:6 explained 72:3, 76:4 explaining 45:23, 57:16 explanation 21:25, 63:8, 78:23, 122:10, 122:12 explicitly 98:18, 99:8, 106:18 exploring 8:2 F face 29:20 fact 15:22, 70:2 fair 39:5, 39:6, 55:2, 67:22, 67:23, 77:16, 77:18, 81:2, 82:9, 88:14, 95:5, 98:15, 98:16 familiar 30:7, 78:6 familiarize 69:15 far 40:1, 61:14 fashion 76:20 fast 116:22 faster 115:22, 116:15, 116:16, 117:18,	117:20, 119:3 feature 78:5 features 21:21 february 1:14, 4:1, 4:8, 135:16 feeds 105:15 feel 15:11, 87:7, 123:11 ferris 2:11 fetch 74:8, 74:14, 75:4, 75:9, 75:18, 76:24, 77:6, 77:23, 77:24, 90:4, 103:15, 104:2, 104:7, 128:5, 128:18 fetched 77:12 fetches 101:23, 102:9, 102:23, 103:3, 103:25, 106:10, 107:8, 107:12, 107:22 fetching 74:16, 75:11, 75:17, 75:24, 103:23, 104:21, 128:1 few 118:15 field 92:19 fifo 25:25, 26:22, 28:21, 29:9, 34:2, 34:3, 34:6, 34:10, 35:20, 43:23, 43:25, 45:4,	46:5, 46:14, 81:3, 92:25, 93:4, 93:19, 95:20, 95:23, 102:19, 102:20, 104:4, 104:6, 104:10, 104:12, 104:13, 104:18, 104:22, 105:1, 105:15, 106:1, 106:5, 106:9, 106:14 fifo's 104:22 fifos 33:21, 34:2, 34:4, 41:25, 42:3, 42:7, 42:8, 43:6, 43:8, 43:18, 43:19, 44:5, 44:6, 44:22, 44:24, 45:7, 45:12, 45:16, 45:17, 45:24, 46:9, 46:20, 47:1, 47:2, 47:3, 47:12, 47:13, 47:14, 47:16, 47:18, 47:19, 47:20, 47:22, 48:3, 81:4, 96:22, 96:25, 97:1, 97:6, 97:14, 97:20, 98:9, 98:18, 98:19, 98:24 figure 6:24, 6:25, 7:22, 8:18, 8:23, 9:5, 30:11, 53:20, 65:22, 66:2, 66:6, 66:7, 97:4, 97:6, 97:14, 97:15, 97:16, 97:17,
--	--	--	---

97:20, 98:25, 102:21, 104:13, 104:14, 104:16, 104:17, 118:11 file 114:14 fill 40:10, 115:22, 116:3, 116:15, 116:16, 117:19, 117:20, 117:22, 118:21, 119:3 filled 33:25 filling 43:18, 43:19, 43:23 final 82:17 financial 135:12 find 13:5, 21:18, 32:12, 35:12, 66:3, 66:13, 66:14, 85:13, 87:16 finding 123:7 fine 6:17, 7:17, 12:13, 22:23, 33:24, 36:17, 45:13, 79:4, 87:9 fine-grained 84:17 fine-granted 84:17 finish 22:14, 22:19, 22:24, 61:4 finished 60:18, 62:2, 64:8, 65:11, 65:21, 70:20, 72:16, 76:20, 77:15, 102:22,	103:11, 103:14, 103:19, 103:21, 106:9, 123:24, 123:25 finishes 77:8, 77:25, 103:1, 130:16 finishing 127:12, 131:15 first 5:4, 11:10, 12:4, 12:7, 25:18, 33:18, 35:14, 36:7, 37:3, 37:22, 38:10, 39:13, 41:14, 44:20, 50:9, 53:12, 56:6, 57:9, 60:2, 65:20, 70:14, 70:15, 72:14, 72:15, 72:20, 73:9, 73:20, 73:23, 73:25, 74:1, 75:9, 75:13, 76:18, 76:24, 77:18, 77:24, 78:11, 90:3, 90:6, 93:3, 96:5, 123:16, 129:6, 130:6, 132:10 first-in 92:25 first-out 92:25 fiske 84:9, 84:16, 85:13, 85:20, 86:13, 86:20, 86:23, 86:25, 87:2, 87:10, 87:11, 87:14, 87:22, 87:23, 88:6, 88:12, 91:20, 91:21, 100:9, 100:18,	100:20, 101:1, 127:22, 133:1, 133:7, 133:8, 133:12, 133:16, 133:18, 133:20, 133:23, 134:3, 134:8, 134:15 fiske's 100:12 five 88:1 five-minute 83:19 fixed 114:7 flight 128:9 floating-point 63:18, 64:1, 64:12, 65:3, 65:7, 65:10, 67:6, 67:8, 67:18, 69:8, 69:11, 70:2, 70:7, 70:16, 70:25 focus 54:6, 68:9, 97:12 focused 121:19 followed 74:11 following 33:20 follows 5:6 foregoing 135:5, 135:6 forget 61:3, 125:8, 131:13 forgetting 131:16 form 6:8, 7:6, 9:2, 11:9, 15:4, 15:16, 17:5,	17:13, 18:12, 18:18, 18:25, 19:23, 20:8, 20:21, 22:9, 23:10, 23:17, 24:5, 29:3, 31:6, 31:14, 31:25, 32:8, 34:18, 37:24, 38:14, 41:10, 42:10, 42:24, 43:1, 43:21, 44:8, 46:10, 46:12, 56:9, 60:10, 68:4, 68:13, 68:23, 70:4, 72:2, 76:2, 77:2, 78:1, 78:18, 94:5, 95:25, 96:4, 96:24, 100:16, 102:13, 111:21, 112:9, 112:23, 115:6, 117:4, 117:15, 118:7, 118:18, 120:2, 120:25, 125:6, 128:7, 129:5, 129:21, 130:8, 130:18, 133:2, 133:5, 134:10 format 31:17 found 7:14, 87:22, 115:17, 121:19, 123:10 four 124:5 free 15:12, 87:7, 123:11 freely 122:5 front 53:18, 53:20, 132:5
--	---	--	---

fulfills 20:17 full 33:13, 34:2, 34:4, 35:16, 36:7, 38:2, 38:20, 38:21, 40:4, 40:13, 42:13, 42:20, 43:11, 43:16, 60:7, 60:12, 105:23, 111:1, 111:6, 112:2, 121:5, 121:12 fully 60:25 function 9:20, 34:21, 93:2, 99:16 functionality 94:7 functioning 128:23 furnished 38:12 further 16:18, 36:19, 40:18, 41:1, 41:8, 41:17, 42:14, 42:22, 80:1, 81:1 G gave 65:23 general 16:13, 16:17, 16:20, 16:24, 17:7, 17:8, 17:17, 18:24, 20:11, 24:9, 25:10, 25:14, 25:20, 26:7, 26:10, 29:1, 29:8, 29:23, 78:8, 79:8, 79:19, 79:22, 79:24, 91:25,	92:1, 92:5, 92:11, 93:8, 94:10, 94:15, 94:18, 94:20, 96:19, 96:24, 98:1, 98:12, 98:20, 99:1, 99:6, 108:10, 110:14, 110:23, 111:24, 112:1, 112:6, 112:15, 113:9, 113:11, 113:14, 119:7, 120:6, 127:5 generally 29:1, 36:9, 57:24, 124:11 generate 79:7 generated 35:7, 110:9 geometric 27:8, 27:22, 28:2, 29:14, 29:17, 29:25, 30:4, 30:13, 30:14, 52:4, 53:21, 56:14 geometry 10:1, 30:6, 30:13, 30:14, 30:17, 30:21, 30:22, 31:1, 31:5, 31:12, 31:20, 31:23, 34:20, 51:16, 51:17, 51:22, 51:23, 51:25, 52:3, 52:15, 52:17, 52:18, 52:20, 52:22, 53:25, 54:3, 54:13, 54:19, 56:7, 56:12, 56:21, 57:3, 57:20, 58:18, 102:22, 102:23, 102:25, 103:2,	103:3, 103:10, 104:2, 106:8, 106:10, 107:12, 107:14 getting 19:18, 53:9, 78:19, 82:18, 91:5, 96:1, 99:19, 130:3 give 23:22, 89:9, 89:14, 92:21, 117:1, 120:21 given 81:4, 113:13, 135:7 gives 33:19, 58:11 giving 66:19, 96:15 glovsky 2:11 gm 30:14, 30:17, 33:25, 34:4, 34:9, 34:15, 34:16, 34:22, 34:24, 35:1, 35:5, 42:9, 42:23, 43:16, 44:17, 51:23, 55:12 gms 30:7, 33:13, 35:8, 35:15, 35:18, 35:23, 36:2, 36:8, 36:13, 36:18, 36:20, 37:3, 37:7, 37:10, 37:17, 37:23, 38:9, 38:11, 38:16, 38:17, 38:18, 38:20, 38:21, 38:24, 39:18, 39:19, 39:25, 40:4, 40:10, 40:13,	40:17, 40:19, 40:20, 41:1, 41:7, 41:8, 41:16, 41:17, 41:22, 41:24, 42:6, 42:13, 42:15, 42:20, 43:6, 43:8, 43:11, 43:16, 43:18, 44:4, 44:24, 45:6, 45:10, 45:15, 45:18, 45:23, 46:4, 46:8, 46:15, 46:19, 46:25, 47:2, 47:11, 48:2, 48:6, 48:24, 51:8, 51:10, 51:18, 51:20, 52:8, 53:2, 53:4, 54:10, 54:24, 55:6, 55:18 go 5:15, 12:14, 12:18, 12:21, 17:15, 25:2, 35:13, 36:6, 42:6, 44:3, 50:5, 51:12, 63:7, 65:5, 73:20, 73:23, 79:5, 83:16, 84:5, 105:11, 108:14, 109:13, 117:1, 118:8, 118:10, 122:24, 125:16, 126:13, 126:14, 128:18, 131:20 goes 27:9, 30:10, 43:18, 46:17, 50:22, 56:19, 70:18, 103:20 going 5:11, 7:10,
---	--	--	--

