UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

LG ELECTRONICS, INC.,	)	
	)	CASE IPR-2015-00325
Petitioner,	)	Patent 7,742,053 B2
	)	
-VS-	)	CASE IPR-2015-00326
	)	Patent 6,897,871
ATI TECHNOLOGIES ULC,	)	
	)	CASE IPR-2015-00330
Patent Owner.	)	Patent 7,327,369 B2

CONFIDENTIAL - PURSUANT TO THE PROTECTIVE ORDER

VIDEOTAPED DEPOSITION OF LAURENT LEFEBVRE

ON FRIDAY, NOVEMBER 13, 2015

IN MONTREAL, QUEBEC, CANADA.

VIDEOGRAPHER: DAVID OXILIA

COURT REPORTER: CHERIE KLEIN

DIGITAL EVIDENCE GROUP
1726 M Street NW, Suite 1010
Washington, DC 20036
(202) 232-0646

APPEARANCES:

FOR PETITIONER:
MAYER BROWN
BY: BRYAN NESE, ESQUIRE
1999 K Street NW
Washington, D.C. 20006-1101
Tel: 202-263-3266
bnese@mayerbrown.com

FOR PATENT HOLDER
STERNE KESSLER GOLDSTEIN FOX
BY: JONATHAN TUMINARO, PhD, ESQUIRE
TYLER J. DUTTON, ESQUIRE
1100 New York Avenue, NW
Washington, D.C. 20005
Tel: 202-371-2600
jtuminar@skgf.com
tdutton@skgf.com

ROBINS KAPLAN
BY: AARON R. FAHRENKROG, ESQUIRE
BRYAN J. MECHELL
800 Lasalle Avenue, Suite 2800
Minneapolis, MN 55402
Tel: 612-349-0642
afahrenkrog@robinskaplan.com
bmechell@robinskaplan.com

INDEX

WITNESS: LAURENT LEFEBVRE

EXAMINATION	PAGE
BY MR. NESE	5

INDEX OF EXHIBITS

NUMBER	DESCRIPTION	PAGE
Exhibit 1	Laurent Lefebvre Declaration In IPR2015-00325	6
Exhibit 2	Laurent Lefebvre Declaration In IPR2015-00326	6
Exhibit 3	Laurent Lefebvre Declaration In IPR2015-00330	6
Exhibit 4	R400 Architecture Proposal from Steve Morein	46
Exhibit 5	R400 Sequencer Specification Version 0.4	71
Exhibit 6	Version 2.0 of the R400 Sequencer	87
Exhibit 7	United States Patent 7,742,053	112
Exhibit 8	Independent Claims of Patent 7,742,053	112
Exhibit 9	Log File For the R400 Sequencer Emulator Directory	147

UPON COMMENCING AT 9:03 A.M. IN MONTREAL, QUEBEC, CANADA.

BY THE VIDEOGRAPHER: Good morning. This is tape number one of the videotaped deposition of Laurent Lefebvre, taken by Bryan Nese of Mayer Brown in the matter of LG Electronics Inc. versus ATI Technologies ULC. This case is being held in the United States Patent and Trademark Office before the Patent Trial and Appeal Board. The case number is IPR2015-00325, 326 and 330. This deposition is being held at the Omni Hotel in Montreal, Quebec, Canada on November 13, 2015. The time on the video monitor is 9:03 a.m. and my name is David Oxilia. I am the legal videographer for Digital Evidence Group. The Court Reporter is Cherie Klein also with Digital Evidence Group. Will Counsel please introduce themselves and state whom you represent.

MR. NESE: This is Bryan Nese with Mayer Brown LLP for Petitioner LG Electronics Inc.

MR. TUMINARO: Jonathan Tuminaro from the law firm of Sterne Kessler Goldstein & Fox on behalf of the patent owner, ATI Technologies ULC.

MR. FAHRENKROG: Aaron Fahrenkrog from Robins Kaplan LLP on behalf of the patent owner ATI Technologies ULC.

MR. MECHELL: Bryan Mechell from Robins Kaplan LLP on behalf of the patent owner ATI Technologies ULC.

MR. DUTTON: Tyler Dutton from the law firm Sterne Kessler Goldstein & Fox on behalf of the patent owner, ATI Technologies ULC.