10:5, 12:13, 20:13, 25:5, 28:11, 33:7, 33:9, 40:5, 40:17, 40:18, 45:5, 56:22, 59:3, 68:6, 73:15, 75:9, 77:3, 78:21, 83:20, 84:6, 85:19, 87:4, 87:14, 87:22, 90:10, 97:11, 98:4, 105:4, 108:1, 108:15, 130:1, 130:19, 131:21, 132:13, 133:14, 134:23 good 5:9, 5:10, 24:17, 34:8, 59:18, 69:15, 90:22, 132:3, 132:4 grab 59:8 graphic 29:23 graphical 31:3 graphics 8:19, 29:24, 30:1, 30:25, 31:2, 32:14, 53:13, 53:15, 101:23, 102:18, 105:14, 105:15, 105:24, 106:5, 106:24, 106:25, 107:7 ground 5:17, 84:6, 84:22, 108:15 guess 7:15, 14:8, 43:3, 57:15, 82:16, 92:15, 104:15, 123:15,	129:9, 131:12 H hand 5:11, 7:8, 7:10, 21:11, 63:5, 63:24, 85:19, 97:18, 128:9, 128:12, 135:15 handing 109:6 handle 33:9, 39:12, 105:10 handler 88:2 handles 105:6 hanspeter 1:12, 3:3, 4:4, 5:3, 134:23 happen 52:19, 54:3, 62:23, 64:11, 64:12, 76:12, 77:1, 81:13, 81:18, 82:20, 89:22, 111:20 happening 10:2, 35:4, 64:2, 69:9, 71:17, 72:8, 72:9, 72:25, 87:25, 124:19, 124:20 happens 34:12, 38:6, 38:11, 43:5, 43:24, 46:4, 46:18, 46:22, 46:25, 47:8, 47:11, 52:17, 52:21, 54:18, 67:1, 75:1, 75:8, 76:24, 77:23, 88:21 happy 15:9, 18:6,	56:5, 87:4, 87:5, 110:18, 121:20 hard 116:11 hard-disk 95:21, 95:22 head 78:4 heading 25:10, 50:6, 62:6, 84:12 hear 29:4, 47:18 help 54:6, 85:18, 124:6, 126:15 helpful 7:13 helps 121:11, 123:1 hence 134:4 here 4:3, 9:25, 10:10, 11:21, 12:3, 20:22, 25:23, 26:3, 27:5, 28:24, 30:22, 31:4, 35:2, 38:6, 40:3, 44:16, 49:16, 52:11, 55:3, 55:5, 57:16, 59:24, 62:14, 63:13, 66:20, 69:1, 69:18, 76:17, 83:12, 85:8, 85:11, 86:13, 87:18, 87:22, 88:6, 96:16, 98:6, 98:8, 101:9, 102:10, 102:17, 102:20, 103:9, 109:1, 109:20, 111:4, 111:15, 114:20,	115:17, 115:24, 117:7, 117:19, 118:8, 118:20, 119:10, 121:10, 121:16, 123:23, 124:4, 124:12, 128:4, 128:20, 130:1, 131:3, 134:1 here's 39:24 hereby 135:5 hereunto 135:14 herrington 4:12 hey 120:19 high 54:22, 82:13, 120:21, 124:14, 134:16 high-level 79:2, 124:25, 134:7 high-order 105:25 high-performance 77:11 high-priority 80:23, 82:2, 82:7, 82:19 higher 117:2 hold 14:21, 16:7, 90:11, 113:21 honest 56:1, 87:2 honestly 55:9 host 8:20, 30:23, 35:18, 35:23, 36:20, 37:11, 38:25, 39:5, 39:19, 39:24,
--	--	---	---

40:9, 41:2, 41:9, 41:17, 41:25, 42:14, 44:4, 44:24, 46:6, 46:7, 46:11, 46:21, 46:23, 48:2, 56:13, 105:7, 106:22, 106:23 houston 2:6 however 22:17, 22:25, 26:16 hunting 83:16 hypothetical 96:15 hypotheticals 96:1	imply 44:10, 103:8, 103:9, 106:25, 129:9, 129:10 implying 98:6 impression 55:11 include 17:9, 17:20, 26:18, 31:1, 31:17, 31:21, 31:23, 52:15, 54:13, 54:14, 56:14, 56:21, 85:6, 85:10, 91:17, 96:24, 98:9, 106:12, 114:9 included 87:17, 99:1 includes 13:23, 30:1, 30:24, 31:3, 54:1, 86:8, 91:2, 94:16, 98:1 including 53:16, 56:15, 80:2, 93:13, 98:11, 113:11, 113:14 incoming 108:12, 110:16, 111:3, 111:6, 112:7, 112:20, 112:22, 116:8, 119:12, 120:7 incomplete 88:19 independent 127:3 indicate 102:4 indicated 98:14 indicates 106:16	individual 88:19 individually 29:12 inferring 16:1 information 80:3, 99:21, 119:20 input 25:25, 28:21, 29:9, 33:12, 33:21, 34:3, 34:6, 37:5, 37:7, 37:9, 37:13, 38:19, 38:20, 39:19, 39:25, 40:4, 40:10, 40:13, 41:25, 42:3, 42:7, 42:12, 42:20, 43:6, 43:8, 43:10, 43:15, 43:16, 43:18, 43:19, 43:23, 43:25, 44:5, 44:6, 44:24, 45:4, 45:7, 45:11, 45:16, 45:17, 45:24, 46:5, 46:8, 46:13, 46:20, 47:1, 47:3, 47:12, 47:14, 47:18, 47:22, 48:3, 109:11, 109:12, 114:14, 115:21, 115:22, 116:3, 116:14, 116:15, 116:24, 117:13, 117:20, 118:4, 118:15, 118:16, 119:2, 119:3, 119:14, 119:17, 119:21, 120:13, 120:20, 121:3, 121:4	inside 97:1 instance 63:13, 63:15, 66:20, 66:21, 89:7, 103:2, 121:19 instances 106:4 instead 14:14, 43:7, 55:3, 129:4, 129:19, 130:16 instruction-type 99:20 intel 6:20, 8:3, 8:6, 8:16 intensive 27:9, 27:23 interconnection 95:3 interest 135:12 intermediate 89:20, 110:9, 110:11 interpret 59:20, 59:21, 60:3, 60:13, 82:23, 94:23, 94:24 interpretation 17:23, 24:17, 49:5, 74:23, 74:25, 92:4, 94:13 interpreted 10:14 interpreting 74:20, 93:16, 103:9 interrupt 22:18 introduce 78:15, 97:11 introduces 35:15
I			
i" 8:3 i0 69:24 id 85:11 idea 87:3 identified 54:14 identify 11:14, 18:9, 25:24, 85:1 ids 132:21 impact 125:14 implementation 26:17 implemented 6:19, 53:21 implements 91:11 implies 66:22, 99:12, 102:7, 103:18			

introducing 43:6 introduction 39:19, 40:19, 41:1, 41:17, 42:14 introductions 36:19, 41:8, 42:22 intuitively 63:1 invalid 87:10 ipr 1:10, 4:7 issue 127:20 item 25:19, 68:9, 68:10 items 25:19, 25:20, 109:15 itself 52:20, 95:12, 112:19	32:8, 34:18, 37:24, 38:14, 41:10, 42:10, 42:24, 43:1, 43:21, 44:8, 46:10, 46:12, 56:9, 60:10, 70:4, 72:2, 76:2, 77:2, 78:1, 78:18, 94:5, 95:25, 100:16, 102:13, 111:21, 112:9, 112:23, 115:6, 117:4, 117:15, 118:7, 118:18, 120:2, 120:25, 125:6, 128:7, 129:5, 129:21, 130:8, 130:18, 131:20, 132:2, 133:4, 134:18 johnson@bakerbot- ts 2:8 jump 76:9, 91:23	11:20, 20:22, 20:23, 21:3, 21:7, 21:8, 21:9, 21:16, 21:19, 23:12, 23:13, 25:11, 26:2, 26:3, 26:5, 28:22, 62:14, 62:19, 63:6, 63:10, 63:11, 63:15, 63:21, 64:24, 66:17, 66:23, 67:1, 67:4, 67:15, 68:25, 69:2, 76:11, 80:12, 125:9, 125:13, 127:21 L labeled 10:24, 97:6 language 11:2, 19:4, 24:16, 49:6, 61:6, 61:16, 64:16, 69:6, 69:18, 75:2, 76:6, 76:8, 79:2, 79:14, 79:16, 85:23, 92:7, 93:15, 95:7, 111:2, 116:12, 118:20, 119:19, 125:22 large 16:4, 16:7, 16:11, 97:17 largely 74:10 larger 8:24, 9:3, 12:17, 13:20, 15:20 last 48:21, 50:16, 58:6, 76:10, 102:23, 103:1,	103:11, 103:14, 103:19, 103:21, 106:9, 114:17 later 32:4 lead 34:24 leads 57:6 least 32:10, 32:11, 32:17, 34:22, 54:1, 77:16, 101:8, 122:23, 127:1, 128:23 leaves 33:1 leaving 73:25 left 7:18 let's 10:5, 10:17, 11:22, 12:3, 12:17, 18:20, 21:13, 21:20, 23:20, 25:2, 26:24, 29:13, 33:2, 33:3, 35:13, 36:5, 38:23, 39:7, 49:21, 57:4, 60:17, 61:5, 61:7, 61:9, 67:4, 70:13, 70:14, 70:18, 71:1, 71:2, 71:3, 71:6, 71:9, 72:13, 74:4, 75:10, 79:5, 79:25, 80:10, 83:18, 85:21, 86:23, 89:9, 89:15, 91:23, 93:4, 97:12, 105:11, 109:13, 118:9, 118:11, 121:25,
J jeff 23:1 jeffrey 2:4, 2:8, 4:19 job 1:21 johnson 2:4, 3:5, 4:19, 4:23, 7:6, 9:2, 11:9, 15:4, 15:16, 17:5, 17:13, 18:12, 18:18, 18:25, 19:23, 20:8, 20:21, 22:9, 22:13, 22:18, 22:23, 23:10, 23:17, 24:5, 29:3, 31:6, 31:14, 31:25,	K keep 22:19, 90:13 keeping 126:23 key 76:17 kind 14:19, 36:10, 54:21, 62:24, 62:25, 77:13, 90:11, 90:12, 94:14, 98:11 knowing 88:10, 131:11 knows 115:4 kohn 5:21, 6:1, 11:7, 11:12,		

126:13, 129:13, 131:20 level 54:22, 80:18, 91:21, 124:14, 134:16 levels 54:22, 134:2 levin 2:11, 4:18 likely 90:5, 104:10 limitation 49:25, 123:2, 123:7, 133:24 limitations 113:24 limited 14:20, 120:13 lindholm 108:16, 108:21, 108:22, 108:24, 109:3, 109:5, 109:7, 109:16, 109:19, 110:2, 110:7, 110:13, 110:17, 111:8, 112:3, 112:4, 113:2, 113:7, 113:17, 113:22, 113:23, 114:1, 114:8, 114:19, 114:20, 115:3, 115:11, 115:16, 115:17, 116:12, 117:6, 118:9, 118:11, 118:13, 119:21, 119:23, 120:12, 120:19, 121:1, 121:20, 121:22, 122:5, 123:18, 123:21, 124:13, 124:14, 124:21, 125:10, 125:13, 126:16, 126:19, 126:20, 127:18, 128:4, 128:15, 129:1,	129:16, 130:5, 130:11, 130:14, 130:19, 130:24, 131:9, 131:11 line 13:6, 62:11, 96:13, 108:8, 115:19, 117:8, 118:3, 130:20 lines 109:9, 115:7, 118:15 literally 63:1 little 8:14, 9:24, 19:5, 19:17, 19:18, 26:21, 35:11, 45:1, 54:6, 54:22, 56:2, 68:2, 68:7, 69:16, 73:1, 79:1, 79:3, 95:6, 109:18, 116:11 llp 4:12 loaded 62:12 local 26:18 locally 27:23 located 94:20 location 75:14, 75:16, 89:24, 89:25, 90:6 locations 75:19, 90:4 lockstep 62:25 logic 81:21, 81:24 long 77:14, 87:3, 90:15, 92:16,	94:20, 95:13, 113:13, 113:18 longer 41:24, 74:21, 84:6 look 11:20, 12:14, 13:18, 15:7, 15:9, 15:12, 15:13, 16:22, 20:23, 21:10, 21:12, 23:11, 28:6, 28:8, 32:2, 33:2, 35:11, 50:10, 64:16, 69:23, 79:25, 85:22, 86:20, 86:23, 87:8, 94:15, 97:4, 99:22, 100:1, 104:15, 104:17, 108:6, 109:5, 109:8, 109:18, 113:2, 113:25, 121:20, 122:10, 123:11, 131:9, 133:21 looked 17:25, 64:25, 107:11, 125:18 looking 7:22, 8:23, 9:5, 11:1, 12:7, 17:21, 23:15, 24:16, 32:2, 52:12, 53:18, 65:22, 76:5, 83:11, 87:14, 87:23, 97:14, 98:3, 101:9, 102:14, 102:15, 104:12, 108:15, 109:19, 109:22, 112:10, 115:10, 116:13 looks 79:21, 126:14 loose 79:1	loosely 93:15 lost 14:15, 25:22 lot 27:7, 27:21, 90:23, 117:1, 117:2, 120:20, 134:1 louisiana 2:5 low-priority 80:22, 82:1, 82:6, 82:13, 82:18 lower 53:19 lunch 107:25 M made 86:8 magnetic 96:11 maintain 12:23, 12:25, 19:11, 20:6, 20:10, 23:9, 101:18 maintained 16:20, 24:9 maintaining 16:14, 16:25, 17:3, 17:8, 17:18, 18:10, 18:11, 18:24, 19:13, 19:15, 20:5, 20:12, 22:8, 92:2 maintains 10:21, 11:4, 19:7, 19:22, 20:4, 25:15, 102:5 major 78:16 make 8:9, 10:9,
--	--	--	--

28:19, 32:3, 48:24, 55:13, 61:14, 76:23, 83:14, 89:11, 90:18, 91:14, 92:11 makes 81:11, 94:3, 122:25 making 81:18, 89:16, 121:8 management 20:2, 21:8 manipulation 59:12, 59:23, 60:1, 60:6, 60:14, 64:20, 85:24, 85:25, 88:17, 126:12, 127:16 manner 84:18, 126:24 many 78:21, 88:8, 121:23 map 48:6 mapped 55:25 mapping 9:9, 11:2, 34:23, 49:20 maps 49:15 march 135:17 marked 3:8 marks 134:22 massachusetts 1:13, 4:13 master 96:23, 98:25, 99:3, 100:6, 100:7, 100:12, 100:15, 100:22,	101:5, 101:14, 101:16, 101:22, 102:4, 102:18, 102:20, 104:5, 104:18, 104:20, 104:22, 105:1, 105:5, 105:11, 105:13, 105:24, 106:4, 106:17, 106:19, 106:21, 106:25, 107:7, 107:15 matches 126:11 matrix 58:7, 58:10, 58:14 matter 4:5, 5:13 maybe 22:5, 28:12, 32:15, 60:15, 60:18, 74:4, 78:5, 79:1, 88:22, 89:15, 90:25, 94:8, 97:9, 103:12, 124:5, 124:9, 124:13 mean 9:3, 9:15, 9:19, 10:3, 13:21, 18:14, 23:18, 26:23, 43:3, 43:24, 48:4, 49:24, 51:6, 51:21, 52:24, 53:12, 54:18, 55:2, 55:17, 56:4, 56:5, 56:11, 58:5, 59:15, 60:11, 60:22, 60:23, 61:3, 61:22, 64:6, 66:8, 66:12, 67:22, 69:2, 70:5, 75:2,	77:4, 78:2, 83:13, 84:17, 87:21, 91:6, 91:9, 94:22, 94:25, 95:15, 95:18, 96:2, 98:10, 99:9, 99:10, 99:23, 102:11, 103:6, 104:17, 106:20, 111:22, 112:4, 120:16, 121:6, 121:16, 124:13, 125:20, 126:5, 128:16, 129:6 meaning 13:25, 29:5, 29:18, 31:13, 32:19, 34:7, 34:13, 46:14, 62:23, 74:20, 88:9, 120:9 means 15:23, 24:10, 24:14, 38:18, 61:21, 62:1, 64:5, 93:10, 121:4 meant 9:10, 41:4, 57:1, 57:2, 57:15, 58:6, 58:7, 60:12 mechanism 103:13 media 4:3 medium 96:10, 96:11 meet 61:16, 75:2 meeting 52:8, 76:6 meets 49:25, 69:6, 69:18, 76:8, 110:13, 119:18 megahertz 81:6	memories 92:10, 94:19, 95:4, 96:22, 96:25, 97:1, 97:13, 98:18, 98:22, 98:24 memory 11:16, 11:23, 13:22, 20:2, 21:8, 26:7, 27:7, 27:8, 27:22, 60:19, 64:10, 75:14, 75:15, 89:1, 89:24, 89:25, 90:4, 90:6, 92:10, 92:16, 93:1, 93:2, 93:6, 94:8, 94:9, 94:14, 97:5, 97:21, 97:22, 98:9, 98:13, 98:14, 98:15, 99:6, 99:10, 99:12, 99:15, 99:20, 111:5, 111:16, 119:11, 119:13, 125:17 mention 8:17, 26:1, 26:9, 28:7, 28:10, 29:23, 32:4, 32:12, 62:11, 85:7, 87:22, 134:17 mentioned 16:4, 19:5, 21:15, 21:16, 28:24, 62:17, 66:7, 108:25 mentions 27:11, 28:13, 28:25, 30:4 meshes 106:1 message 30:10, 30:11,
--	--	---	---

30:14, 30:17, 30:18, 30:21, 31:5, 31:12, 31:15, 31:16, 32:6, 32:20, 32:21, 34:7, 49:16, 51:6, 51:10, 51:25, 52:22, 54:17, 55:16, 55:17, 55:21, 56:7, 56:12, 56:17, 56:19, 58:18, 58:22, 58:23, 58:24 messages 30:6, 30:22, 31:1, 31:19, 31:20, 31:23, 32:5, 32:9, 32:12, 32:18, 34:19, 34:20, 34:22, 49:16, 51:17, 51:22, 51:24, 52:14, 52:18, 52:19, 52:20, 53:7, 54:13, 55:4, 55:18 met 43:17, 101:22, 111:18, 123:7 method 122:4, 123:21 michael 2:20, 4:10 microprocessors 8:3 middle 8:2, 65:2, 115:12 might 26:19, 37:17, 56:21, 59:24, 59:25, 60:13, 65:7, 65:8, 65:13, 65:14, 83:13, 94:1,	98:9 mind 8:12, 10:8, 28:11, 85:16, 121:4, 126:23 mintz 2:11, 4:17 misquoted 14:11 missed 6:16, 28:12 missing 117:19 misspoke 39:10 mistaken 11:21 mistyped 107:23 misunderstood 40:25 mixing 19:2 mode 63:17, 65:2, 65:11, 66:4, 66:10, 66:21, 67:16, 67:17, 68:13, 68:19, 69:14, 70:16, 73:5 model 52:4 models 27:8, 27:22, 28:2, 29:14, 29:17, 30:1, 30:4, 31:3 modes 63:18 modification 117:24, 118:3 modifications 121:23 modified 119:17 modify 116:23, 117:12,	118:1 modifying 115:13, 115:19, 117:9, 118:13, 118:17, 119:1, 121:1 module 96:23, 98:25, 99:4, 100:6, 100:7, 100:12, 100:13, 100:15, 100:22, 101:5, 101:14, 101:16, 101:22, 102:5, 102:18, 102:20, 102:22, 102:25, 103:2, 103:10, 104:2, 104:5, 104:18, 104:20, 104:22, 105:5, 105:12, 105:13, 105:24, 106:4, 106:8, 106:17, 106:19, 106:21, 106:25, 107:7, 107:14, 107:15 monitor 4:9, 116:17, 119:4, 119:9, 121:6, 121:7, 129:9 monitored 129:15 monitoring 116:21, 117:17, 118:25, 119:10, 120:17 monitors 118:22, 129:11 more 7:2, 15:21, 15:23, 15:24, 19:17, 21:18, 23:12, 27:7, 32:21, 34:9, 34:14, 49:17, 49:21, 52:16, 54:22, 56:2,	57:24, 63:3, 63:4, 69:16, 69:19, 78:22, 79:1, 79:3, 85:22, 86:9, 89:15, 106:13, 111:6, 111:17, 112:1, 116:7, 116:10, 120:14, 120:22, 121:5, 121:9, 121:12, 121:21, 127:1, 127:12, 129:24, 131:9, 134:6, 134:15 morning 5:9, 5:10 mountain 2:14 move 26:14, 61:5, 62:24, 79:4, 83:17, 121:11 much 116:5, 120:13 multi 126:25 multiple 7:22, 128:8, 128:9, 128:22 multiplication 65:24 multiplier 65:25 multiply 58:13 multiplying 58:10 multiprocessor 128:8 multithread 129:23, 131:11 multithreading 124:15, 127:6, 127:11 must 99:7, 101:18 myself 9:24, 28:17,
--	--	--	---

69:16	notarial	observation	43:7, 43:19,
N	135:15	31:19	45:16, 82:2,
necessarily	notary	obvious	82:7, 82:17,
24:23, 25:1	135:4	121:12	101:14, 107:21,
need	nothing	obviously	123:9, 125:21,
11:20, 14:7,	5:5, 134:20	118:22, 120:10	126:6
14:13, 21:25,	notion	occurs	open
60:11, 74:4,	63:10	34:12	10:18, 33:1
86:21, 90:3,	now"	office	operate
134:8	45:8	1:1, 4:7	75:6, 116:7
needed	number	officer	operated
27:8, 27:22,	114:12	135:4	35:24, 38:25,
95:2	O	often	45:18, 69:22,
needs	objection	91:10	70:24
12:17, 12:23,	7:6, 9:2, 11:9,	oh	operating
12:25, 13:20,	15:4, 15:16,	9:18, 12:9,	52:7, 69:13,
15:20, 15:23,	17:5, 17:13,	13:8, 19:15,	70:25, 73:24,
15:24, 22:21,	18:12, 18:18,	20:13, 109:5,	75:4, 75:5
34:24, 64:18,	18:25, 19:23,	117:20	operation
105:10	20:8, 20:21,	on-chip	37:1, 37:20,
neither	22:9, 23:10,	11:16, 11:23,	38:3, 39:17,
46:11, 96:3,	23:17, 24:5,	13:13, 13:14,	44:19, 49:19,
135:10	29:3, 31:6,	13:25, 14:3,	59:23, 60:1,
never	31:14, 31:25,	20:1, 20:3,	60:2, 60:6,
96:7	32:8, 34:18,	20:6, 20:15,	60:7, 60:14,
new	37:24, 38:14,	20:17, 20:20,	60:15, 60:22,
27:6, 89:11,	41:10, 42:10,	20:25, 21:13,	61:8, 61:9,
90:25, 97:10,	42:24, 43:1,	21:15, 21:21,	61:12, 61:13,
97:11	43:21, 44:8,	22:4, 22:6,	61:17, 61:18,
next	46:10, 46:12,	23:3, 23:8,	62:1, 63:22,
6:2, 8:1, 33:7,	56:9, 60:10,	23:14, 26:2,	63:24, 64:6,
34:8, 44:19,	70:4, 72:2,	26:6, 28:21	64:12, 65:13,
71:4, 88:3,	76:2, 77:2,	once	67:6, 67:7,
95:13, 97:5,	78:1, 78:18,	46:19, 48:3,	67:14, 69:1,
101:24, 102:24,	94:5, 95:25,	103:12, 103:13	69:8, 70:7,
103:3, 103:15,	100:16, 102:13,	one-to-one	70:8, 70:16,
103:16, 103:20,	111:21, 112:9,	9:9, 34:23,	71:7, 71:10,
103:22, 104:3,	112:23, 115:6,	49:20	71:15, 71:16,
105:12, 105:18,	117:4, 117:15,	ones	72:18, 72:20,
106:7, 106:10,	118:7, 118:18,	28:24, 82:14,	89:23
107:8, 107:12,	120:2, 120:25,	125:2	operative
107:22, 131:15	125:6, 128:7,	ongoing	80:2, 101:18,
non-overlapping	129:5, 129:21,	71:8, 74:1	102:5
130:4	130:8, 130:18,	only	opinion
none	133:2, 133:4,	8:19, 8:22,	18:16, 19:25,
3:8, 38:11,	134:10	15:10, 20:10,	20:3, 20:24,
72:21	objects	29:2, 38:3,	21:23, 22:2,
	110:1	42:7, 42:8,	23:4, 87:9,