BY THE VIDEOGRAPHER: Thank you. The court reporter will now swear in the Witness.

WITNESS SWORN

EXAMINATION

BY MR. NESE:

Q. Good morning, Mr. Lefebvre.

A. Good morning.

Q. I apologize if throughout the day I get the pronunciation of your name incorrect. My French is terrible, so I will thank you to excuse me. Sir, you are here because you submitted three Declarations in these IPR proceedings. Is that right?

A. That is correct, yes.

Q. Okay. And I would like to hand you those now. So we will mark as Exhibit 1 your Declaration in IPR2015 325. And unfortunately I am not going to have enough samples for everyone today. Mr. Mechell, I do have one for you. I would also like to mark as Exhibit 2 Mr. Lefebvre's Declaration in IPR2015-00326, and I will mark as Exhibit 3 Mr. Lefebvre's Declaration in IPR2015 330.

Exhibits 1, 2 and 3 were marked for identification.

BY MR. NESE:

Q. Mr. Lefebvre, could you please take a look at these Declarations and confirm that they are indeed accurate copies of the

Declarations that you submitted in these proceedings?

A. Yes, these are accurate copies of the Declarations that I signed.

Q. Thank you. Did you write each of these Declarations?

A. I did not write these Declarations. These were written for me by Lawyers working for ATI AMD. I did review, however, each and every one of these Declarations, read through them all, provided some comments and corrected inaccuracies, and as well as reviewed the calendar, dates and names of everyone cited in these Declarations.

Q. But these Declarations are written in your words; correct?

MR. TUMINARO: Objection. Form.

THE WITNESS: They are written in whoever wrote them words, but I did review them for accuracy.

BY MR. NESE:

Q. Did you receive any compensation for

signing these declarations?

A. I did not receive any compensation for signing these Declarations, outside of my normal salary from AMD, I mean.

Q. So no consulting fee?

A. No, not that I am aware of. Just my regular salary.

Q. I would hope if you got a consulting fee, you would be aware of it. I would want to be.

A. I did not receive a consulting fee, no.

Q. Sir, other than the documents you referred to in your Declarations, did you review any documents in the course of preparing these Declarations?

MR. TUMINARO: Objection. Form.

THE WITNESS: I reviewed the documents cited in the Declarations, yes.

BY MR. NESE:

Q. Did you review any other documents other than those cited in the Declarations?

A. I did not review directly any other documents other than those cited in these Declarations.

Q. Did you indirectly review any other documents other than those cited in the Declarations?

A. I reviewed perforce logs that we used to build the calendar, the diligence calendar, but no. I mean, I just reviewed the Declaration documents and that's pretty much it, as well as the Declarations themselves, obviously.

Q. Are there any errors in your Declarations, that you are aware of?

A. I am aware that a citation is missing or mislabelled in one of my Declarations that I found while reviewing it after the fact, and that's in the 053 Declaration, to be exact. It's missing an Exhibit in the Exhibit list.

Q. You are referring to Exhibit 1?

A. I am referring to Exhibit 1.

Q. What is the name of the citation

that's missing? Excuse me. I believe you said that there was an Exhibit missing. Which Exhibit is missing?

A. On page 29, Exhibit 2074 is cited, and it is not present in the list of Exhibits of that Declaration.

Q. Thank you for pointing that out. Are there any other errors that you are aware of in your Declarations?

A. I am not aware of any other errors in my Declarations.

Q. Mr. Lefebvre, have you ever been deposed before?

A. I have never been deposed before, no.

Q. Okay. If it's okay with you, I would like to go over just a few ground rules for the deposition, to make things go a little smoother today. Is that all right?

A. Yes, that's all right.

Q. Okay. As you may have noticed there is a Court reporter here, Ms. Klein. Ms. Klein

is going to be taken down everything that I am saying as well as everything that you are saying, and you heard her mention earlier it would be helpful if we both speak slowly. Is that okay with you?

A. Yes, that's okay with me. Is this speed correct? Okay.

Q. And also, because she needs to take down everything we are both saying, we need to have verbal statements on the record. So today can you make sure to give me verbal responses to the questions that I will be asking you?