97:11, 97:25, 100:5 opinions 48:11, 97:10, 112:18, 133:16, 134:7 order 11:19, 70:7, 95:4, 127:2, 127:3, 130:5 ordinary 93:18 organization 83:6 originally 14:2 orrick 4:12 other 8:21, 9:23, 10:2, 22:5, 22:10, 22:12, 23:2, 24:12, 24:15, 26:14, 28:23, 31:18, 32:10, 32:12, 32:14, 32:22, 35:21, 36:1, 43:12, 44:2, 44:9, 47:25, 53:3, 58:25, 62:23, 64:3, 65:4, 67:5, 67:6, 68:25, 72:5, 73:6, 75:18, 76:7, 77:15, 77:18, 81:4, 82:20, 83:9, 90:16, 91:4, 91:17, 95:13, 96:12, 98:21, 98:22, 100:21, 101:5, 101:17, 106:15, 128:12, 133:10, 134:18 others 85:1, 121:22	otherwise 113:19, 121:6, 121:7, 135:13 out 14:6, 15:19, 15:21, 16:3, 17:14, 28:12, 33:21, 35:12, 35:20, 39:13, 51:1, 63:3, 63:4, 66:20, 90:24, 93:3, 118:11, 121:18, 129:13 outcome 135:13 output 34:2, 34:4, 34:10, 35:20, 44:22, 47:2, 47:12, 47:16, 47:19, 47:20, 115:23, 116:3, 116:15, 116:16, 116:18, 117:22, 118:5, 118:21, 118:23, 119:4 outset 37:19 outside 93:6, 93:11, 95:9, 95:21, 96:4, 96:6 over 22:14, 76:25, 87:10 overlapping 130:1, 131:14 overview 83:14, 101:12 owner 1:8, 2:10 P packet 34:8, 81:3 packets 26:14, 33:25,	34:1, 34:4, 34:5, 80:23, 80:25, 81:9, 82:1, 82:2, 82:7, 82:8, 82:11, 82:13, 82:18, 82:19, 83:5 page 3:3, 5:16, 6:3, 6:24, 7:14, 7:15, 7:16, 7:18, 8:1, 8:2, 10:13, 12:10, 12:20, 13:3, 13:18, 15:11, 16:23, 25:4, 25:6, 26:10, 30:16, 32:2, 33:2, 33:8, 33:11, 33:18, 35:14, 36:6, 37:2, 39:1, 39:9, 39:14, 39:23, 40:23, 43:14, 44:20, 48:20, 50:5, 50:16, 51:12, 53:19, 54:7, 62:4, 80:11, 80:13, 80:21, 81:1, 84:7, 84:11, 87:23, 96:17, 97:5, 102:15, 105:5, 105:18, 105:20, 105:21, 106:15, 107:9, 107:21, 108:15, 108:18, 108:19, 108:20, 110:6, 111:19, 114:4, 114:10, 114:22, 114:24, 115:11, 122:2, 123:17, 128:21, 129:23, 132:10, 133:13 pages 1:22, 63:7,	66:4, 66:9 paragraph 5:15, 5:23, 6:3, 8:12, 10:13, 11:6, 11:11, 11:15, 12:4, 12:8, 13:11, 15:6, 15:18, 20:25, 21:2, 25:3, 25:6, 25:24, 27:5, 33:18, 35:16, 36:7, 36:10, 36:15, 37:2, 38:5, 39:7, 39:8, 39:9, 41:12, 44:20, 44:21, 48:17, 50:17, 50:20, 50:21, 50:24, 51:13, 52:11, 52:12, 53:19, 54:23, 62:3, 65:1, 69:3, 80:10, 80:21, 83:12, 84:5, 84:15, 96:18, 98:5, 100:2, 100:9, 101:10, 102:2, 102:16, 105:5, 105:14, 105:19, 105:23, 107:6, 108:14, 109:9, 109:20, 109:24, 110:6, 110:20, 112:10, 113:10, 114:5, 114:16, 114:24, 115:2, 115:11, 115:12, 118:12, 122:2, 122:3, 123:9, 123:18, 126:15, 126:19, 128:21, 130:24, 132:11, 132:13, 132:20, 132:25, 133:6, 133:7, 133:14
--	--	---	---

paragraphs 124:5, 126:13, 133:15, 133:22 parallel 63:16, 67:10, 67:11, 67:12, 67:16 parallelism 68:5, 68:13, 68:16, 68:18, 68:23, 73:3, 78:16 paraphrasing 33:23 parentheses 10:15, 51:23 part 10:3, 13:25, 16:21, 18:3, 26:10, 27:10, 27:12, 28:14, 32:11, 32:14, 40:15, 40:22, 41:18, 51:24, 52:5, 64:2, 64:3, 64:10, 65:22, 67:1, 82:8, 91:15, 93:7, 93:20, 94:4, 95:4, 95:23, 96:14, 97:25, 109:14, 117:12, 117:21, 117:22, 119:13, 127:10, 130:21 partial 87:21 particular 5:19, 6:17, 7:12, 31:7, 31:17, 78:5, 79:14, 93:2, 107:20, 121:15, 121:19 parties 135:12 parts 9:6	pass 131:18 passage 31:7, 32:1 patent 1:1, 1:2, 1:8, 2:10, 4:6, 5:12, 5:18, 5:24, 10:18, 11:18, 11:25, 17:10, 20:7, 59:9, 59:19, 61:20, 84:1, 92:14, 92:15, 93:15, 96:3, 108:7, 126:16 patents 122:5, 123:21 path 67:12 paths 67:10, 67:11 pcus 110:23 people 68:2, 91:14, 93:14, 93:24, 93:25, 94:11, 94:22 percent 91:10 perfectly 87:2 perform 51:16, 59:16, 69:4, 71:20, 81:25, 88:18, 109:25, 110:13, 120:5, 129:19 performed 52:13, 66:23, 81:20, 125:23, 127:23 performing 86:18, 108:8, 111:9, 112:13, 121:25, 130:6 performing-operations-at-various 84:23	performs 53:20, 59:11, 64:19, 85:24, 112:5, 123:4 period 29:19, 42:2, 46:8 permits 47:11, 115:21, 116:14, 119:3 permitted 45:15, 45:17, 46:19, 47:1 person 76:15 personally 93:21, 94:24 petitioner 1:5, 2:2 petitioners 4:24 pfister 1:12, 3:3, 4:4, 5:3, 5:9, 23:2, 59:8, 83:25, 108:6, 132:3, 134:23 ph 1:12, 3:3, 5:3 phase 37:15, 44:19, 45:15, 48:1 phases 37:1 phrase 76:17, 122:24, 123:2 physical 95:3 physically 81:11 picking 53:17, 57:5, 60:20, 66:20 piece 13:8, 19:17, 67:21, 111:22 pieces 132:15	pipeline 53:22, 62:12, 62:17, 62:18, 63:1, 64:10, 67:20, 67:25, 68:6, 68:22, 70:19, 77:13, 88:1, 88:12, 109:25, 128:24 pipelined 62:22, 63:18, 67:17, 68:19 pipelining 63:20, 64:3, 64:5, 65:11, 67:19, 68:4, 68:20, 68:24, 73:6, 73:8, 73:12, 76:16, 76:19, 76:20, 78:10, 78:14, 78:17 pixel 24:11, 49:2, 50:10, 50:14, 51:2, 51:4, 51:6, 52:13, 54:24, 55:6, 55:12, 55:20, 56:18, 57:13, 57:15, 57:18, 58:3, 58:12, 58:15, 59:12, 60:1, 60:6, 64:20, 69:4, 71:16, 72:18, 79:7, 85:25, 88:17, 109:11, 109:12, 109:17, 112:22, 113:8, 113:16, 113:19, 114:6, 114:11, 114:14, 115:4, 115:14, 117:20, 118:16, 118:21, 118:23, 122:6, 123:19, 123:22, 123:25, 124:1,
---	---	---	---

124:14, 124:16, 124:19, 124:22, 124:23, 126:7, 126:8, 126:24, 127:16 pixels 56:20, 56:22, 56:25, 57:6, 57:11, 57:12, 57:14, 57:20, 58:13, 125:21 place 4:12, 7:12, 39:13, 99:12, 123:9 places 133:10 planet 4:11, 5:1 please 4:15, 5:16, 10:18, 12:21, 19:3, 46:1, 80:16, 86:24, 122:13 plural 6:10, 6:11 plus 98:18, 98:21 point 6:17, 7:12, 23:14, 28:12, 39:4, 41:20, 42:1, 45:22, 48:11, 50:2, 52:7, 52:10, 63:9, 65:15, 66:6, 70:8, 70:10, 70:13, 76:10, 78:9, 79:23, 80:13, 80:15, 81:14, 84:15, 86:21, 86:24, 89:11, 89:20, 90:9, 95:18, 95:19, 96:9, 100:14, 107:1, 107:3,	109:16, 121:18, 123:8, 126:16, 129:2, 129:17, 130:15, 131:12 pointed 17:14, 66:2, 66:9, 70:10, 105:3, 123:15 pointers 88:5, 88:9 pointing 15:21, 28:1, 65:25, 70:14, 88:9, 96:5, 97:2, 100:17, 100:22, 100:25, 101:4, 101:14, 108:24, 114:25, 133:17 points 14:6, 15:19, 16:3, 33:21, 67:5, 134:17 policy 81:2 polygons 30:23, 30:24, 30:25, 32:10, 32:11, 32:17, 35:19, 106:1 popeo 2:12 portions 133:23 posed 131:5 posita 13:21, 13:24, 14:24, 80:18, 81:12, 81:19, 86:7, 93:22, 93:23, 93:24, 96:16, 98:10, 99:9, 99:13, 99:14, 103:9, 106:20, 107:4, 110:21, 111:5, 111:19, 111:23,	112:18, 131:10 position 58:8, 58:9 possibility 14:3 possible 14:22, 14:25, 32:13, 33:12, 40:12, 58:13, 58:15, 77:10, 77:12, 90:9, 93:17, 94:13 possibly 18:17, 18:19, 28:22, 28:23, 98:21 post 46:22 potentially 26:20, 26:25, 27:16, 28:20, 98:19 precise 75:13, 79:3, 91:11 precludes 24:23 present 2:19 presumably 8:21, 26:5, 50:25, 110:10, 113:2 presume 61:7, 66:10 pretty 101:13 previous 9:9, 16:22, 21:20, 37:17, 38:16, 51:12, 84:11, 85:4, 104:4 primitive 101:24, 102:24, 103:4, 103:15, 103:16, 106:11, 107:8, 107:12,	107:22 primitives 29:23, 30:1, 30:4, 30:25, 31:2, 32:14, 53:25, 54:2, 56:15, 104:6, 104:7, 105:25 prior 5:19, 79:14, 92:8, 96:4, 132:16, 134:14 priorities 114:1, 121:24 prioritization 134:3 prioritize 113:19 prioritizing 113:16 priority 82:14, 113:13, 114:5, 114:9, 115:13, 115:20, 116:24, 117:2, 117:10, 117:12, 117:25, 118:2, 118:3, 118:14, 118:17, 119:16, 120:21, 121:2, 121:14 probably 24:6, 31:2, 55:16, 55:18, 61:19, 62:1, 93:10, 96:8, 96:13, 96:14, 121:22 problem 33:9, 33:15, 36:5, 36:11, 39:12, 39:17, 40:16, 41:15, 54:7 proceed 116:22 proceeds 116:19
---	---	---	--

process 36:17, 57:17, 79:11, 81:8, 81:12, 82:14, 84:16, 126:22, 126:25, 127:6, 127:11, 129:4, 129:24, 131:2 processed 54:17, 63:2, 65:10, 68:10, 127:2 processes 73:15 processing 7:3, 8:20, 9:24, 10:1, 10:2, 48:19, 48:25, 49:3, 49:6, 49:13, 49:18, 52:17, 53:22, 53:23, 54:1, 54:2, 54:18, 54:19, 55:17, 55:20, 55:21, 55:24, 56:18, 57:13, 57:15, 57:18, 57:22, 57:23, 57:25, 58:3, 58:12, 58:20, 58:23, 68:14, 103:21, 110:9, 113:15, 114:2, 116:19, 116:22, 124:15, 126:25, 127:7, 127:8, 127:9, 128:19, 129:3, 129:24, 130:1, 130:4, 131:11 processor 6:20, 7:1, 7:5, 7:20, 7:24, 8:7, 8:11, 8:16, 8:19, 9:13, 10:14, 10:16, 13:13, 15:20,	15:23, 26:1, 27:7, 29:25, 34:10, 34:16, 35:5, 36:13, 51:16, 53:8, 57:6, 57:10, 59:11, 62:22, 64:18, 65:15, 69:3, 71:6, 71:9, 71:22, 71:24, 74:11, 75:1, 77:11, 78:4, 78:6, 79:6, 81:16, 81:17, 84:12, 85:24, 88:1, 88:6, 88:13, 91:11, 93:5, 93:6, 93:11, 93:19, 95:15, 95:17, 95:20, 95:23, 99:3, 101:19, 102:6, 102:8, 105:7, 106:24, 108:9, 123:3, 123:5 processor's 34:21 processor-depend- ent 90:22 processors 7:19, 7:23, 30:13, 33:22, 34:1, 34:12, 34:13, 35:21, 36:13, 44:23, 47:3, 47:14, 47:22, 53:23, 71:19, 78:14, 96:23, 98:1, 98:7, 105:16 prog 97:21, 98:14 program 12:17, 13:20, 13:21, 13:24, 14:4, 14:21,	15:20, 15:24, 16:2, 16:7, 16:10, 21:22, 27:12, 98:15, 99:19, 127:1 programmable 96:22, 96:25, 97:5, 97:13, 97:22, 98:13, 98:17, 98:24, 99:6, 114:7 programs 13:22 progress 34:14 provide 80:2, 121:14 provides 80:7 proximity 95:3 ptab 4:6 public 135:4 purpose 16:14, 16:17, 16:20, 16:24, 17:7, 17:17, 18:24, 20:11, 24:9, 25:10, 25:15, 25:20, 26:7, 26:10, 29:1, 29:8, 79:8, 79:20, 79:22, 79:24, 91:25, 92:1, 92:5, 92:11, 93:8, 94:10, 94:15, 94:18, 94:21, 96:19, 98:12, 98:20, 99:1, 99:6, 99:10, 108:10, 110:14, 110:24, 111:24, 112:2, 112:6, 112:15, 113:9, 113:12,	113:14, 119:7, 120:6 push 63:3, 64:5 put 34:1, 44:4, 46:5, 46:8, 46:19, 47:1, 47:12, 52:24, 63:2, 83:5, 85:8, 90:12, 104:5, 131:13 puts 80:25, 102:19, 105:25, 106:7 putting 43:7, 106:5 Q question 11:3, 14:15, 14:16, 15:7, 15:14, 18:2, 18:7, 18:13, 19:4, 22:22, 22:24, 23:12, 23:18, 28:11, 28:16, 29:7, 29:13, 29:15, 29:20, 34:6, 35:10, 36:16, 41:13, 56:3, 56:5, 59:18, 61:2, 64:22, 67:25, 69:15, 70:12, 77:4, 79:18, 80:13, 89:6, 90:22, 91:22, 96:2, 96:7, 100:1, 100:24, 108:23, 112:24, 113:1, 113:3, 122:11, 129:16, 130:13, 130:14, 131:5, 131:8, 134:6 questioned 132:21
--	--	---	---

questions 21:17, 22:16, 32:5, 78:21, 134:19 queue 33:12, 37:7, 37:9, 37:13, 38:19, 38:20, 39:25, 40:4, 40:13, 42:20, 43:16 queues 37:5, 38:20, 40:10, 42:12, 43:10, 43:15 quick 83:18 quickly 8:9, 8:12, 10:8, 41:12, 66:13, 77:10, 77:12 quite 38:15, 43:20, 43:23, 77:3, 120:3 quotation 106:15 quote 12:16, 12:18, 13:6, 13:9, 13:12, 13:19, 27:5, 41:15, 48:21, 64:25, 69:17, 82:8, 107:11, 115:11 quoted 13:2, 113:21, 130:21, 131:3, 133:19 quotes 21:7, 114:20 quoting 13:11, 20:22, 39:21, 39:22, 110:7, 117:6, 126:19, 129:22, 130:23	R rasterization 57:6, 57:11, 57:17 rasterized 57:4, 57:5 rather 53:24, 69:24, 99:20 rationale 84:22 rdokhanchy@mintz 2:17 rdr 1:23, 135:22 read 8:12, 11:25, 12:2, 15:19, 17:10, 29:19, 32:24, 40:24, 56:2, 66:14, 87:4, 106:3, 106:7, 114:23, 115:16, 116:9, 116:11, 116:12, 121:16, 130:2, 134:8 reading 24:23, 29:18, 29:22, 31:4, 39:21, 39:23, 63:15, 79:13, 88:11, 110:17, 114:3, 114:15, 114:16, 117:7, 118:25, 135:9 reads 121:16 realizing 28:13 really 18:2, 19:1, 23:12, 42:12, 49:15, 51:23, 53:14, 55:16, 56:10, 63:8, 66:13, 73:8,	76:17, 89:13, 94:6, 129:14 realtek 1:4, 4:5, 4:20, 4:23 reason 8:16, 8:22, 104:23, 121:13 reasonable 14:24, 16:9 recall 15:8, 83:25, 84:3, 99:8, 99:22, 99:25, 101:11, 132:9, 132:14, 133:21 recalling 85:7 receipt 57:12 receive 107:1, 109:22 received 127:4 receives 51:22, 57:10, 109:10 receiving 51:17, 52:14 recess 59:5, 83:22, 108:3, 131:23 recited 49:23 recites 99:18 recollecting 91:20 recollection 133:22 record 59:4, 59:7, 83:21, 83:24, 108:2, 108:5, 131:20, 131:22, 131:25, 134:24, 135:7 redirect 131:19	reduced 135:8 refer 74:14, 83:13, 96:4, 101:25, 105:2, 115:24, 134:13 reference 6:8, 6:19, 7:9, 7:11, 15:8, 17:22, 21:11, 21:14, 35:11, 63:6, 85:6, 85:16, 85:20, 87:3, 87:18, 87:19, 89:2, 100:22, 104:25, 112:18, 131:9, 133:1, 133:10, 133:16 referenced 11:21 references 21:4, 25:11, 85:10, 108:16, 109:3, 109:5, 109:7, 110:18, 113:7, 125:11 referencing 74:15, 110:2 referred 54:20, 82:5, 86:10, 93:11, 93:12, 99:4 referring 6:23, 10:25, 13:14, 26:2, 31:7, 31:9, 44:12, 46:1, 47:7, 49:14, 51:23, 56:24, 62:14, 63:10, 63:20, 65:1, 65:6, 67:13, 70:23, 72:12, 72:18, 73:8, 75:18, 75:20, 79:14, 80:20,
--	--	--	--

85:3, 85:12, 86:22, 96:21, 104:14, 106:11, 110:6, 112:11, 113:24, 114:8, 114:10, 114:19, 117:21, 123:12, 123:13, 132:21 refers 80:22, 82:10, 103:19, 122:22, 124:3 refresh 78:24, 125:17, 133:22 regarding 25:9, 62:8, 84:13, 100:3, 110:12 regardless 130:10 regards 5:19, 5:25 register 16:14, 16:17, 16:20, 16:24, 17:7, 17:17, 18:24, 19:16, 20:11, 22:10, 22:12, 23:25, 24:1, 24:10, 25:10, 25:15, 25:20, 26:8, 26:11, 28:15, 28:20, 29:2, 29:8, 29:9, 29:11, 79:8, 79:20, 79:23, 79:24, 90:1, 91:25, 92:1, 92:5, 92:11, 92:19, 92:22, 93:1, 93:9, 93:14, 94:2, 94:8, 94:19, 95:9, 95:12, 95:16, 95:24, 96:14, 96:20,	96:24, 98:1, 98:12, 98:20, 99:2, 99:7, 108:10, 110:14, 110:24, 110:25, 111:13, 111:24, 112:2, 112:6, 112:15, 113:9, 113:12, 113:15, 114:13, 119:8, 120:6 registered 135:3 registers 26:19, 28:23, 90:25, 91:4, 98:21 reiterate 19:24 relate 54:10, 113:4, 113:5, 116:9 related 116:20, 134:8, 135:11 relationship 34:15, 56:8, 56:16, 86:5 relevance 86:17 relevant 13:8 relying 21:14, 66:17 remaining 44:23 remember 10:19, 127:21, 127:24, 134:12 rendering 10:1, 30:17, 32:5, 32:6, 32:9, 32:11, 32:18, 34:7, 34:14, 34:20, 52:13, 52:18, 56:17, 96:22, 98:1, 98:7, 99:3	repeat 19:3, 100:24, 130:12 report 5:16, 6:11, 6:14, 6:16, 25:3, 25:5, 25:6, 48:6, 48:8, 50:4, 50:19, 54:23, 55:7, 55:12, 62:4, 63:6, 63:9, 63:15, 65:1, 66:18, 67:14, 83:10, 83:11, 84:5, 85:2, 85:22, 86:3, 87:7, 87:8, 87:19, 96:18, 97:9, 97:12, 98:4, 100:2, 101:9, 101:21, 107:6, 107:15, 108:14, 109:8, 109:19, 109:23, 109:24, 110:7, 115:10, 122:3, 123:7, 123:11, 126:14, 132:17, 133:10, 133:11 reported 1:23 reporter 4:21, 4:25, 135:1, 135:4 represent 4:16 representing 4:11, 5:1 requested 135:10 require 27:7, 48:15, 102:8 required 23:9, 66:24, 67:1, 69:5,	71:21, 79:16, 81:7, 81:17, 112:25, 124:7, 131:7 requirement 16:13, 17:24, 18:11, 18:23, 19:11, 22:7, 60:9, 62:9, 84:1, 84:14, 84:24, 92:12, 92:16, 108:7, 110:13, 129:8 requirements 25:14 requires 10:20, 11:3, 31:10, 59:10, 81:13, 111:17, 127:14, 127:15, 129:7, 130:9 requiring 102:9, 119:7 resources 113:14, 114:12 respect 132:24 response 24:19, 51:17, 52:14, 79:7, 80:3 responsible 105:7 rest 15:8, 32:3 result 41:24, 60:18, 64:9, 64:13, 89:17, 89:25 results 24:1, 63:3, 63:4, 74:10 resume 60:2 return 48:1 reverse 55:24
---	---	--	--

Transcript of Hanspeter Pfister, Ph.D.
Conducted on February 16, 2024

60

reza 2:13, 4:17 rm 30:18, 34:1, 34:4, 34:7, 34:15, 34:16, 34:22, 34:25, 35:2, 35:6, 36:19, 38:9, 38:11, 38:13, 38:22, 39:18, 40:17, 41:7, 41:16, 41:22, 42:9, 42:22, 42:23, 43:9, 43:17, 43:25, 44:17, 45:4, 45:5, 51:5, 51:6, 55:12, 56:23, 56:24, 57:2, 57:3, 57:8, 57:9, 57:12 rms 35:7, 35:8, 35:15, 35:20, 35:24, 36:1, 36:8, 36:13, 37:15, 37:17, 38:17, 38:25, 39:5, 40:21, 41:5, 42:3, 42:7, 43:7, 43:19, 43:24, 44:7, 44:23, 45:6, 45:11, 45:16, 45:19, 46:14, 47:13, 47:21, 49:2, 54:24, 55:6, 55:8, 55:21, 55:25 road 2:14 robin 88:4 room 34:9, 111:10,	113:8, 115:4, 115:5, 116:8, 119:22, 120:22 round 88:3 run 128:17 running 81:6, 128:17 S safee 2:20, 4:10 said 6:14, 8:17, 9:18, 10:6, 20:15, 20:17, 26:25, 27:2, 27:13, 31:21, 40:25, 41:1, 44:11, 47:6, 47:19, 62:2, 67:20, 72:7, 72:23, 72:24, 73:7, 75:10, 78:13, 78:25, 100:25, 106:16, 125:3, 134:14, 135:7 sake 70:15 same 8:11, 31:15, 32:5, 32:20, 35:14, 48:12, 66:23, 66:25, 68:6, 69:10, 69:22, 71:25, 72:9, 81:9, 90:17, 93:20, 94:4, 95:5, 95:23, 120:23, 120:24, 127:20, 128:4 sample 109:17, 110:8, 110:10, 114:6, 115:14, 115:21,	116:2, 116:6, 116:14, 117:11, 118:4, 119:2, 127:2, 127:7, 127:8, 127:9 samples 110:4, 115:14, 115:20, 117:10, 121:2, 126:23, 126:24, 127:1, 127:3, 127:12, 129:25, 131:2 san 2:15 satisfy 17:24, 18:10, 18:23, 19:10, 22:7, 60:8 saves 88:2 saw 7:23, 37:10, 39:1, 39:3, 40:22, 122:23 saying 8:23, 11:12, 12:16, 21:3, 29:4, 29:6, 29:7, 36:12, 37:3, 46:3, 50:5, 52:12, 52:21, 55:4, 66:10, 70:1, 70:24, 73:23, 73:24, 76:25, 77:22, 77:25, 81:19, 82:16, 86:13, 91:18, 97:22, 104:24, 109:20, 110:7, 110:20, 111:4, 113:22, 114:18, 117:12, 117:25, 118:1, 120:1, 120:4, 123:16, 123:18, 126:6, 128:20 says 8:2, 13:12,	14:2, 14:12, 16:16, 16:23, 17:7, 17:17, 21:24, 24:6, 24:14, 24:17, 25:9, 27:6, 27:21, 30:22, 33:14, 34:11, 35:1, 35:18, 36:21, 38:8, 38:12, 38:17, 38:24, 39:12, 40:12, 40:20, 42:19, 44:22, 45:3, 45:10, 46:13, 46:16, 49:7, 49:10, 61:19, 67:14, 79:22, 80:6, 80:9, 84:16, 86:3, 86:16, 87:24, 99:8, 103:23, 103:25, 104:5, 104:25, 105:13, 105:19, 106:18, 108:8, 111:13, 111:15, 116:6, 116:23, 117:9, 118:2, 118:20, 118:24, 119:1, 119:10, 121:1, 123:3, 126:20, 127:11, 128:15, 130:25 scalar 63:19, 64:1, 65:4, 65:9, 65:13, 65:14, 65:16, 67:7, 67:9, 67:18, 69:8, 69:12, 70:3, 70:8, 71:10 scenario 33:12, 33:20, 34:3, 35:3, 36:23, 38:1, 38:3, 38:19,
--	--	---	--

39:24, 40:3, 40:12, 41:18, 41:19, 42:1, 42:11, 42:20, 43:13, 43:14, 44:5, 44:15, 45:9, 48:4, 75:3 scheduled 126:22, 131:1 scheduler 126:21, 130:25 scheduling 81:2, 100:12 scheme 88:4 scoreboard 90:12 seal 135:15 search 66:12 second 11:2, 39:23, 40:23, 72:21, 73:10, 73:19, 73:21, 73:23, 75:11, 75:15, 76:21, 77:6, 77:19, 77:23, 78:11, 81:3, 85:22, 102:2, 102:15, 105:14, 105:19, 105:23, 114:5, 130:5 secondly 96:8 section 5:22, 13:9, 25:22, 25:23, 27:5, 33:4, 33:7, 33:19, 38:5, 80:11, 83:12, 87:23, 101:8, 102:15, 105:2, 105:12, 105:18, 105:23, 121:16, 122:20, 133:11, 133:17,	133:19, 134:1, 134:16 sections 134:9 see 6:5, 8:1, 10:21, 11:11, 12:5, 13:3, 16:12, 16:15, 30:11, 30:18, 33:5, 35:16, 36:20, 39:14, 44:25, 45:1, 50:6, 50:9, 50:13, 50:16, 51:2, 51:18, 59:15, 62:6, 62:9, 62:10, 62:11, 72:23, 79:5, 79:8, 79:25, 80:4, 80:5, 84:11, 84:20, 84:21, 86:21, 91:24, 97:5, 99:23, 102:21, 105:3, 108:7, 108:12, 108:25, 115:8, 120:14, 120:16, 122:3, 122:7, 127:10 seeing 123:17 seem 14:24, 32:9, 39:21, 44:9, 69:9, 120:23, 129:9, 129:10 seemed 19:18, 132:16 seems 5:20, 24:20, 32:11, 44:16, 45:8, 74:14, 102:17, 102:25, 103:8, 103:9, 115:24, 116:17 seen 26:18, 38:24,	92:25, 93:25 select 79:17 selected 8:7, 16:19, 24:9, 79:7, 79:19, 79:20, 79:22, 79:23, 79:24 selecting 79:12 selection 80:2, 82:4, 82:12, 82:15, 82:21, 83:4, 85:4 selzer 84:8, 96:19, 96:21, 97:19, 97:20, 99:5, 99:8, 99:19, 99:23, 100:9, 100:14, 100:17, 100:20, 100:21, 101:1, 101:4, 101:12, 101:14, 101:22, 102:14, 105:4, 107:7 semiconductor 1:4, 4:5 send 110:22, 112:1, 112:12 sending 39:5, 39:25, 40:10 sense 8:15, 61:14, 68:8, 78:15, 81:11, 89:12, 90:18 sent 29:24, 30:12, 34:9, 35:20, 37:4, 37:8, 37:10, 37:23, 38:24, 41:25, 45:16, 45:17,	45:24, 53:25, 57:9, 82:3 sentence 11:10, 13:10, 13:12, 17:17, 21:6, 21:21, 27:21, 38:9, 38:10, 39:23, 40:23, 44:21, 47:7, 48:21, 50:17, 63:14, 84:15, 101:25, 102:2, 103:7, 104:4, 106:7, 107:20, 107:21, 114:5, 114:17, 117:16, 129:22 sentences 22:15 separate 16:12, 17:2, 17:6, 29:9, 93:20 separately 29:9 sequence 62:24, 67:21, 68:1, 68:8, 68:10, 69:12 sequencer 10:19, 10:21, 11:4, 11:8, 11:13, 11:17, 11:24, 12:22, 12:25, 13:17, 16:13, 16:16, 17:3, 19:11, 19:15, 19:22, 19:25, 20:10, 50:6, 50:8, 100:3, 100:5, 100:11, 100:15, 100:18, 100:23, 101:1, 101:6, 101:17 serial 68:1 series 53:20, 68:14
--	--	---	--

set 47:9, 69:24, 78:3, 102:7, 135:14 several 8:2, 10:23, 22:10, 25:24, 34:12, 34:13, 74:7 shaded 35:19, 53:24 shader 5:24, 6:4, 6:8, 16:23, 59:11, 80:1, 108:10 shading 54:1 short 29:19 should 7:15, 53:14, 55:16, 76:25, 85:5, 87:10, 87:17, 88:16, 94:8, 100:11, 117:1, 120:20 showing 26:4 shows 6:25, 124:7 shut 40:18, 41:1, 42:14, 43:6, 44:17 shutdown 40:22, 41:4 shuts 36:19, 39:18, 41:8, 41:16, 42:22 sic 126:25 side 30:17 signal 80:4, 80:8 signature-5tmlq 135:20	signing 135:10 similarly 49:2, 109:10 simple 89:15 simpler 79:18 simply 11:3, 14:18, 49:21 simultaneously 69:5, 71:17, 71:21, 72:8, 72:25, 73:7, 73:13, 73:16, 73:25, 126:25, 127:11, 127:23, 128:10, 129:24 since 16:3, 21:16, 97:24 single 92:13, 93:7, 95:5 singular 6:13, 6:14, 6:15, 126:8 six 88:4, 109:9 skill 76:15 skilled 62:21, 64:4, 68:3, 70:6, 74:6, 88:11, 88:20, 93:10 skipped 27:12, 116:4 skipping 117:23 slight 24:22 slightly 48:18 slot 81:10 slots 80:23, 81:7,	81:11, 82:11 slower 116:16, 116:17, 117:21, 117:22, 118:21, 118:22 slowly 40:21 smaller 8:25, 9:4, 9:10 solution 36:11, 40:15, 41:6, 41:14 some 8:21, 9:23, 10:1, 23:13, 26:18, 26:19, 39:25, 56:21, 57:3, 60:14, 60:24, 61:24, 62:7, 62:23, 65:22, 67:3, 69:13, 70:8, 70:9, 70:13, 70:19, 70:20, 72:16, 72:20, 76:20, 77:13, 79:23, 81:13, 81:17, 82:20, 89:3, 90:12, 90:14, 90:24, 91:3, 91:4, 93:14, 93:24, 94:22, 95:18, 95:19, 98:9, 102:7, 103:10, 106:25, 109:16, 110:10, 111:5, 119:7, 120:10, 122:9, 122:10, 124:4, 124:12, 124:18, 124:19, 129:2, 129:9, 129:17, 129:25, 130:15, 131:12, 134:17 somebody 64:4, 68:3, 70:6, 74:6,	88:11, 88:20, 93:9 somehow 95:14 something 8:25, 24:14, 28:12, 30:10, 30:12, 34:11, 40:18, 46:4, 46:18, 51:7, 51:11, 52:4, 52:5, 54:11, 61:22, 64:11, 78:20, 89:2, 89:10, 89:11, 90:16, 93:6, 93:10, 94:10, 97:8, 97:21, 105:3, 115:17, 118:22, 129:1, 129:10, 129:16 sometimes 91:14, 91:16, 94:12, 95:8 somewhere 35:9, 60:16, 106:24, 107:2 soon 36:18, 38:8, 38:12, 38:17, 38:21, 39:17, 40:16, 41:7, 41:15, 42:21, 44:16 sorry 4:23, 7:15, 9:18, 12:6, 14:11, 14:15, 18:13, 19:14, 19:15, 20:13, 20:14, 25:5, 25:22, 29:6, 39:8, 39:9, 40:25, 41:4, 41:5, 43:22, 44:20, 47:6, 49:18, 50:18, 51:12, 52:18,
---	---	---	---

53:6, 53:23, 57:1, 69:24, 75:17, 76:10, 81:15, 91:5, 98:4, 100:24, 104:15, 105:20, 106:25, 107:17, 108:17, 114:4, 114:15, 115:9, 117:19, 122:17, 125:22, 128:21 sort 9:20, 78:23, 89:7, 121:23, 129:8, 129:9 sorts 92:21 sounds 74:22 source 36:1 space 12:17, 13:20, 13:21, 13:24, 14:4, 14:20, 15:20, 15:24, 16:2, 16:8, 16:10, 27:9, 27:13, 27:23, 42:2, 43:7, 43:24, 44:6, 45:4, 46:13, 96:4, 96:6, 108:11, 110:15, 111:2, 111:14, 111:16, 112:7, 112:16, 112:20, 112:21, 113:11, 113:14, 113:18, 114:13, 116:9, 119:8, 119:11, 119:14, 119:25, 120:7, 120:11, 120:18, 121:8 speak 26:22 speaking 36:9, 54:22,	57:24, 74:10, 124:12 special 105:15 specific 18:5, 46:1, 63:8, 92:14, 92:16, 93:16 specifically 49:24, 63:17, 67:16, 92:7, 109:1, 111:8, 112:8, 125:18, 133:17 specifics 69:19, 71:18, 82:24 specifies 34:24 specify 19:1, 32:25, 36:3, 87:19 speed 13:12 splines 105:25 spot 96:16 sram 14:4, 14:7, 14:10, 14:11, 14:13 stage 65:11 stages 60:8, 62:8, 62:13, 62:15, 63:11, 63:23, 64:14, 64:21, 73:16, 76:12, 86:19, 87:1, 87:12, 122:7, 122:15, 122:24, 123:4, 123:10, 124:8, 126:2, 126:17, 128:24 stages-of-comple- tion 84:24	stall 77:13, 77:14 stands 97:22, 104:23 start 11:22, 21:20, 33:3, 40:7, 49:21, 65:13, 70:11, 73:19, 78:10, 79:18, 87:3, 87:14, 108:19, 109:13, 123:15, 124:9, 124:14, 128:18 started 72:4 starting 21:17, 25:3, 25:24, 59:22, 65:14, 72:13, 80:21, 90:2, 101:9, 133:14 starts 5:22, 13:7, 13:19, 21:2, 26:22, 33:11, 40:9, 41:6, 44:21, 50:21, 70:18, 71:9, 72:15, 73:10, 125:20, 126:9 startup 37:20, 38:7, 38:23, 40:7, 105:7, 105:10 state 4:16, 44:4, 88:2, 91:3, 91:4, 91:7, 91:12, 91:15 statement 10:10, 86:2, 110:24, 111:19 states 1:1, 4:6, 6:19, 23:4, 99:5 stating 8:6	stay 118:9 stenographically 135:8 step 10:5, 37:22, 41:14, 71:4, 74:4, 89:9 steps 63:2, 74:8, 74:11 still 8:10, 24:22, 40:20, 48:9, 60:15, 65:19, 71:8, 72:20, 73:9, 74:1, 74:23, 78:12 stop 20:11, 22:13, 59:25, 70:22, 129:3 stopped 45:3, 63:14 stopping 22:19 storage 14:20, 16:5, 26:15, 26:19, 95:21, 95:22, 111:1, 115:25, 116:18, 116:20, 116:21 store 14:9, 14:17, 14:18, 18:21, 20:18, 20:20, 23:25, 59:10, 59:15, 62:7, 62:10, 65:17, 66:17, 70:1, 70:9, 72:12, 74:13, 74:15, 74:17, 75:12, 75:15, 75:16, 75:19, 75:23, 77:21, 80:3, 86:2, 88:10,
--	---	--	---

89:25, 91:1, 94:10, 94:14, 99:10, 99:13, 99:23, 107:2, 108:12, 110:16, 111:3, 111:6, 111:16, 111:24, 112:7, 112:20, 112:21, 114:13, 119:11, 119:15, 119:22, 120:7, 121:5, 121:9, 123:2, 123:3, 123:6 stored 13:24, 14:1, 16:3, 16:11, 24:19, 27:11, 27:23, 28:9, 29:15, 29:17, 30:2, 30:5, 75:14, 75:15, 104:20, 106:13, 106:17, 106:18, 106:23, 110:23, 112:15, 120:14 stores 13:17, 15:3, 15:15, 18:16, 19:20, 20:15, 21:1, 22:6, 23:4, 23:14, 26:25, 27:4, 94:20, 99:6, 99:20 storing 13:3, 14:4, 20:19, 21:22, 27:3, 27:16, 27:19, 92:17 street 2:5, 4:13 structure 69:14, 105:15, 105:24 structures 102:19 struggled 132:18	struggling 131:6 stuck 34:11, 34:13 studying 29:21 subclaims 10:23 subsequent 126:13 substitute 130:7 subtle 9:22 successfully 107:19 suffice 60:5 sufficient 113:13 suite 4:13 summarizing 117:8 summary 133:16 supervision 135:9 supervisor 105:6 supporting 63:10 sure 7:10, 8:9, 9:5, 9:15, 10:9, 11:1, 18:15, 21:10, 23:18, 28:8, 32:3, 32:16, 36:15, 41:14, 55:13, 56:1, 69:20, 72:3, 76:23, 79:3, 83:14, 92:24, 100:25, 103:13, 117:6, 121:8 sutcliffe 4:12	switch 61:17, 64:19, 65:16, 70:12, 72:1, 72:10, 72:23, 73:18, 73:20, 74:20, 75:7, 75:8, 76:11, 84:1, 84:18, 88:15, 88:18, 88:21, 89:22, 90:10, 90:15, 90:20, 90:21, 91:3, 121:25, 123:22, 127:8, 127:13, 127:23, 128:5, 128:13, 128:16, 129:12, 134:3 switched 72:5, 77:20, 84:8, 90:21, 90:23, 90:24, 91:1, 91:2, 123:19, 124:8, 125:24, 126:2 switches 126:17 switching 19:16, 36:8, 36:12, 59:14, 64:22, 65:18, 66:16, 69:25, 71:5, 71:12, 72:11, 72:17, 72:19, 74:12, 74:17, 75:12, 75:17, 75:20, 75:22, 75:24, 76:21, 85:14, 86:1, 86:3, 86:14, 86:25, 87:11, 87:24, 87:25, 88:7, 88:23, 89:4, 122:12, 122:14, 124:1, 124:2, 124:23, 127:17, 133:24	sworn 5:2, 5:4 synchronizing 105:8, 105:9 system 6:25, 7:2, 8:18, 38:2, 40:9, 40:17, 41:7, 43:5, 43:18, 44:3, 48:1, 77:5, 95:14, 101:23, 105:6, 107:8, 110:2, 112:5, 113:7 systems 107:3 T take 21:11, 59:1, 61:9, 63:19, 67:19, 74:4, 83:18, 87:5, 89:9, 93:4, 129:13 taken 59:5, 83:22, 108:3, 131:23, 135:5, 135:8 takes 88:8, 90:15 taking 4:12, 19:17, 76:16, 78:9, 123:23, 124:21 talk 10:17, 11:2, 12:3, 18:5, 21:13, 26:24, 36:5, 36:22, 36:25, 38:6, 38:23, 45:13, 69:19 talked 12:12, 12:23, 18:15, 19:4, 44:13, 49:16,
---	---	---	---

52:2, 58:23, 78:20, 91:24, 99:17, 125:1, 125:7, 127:22, 127:25 talking 9:24, 20:24, 21:3, 21:6, 21:9, 22:14, 23:3, 27:3, 27:9, 27:15, 37:5, 37:19, 38:2, 39:7, 45:21, 48:9, 48:10, 52:23, 53:7, 54:10, 54:21, 58:19, 58:20, 58:21, 68:3, 68:20, 74:12, 75:7, 75:25, 77:16, 78:22, 87:24, 88:8, 91:19, 92:6, 92:7, 93:8, 94:6, 95:11, 96:9, 96:10, 113:22, 114:23, 116:5, 117:17, 122:17, 125:10 talks 15:24, 52:2, 66:3, 114:1, 123:10, 124:14 task 49:18, 55:20, 56:8, 56:12, 56:18, 57:13, 57:15, 57:16, 57:18, 57:22, 57:23, 58:1, 58:3, 58:12, 101:24, 102:7, 102:23, 102:24, 103:1, 103:4, 103:11, 103:15, 103:16, 103:20, 103:21, 104:3,	104:21, 106:9, 106:11, 106:12, 107:9, 107:13, 107:22 tasks 48:19, 48:23, 48:25, 49:3, 49:6, 49:13, 49:15, 54:24, 55:3, 55:6, 55:8, 55:15, 55:24, 104:8, 104:19, 104:25 teach 100:10 teaches 118:13 technically 49:7 technologies 1:7, 4:6 tell 18:1, 23:15, 28:8, 48:20, 56:4, 85:12, 85:13, 87:11 temporarily 107:2 temporary 14:20, 16:5 ten-minute 59:2 term 7:8, 8:11, 9:15, 17:8, 43:2, 55:3, 55:16, 57:16, 61:25, 62:20, 67:24, 91:9, 92:4, 94:9, 99:12, 102:7, 109:16, 110:8, 119:9, 126:12 termed 6:4 terminology 73:2 terms 9:4, 11:14,	34:14, 52:6, 68:20, 91:10, 91:18, 95:3 testified 5:6 testify 5:4, 121:10 testimony 135:7 texas 2:6 text 44:25, 45:1, 45:2, 122:7, 124:5 textures 27:8, 27:22, 28:2, 29:14, 29:17, 30:4, 32:15 th 4:8, 135:15 themselves 4:16, 98:8 theories 124:6 theory 28:2, 122:14, 126:1, 128:4, 130:12 thereafter 135:8 thereby 68:15 therefore 23:8 therein 108:11, 110:15 thing 7:4, 31:15, 44:12, 46:25, 47:8, 47:11, 50:15, 90:3, 101:14, 106:7 things 8:21, 22:5, 30:3, 31:18, 32:16, 32:22,	37:22, 53:3, 58:25, 62:23, 64:14, 68:17, 90:23, 91:2, 93:11, 94:3, 94:16, 101:18, 118:16, 120:23 thinking 9:25 third 10:25, 35:15 thought 41:4, 72:3, 78:13 thread 84:19, 85:15, 86:4, 86:7, 86:8, 86:17, 88:22, 109:10, 110:21, 111:25, 112:11, 115:13, 115:20, 117:9, 119:1, 119:16, 121:24, 124:19, 124:20, 124:23, 124:24, 126:8, 126:9, 127:13, 134:2 threads 86:3, 86:6, 86:12, 123:19, 123:20, 123:24, 123:25, 124:2, 124:16, 125:2, 125:3, 125:12, 125:13, 125:14, 126:2, 126:7, 127:13, 128:8, 128:23, 134:4 through 15:9, 95:14, 99:22, 100:1, 113:2, 117:1, 122:4, 126:13, 131:9, 133:14 throughout 15:13, 55:11 till 90:10
--	--	---	---

time 4:8, 29:19, 42:2, 45:9, 59:4, 59:7, 66:23, 67:1, 68:6, 69:10, 70:13, 71:25, 72:9, 76:25, 77:19, 80:23, 81:7, 81:10, 81:11, 82:11, 83:16, 83:21, 83:24, 84:19, 86:3, 87:5, 88:18, 90:6, 90:15, 90:17, 108:2, 108:5, 110:19, 131:22, 131:25, 134:11 time-slot 83:6 times 108:25, 132:14, 132:18, 134:5, 134:12 title 62:10 today 4:10, 4:22, 5:1, 132:9, 132:14 today's 4:7 together 11:16, 11:23, 20:1, 26:7, 89:16 told 51:20 took 29:18 top 30:17, 36:6, 43:15, 50:16, 50:22, 78:4, 98:25, 111:19, 129:23 towards 53:13, 133:12	track 90:13 trademark 1:1, 4:7 transcript 135:6 transform 58:9 transformation 57:24, 58:2, 58:4, 58:10 transformed 30:24, 35:19, 53:24, 56:21, 57:3, 58:11 transforming 54:1, 58:7, 110:1 transition 45:22 transmitting 42:6, 42:7, 44:4 trap 88:2 traverses 102:18, 105:14, 105:24 trial 1:2 triangles 106:1 tricky 95:6 trigger 42:19, 44:18 triggering 38:10, 41:21, 42:8, 42:12, 42:13, 42:16, 42:17, 42:21, 42:23, 43:2, 43:4 true 55:18, 55:24, 110:25, 119:6, 128:12, 135:6 truth 5:5	truthfully 15:7 trying 13:5, 33:16, 129:14 turn 17:25, 48:17, 54:8, 57:20, 59:9, 62:3, 80:10, 96:17, 100:2, 132:10, 133:13 turned 38:11, 38:16, 38:22, 38:25, 39:18, 42:9, 42:21, 45:6, 45:11, 45:19, 97:16 turning 40:21 turns 36:19, 38:9, 40:17, 41:7, 41:16, 41:22, 42:23, 43:8, 43:16, 44:17 two 43:15, 43:17, 49:20, 63:25, 65:3, 65:25, 66:4, 66:22, 67:2, 67:9, 67:10, 68:17, 68:18, 69:7, 73:2, 73:15, 75:19, 77:17, 81:10, 89:16, 89:18, 90:4, 94:3, 95:4, 100:10, 100:19, 106:4, 109:7, 118:16, 120:23, 125:10 type 29:13, 29:16, 54:2, 65:7, 65:8, 73:5,	93:1, 114:6 types 24:12, 26:18, 28:9, 63:25, 65:3, 68:18, 73:3, 134:5 typewriting 135:9 typically 14:20, 16:5, 16:6, 16:7, 53:21, 62:22, 93:12 typo 87:18, 107:17, 132:17 U ulc 1:7, 4:6 ultimately 95:15 unclear 26:21 under 28:1, 45:23, 109:21, 109:23, 135:9 understand 11:19, 18:13, 29:4, 29:5, 30:9, 33:15, 53:1, 62:21, 63:1, 64:4, 69:1, 70:6, 72:9, 77:3, 79:11, 80:18, 81:12, 90:19, 98:10, 106:20, 107:4, 113:5, 119:25, 120:3, 126:15, 134:7 understanding 30:20, 32:6, 32:23, 32:25, 33:10, 55:14, 60:24, 66:15, 133:18
--	--	---	--

understood 13:21, 76:15, 99:9 undetermined 17:20 unfinished 123:19, 123:20, 124:1, 124:23 unfortunately 61:21, 118:19 unified 5:23, 6:4, 6:8, 16:23, 59:11, 108:10 unit 6:4, 6:9, 6:10, 6:19, 7:4, 7:7, 7:24, 8:20, 8:24, 9:6, 9:7, 9:18, 9:20, 9:23, 10:4, 10:7, 10:14, 11:16, 11:24, 20:2, 21:8, 59:11, 64:18, 65:25, 66:1, 75:9, 75:18, 79:6, 84:12, 85:24, 101:19, 102:6, 109:10, 110:22, 111:25, 112:12, 123:3, 123:5, 126:25, 128:13, 128:22, 129:24 united 1:1, 4:6 units 6:10, 6:11, 7:19, 7:23, 25:24, 51:16, 53:23, 62:25, 110:22, 112:12 unless 52:9, 77:13, 78:2, 108:10, 110:14, 111:13, 120:6	unshown 82:17 unstored 119:23, 120:9, 120:11 until 81:4, 112:5, 116:7, 120:6 usage 93:18 use 7:15, 7:16, 7:17, 14:13, 43:2, 48:18, 57:16, 93:14, 93:25, 94:11, 132:13 useful 36:14 uses 7:8, 7:20, 109:16 using 8:10, 8:11, 9:16, 15:25, 17:23, 55:3, 61:5, 62:20, 88:3, 110:8, 124:15, 127:1 V vague 73:1, 82:24 value 89:24, 89:25 values 89:16, 89:18, 90:4 various 59:13, 59:16, 60:4, 60:8, 61:19, 62:8, 62:13, 62:15, 63:11, 63:23, 64:14, 64:17, 64:21, 64:23, 68:22, 69:9, 71:13, 73:13,	73:16, 76:12, 84:2, 84:13, 85:25, 86:19, 87:1, 87:12, 88:15, 122:1, 122:7, 122:13, 122:15, 122:24, 123:4, 123:8, 123:10, 124:8, 126:2, 126:17, 127:16, 132:15 verbatim 40:24 versa 56:12 vertices 56:15, 110:1, 110:3, 110:4, 125:21 via 35:20, 95:21 vice 56:12 video 4:8, 4:11 videographer 2:20, 4:3, 4:10, 4:21, 4:25, 59:3, 59:6, 83:20, 83:23, 108:1, 108:4, 131:21, 131:24, 134:22 videotaped 1:12, 4:4 view 19:10, 19:21, 83:4, 94:18, 107:4, 123:23, 124:22, 125:5 virtue 20:5, 38:9 voice-identify 4:15 vram 28:14, 28:21 W wait 90:8, 90:10	waiting 82:2, 82:8, 89:1 want 8:9, 10:9, 12:13, 15:13, 19:24, 22:17, 22:25, 36:22, 36:25, 42:16, 46:3, 55:13, 66:24, 80:13, 85:9, 121:18, 123:9, 134:7 wanted 13:13, 14:7, 14:13, 28:17 way 8:11, 15:2, 28:10, 39:11, 39:12, 39:16, 52:24, 59:21, 60:3, 60:13, 62:25, 66:25, 72:21, 79:21, 82:23, 93:17, 94:1, 94:16, 94:23, 94:24, 105:4, 116:12, 121:12, 125:21, 126:6, 126:7, 130:2, 131:13 ways 101:22 we'll 28:16, 45:13, 46:17 we're 12:13, 15:22, 20:24, 21:6, 25:5, 27:3, 53:7, 54:21, 58:19, 58:20, 58:21, 70:14, 72:12, 74:12, 74:15, 74:17, 76:5, 77:16, 88:23, 89:4, 91:19, 92:6,
---	--	---	---

92:7, 93:8, 94:6, 101:9, 108:15, 112:10, 118:12, 122:20, 125:10, 131:24, 133:6, 134:23 we've 38:1, 38:24, 68:19, 78:21, 92:24 weren't 39:4 whatever 9:4, 15:13, 40:8, 58:6 whenever 43:24, 45:3, 46:13 whereas 65:19, 67:4, 81:9, 90:13, 90:14, 96:11, 119:23 whereby 67:17, 68:14 wherein 59:10, 123:3 whereof 135:14 wherever 94:19, 95:16, 123:9 whether 15:2, 15:15, 23:16, 23:19, 31:22, 111:10, 120:14, 125:14, 129:15, 133:23 whole 5:5, 14:21, 38:5, 53:12, 78:9, 88:12, 95:14, 133:7, 133:11, 133:19, 134:1, 134:16 withdrawn 6:18, 10:20, 31:12, 40:8,	41:20, 48:23, 55:5, 101:16 within 8:19, 9:8, 9:11, 9:14, 9:19, 15:17, 52:21, 57:2, 57:3, 63:17, 66:4, 67:17, 95:20, 96:22, 98:19, 98:22, 98:25, 100:14, 100:21, 101:4, 104:18, 108:9, 108:24 without 88:11, 110:17 witness 5:2, 43:22, 131:18, 135:14 word 7:20, 9:16, 9:17, 15:12, 29:19, 48:18, 61:3, 111:12, 117:19, 125:23 words 26:15, 32:10, 65:4, 116:4, 117:23 work 15:21, 15:23, 36:14, 78:14, 83:1, 83:3, 121:13 working 36:8, 36:12, 44:23, 47:3, 47:14, 47:22, 74:22, 74:23 workload 114:12 worst 33:12, 40:12, 42:19 wouldn't 9:14, 42:18, 43:20, 43:23,	60:25, 68:7, 89:4, 112:1, 121:13 wrap 134:21 writeback 74:9 written 60:19, 64:9, 64:13 wrong 47:6, 48:13, 107:16 wrote 11:22, 11:23, 107:6, 107:15, 107:16 X x 1:3, 1:9 Y y'all 22:13 yada 109:21 yeah 10:11, 11:11, 12:9, 12:15, 13:8, 15:10, 16:6, 18:5, 19:7, 23:21, 24:3, 24:16, 30:3, 31:4, 35:17, 36:17, 41:23, 45:1, 45:13, 46:24, 50:7, 50:23, 51:14, 51:19, 54:5, 55:2, 55:4, 55:15, 55:18, 59:18, 62:14, 62:20, 63:13, 72:11, 73:1, 79:21, 80:5, 80:17, 82:10, 86:20,	95:12, 98:21, 99:25, 105:17, 105:22, 108:18, 109:3, 109:16, 111:25, 115:1, 117:16, 117:20, 120:10, 122:22, 123:12, 128:20, 129:22, 130:23, 131:4, 132:18, 134:1, 134:11 . .1 105:5, 105:13 .1234 2:7 .1500 2:16 0 00922 1:10, 4:7 01 1:15, 4:1 02 4:9 02116 4:14 1 1 108:5, 129:17 10 59:4, 59:7, 79:25, 80:6, 80:12, 80:14, 81:6 100 91:10 1001 5:11, 59:9 1003 5:12, 132:5 1005 109:6, 109:12, 109:14, 110:7, 112:4, 113:2,
---	--	---	--

Transcript of Hanspeter Pfister, Ph.D.
Conducted on February 16, 2024

69

115:18 1006 109:7, 113:3, 126:20 1007 7:10 1008 11:21, 21:7, 21:9, 26:3, 63:5 1009 97:18 1010 85:19 105 133:13 106 114:17, 114:24 107 114:5, 115:2 109 115:11, 133:15 11 59:9, 62:6, 83:21, 83:24, 84:1, 87:25, 115:19, 117:8, 122:19, 122:22, 122:25, 123:1, 125:3, 125:15 114 96:17, 122:3, 123:9, 123:18 117 126:19 118 128:21, 130:24 119 126:15, 128:21 12 108:2, 118:3 125 84:7, 132:10 13 130:20 132 3:5 135 1:22	140 108:14, 108:17, 109:9, 109:20, 110:20, 113:11 156 53:19 157 7:17, 7:18, 12:5, 12:10, 13:5 158 8:1, 12:5, 12:10, 12:20, 13:3, 13:5, 13:7, 13:18 159 30:16, 32:2, 33:4, 33:18, 35:14, 39:1, 39:14 16 1:14, 4:1, 4:8 160 33:2, 33:11, 36:6, 37:2, 39:7, 39:9, 39:23, 40:23, 43:14, 54:7 167 5:23 169 5:15, 6:3 17 135:15 170 25:3, 25:7, 25:24, 27:6 172 10:13 177 51:12, 51:13, 52:12 178 11:6, 11:15, 12:8, 13:11, 20:25 179 50:21	184 48:18, 48:21, 54:23 189 50:20 19 50:16 194 80:10 196 80:21 2 2 5:18, 129:4, 131:22, 131:25, 134:24, 134:25 2.3 87:23 2000 4:13 2006 130:19 201 62:3, 65:1 2023 1:10, 4:7 2024 1:14, 4:1, 4:8, 135:16 2027 135:17 212 87:23 213 133:14 22 63:7, 66:9 220 117:11, 133:15 222 4:13 229 96:18, 98:5 23 63:7, 66:9 236 100:2, 101:10,	102:2, 107:6 237 100:9 24 130:20 25 135:17 256 84:5, 84:15, 132:11, 132:20, 132:25, 133:6 26 131:22 3 300 2:14 31 108:5 32 108:2, 131:25 3580 2:14 36 115:7 38 134:24, 134:25 3d 58:8, 58:9 4 4 110:2 4.3 105:5, 105:13 4.4 97:16, 97:20, 98:25, 102:15, 104:13, 104:16, 104:17, 105:18, 105:23 4.5 104:14 40 83:21, 105:5, 106:15 42 110:2
--	---	--	--

Transcript of Hanspeter Pfister, Ph.D.
Conducted on February 16, 2024

70

43 102:15, 105:21, 107:9 44 83:24 45 59:4 454 5:12, 5:17, 5:24, 10:17, 11:18, 11:25, 17:10, 20:7, 59:9, 84:1, 108:7 5 50 110:6 524719 1:21 55 108:8 550 109:13 56 59:7, 114:10, 114:23, 114:24 57 114:4, 114:22 58 115:11 6 6.3 109:13 60 44:20, 122:2, 123:17 62 128:21, 129:23 67 108:18, 108:20 68 108:15, 108:19 69 111:19 7 713.229 2:7	77002 2:6 8 81 5:16 82 25:4, 25:6 83 6:24, 26:10 858.314 2:16 86 69:24 860 6:20, 7:1, 8:3, 8:7, 8:16, 8:18, 8:24, 8:25, 9:7, 9:8, 9:12, 9:19, 10:7, 10:15, 10:20, 69:24, 71:18, 75:1, 76:5, 77:9, 78:4, 93:5, 93:6 88 50:6, 50:22 89 50:5, 50:22 892 10:13 9 9 1:15, 4:1, 4:9 90 97:16, 109:24 910 2:5 913 126:16 92 110:6 92130 2:15 95 80:11 96 80:21	97 80:13, 81:2 98 62:4 9b 84:12	
--	---	---	--