A. Yes. I will make every effort to give you a verbal response to any questions that you ask me.

Q. And also it would be helpful for Ms. Klein if we don't talk over each other. I will try to do my best not to talk over you. Can you try to do your best not to talk over me?

A. Absolutely. I will do my best.

Q. Wonderful. I typically take breaks every hour or so. If you need a break before

that, for any reason, please feel free to let me know.

A. Yes.

Q. The only thing I would ask is if we are in the middle of a question, I just would like you to fully answer a question before we go on the break. Is that okay?

A. Yes.

Q. From time to time you may hear your Lawyer, Mr. Tuminaro, make objections to some of the questions that I am asking. However, unless he specifically instructs you not to answer a question, I still need an answer to the question. Can you do that for me?

A. Yes, I will answer the questions unless advised not to by my Counsel.

Q. Great. The last thing is that the Patent Trial and Appeal Board has a rule that prohibits Witnesses from talking about their testimony with their Lawyers while a deposition is going on, and that includes during the breaks. Will you abide by that rule today?

A. Yes, I will obey by that rule today.

Q. Thank you. Sir, what did you do to prepare for today's deposition?

A. For today's deposition I had several discussions with the Attorneys team. Like stated before, I reviewed the specifications listed as Exhibits in my Declaration. I also reviewed Declarations for -- reviewed the perforce logs to make sure that all of the dates and change list numbers were correct as best to my knowledge and obviously, like I said, reviewed the Declaration and also reviewed a little bit of the RTL code that was created at the time.

Q. When you say you reviewed the RTL code that was created at the time, what are you referring to?

A. I am referring to the sq-thread_arb.v file which is presented as an Exhibit to 074 and 2097 in my Declarations.

Q. So did you review any other RTL code other than the Exhibits that are in your Declarations?

MR. TUMINARO: Objection. Form.

THE WITNESS: Not that I recall, no.

BY MR. NESE:

Q. Okay. You also said you had discussions with your Lawyers, and I don't want you to reveal any of the substance of the discussions that you had with your Lawyers, but can you tell me who you met with?

A. As best of my knowledge, I met with everyone in the room today. More often with Jonathan, because he was my primary contact, but I had like discussions with every Lawyer present in the room today. Bryan, Tyler, Aaron and Jonathan.

Q. Except for me; right? We didn't have any discussions before this; right?

A. Outside of today, no. I have never talked to you before.

Q. You haven't had the pleasure yet.

A. Not yet.

Q. Did you meet with any other Lawyers other than the four Attorneys who are

representing ATI Technologies here today?

MR. TUMINARO: Objection. Form.

THE WITNESS: I had some discussions with the internal Counsel of AMD, which I don't know if they are Lawyers or not. Probably, but outside of this Counsel and the AMD representative of the legal departments; no, I did not have any other discussions that I recall.

BY MR. NESE:

Q. And when did you have these meetings with your Lawyers to prepare for your deposition today?

A. We met once in person last year in Montreal, then two more times at my house. If I recall correctly it was some time last year and the year before. And we also met yesterday again in Montreal.

Q. And was it Mr. Tuminaro that you met with a year ago in Montreal?

MR. TUMINARO: Objection to the extent it calls for work product, and I am going to instruct the Witness not to answer.

THE WITNESS: I am going to follow my legal counsel and not answer that question.

BY MR. NESE:

Q. Are you currently employed?

A. Yes, I am currently employed.

Q. Who is your current employer?

A. My current employer is AMD Incorporated in Markham -- Toronto, Ontario.

Q. And what is your current position at AMD?

A. My current position is a fellow position in hardware graphics architecture.

Q. And what are your current responsibilities as a fellow in hardware graphics and architecture?

A. My current responsibilities include specifying and measuring performance of the various graphics products that AMD does today. I am also responsible to answer questions by customers and meet with them to go over hardware graphic specifications. I am also the primary contact for the graphics team to the internal AMD

APU team, so to understand and answer any questions the SOC teams might have in the -- during the creation of the SOC's. And I am responsible to give the executive boards as well as AMD executives performance, power and area estimates for a variety of the graphics processors that AMD develops internally and externally.

MR. TUMINARO: Bryan, at this time I am going to designate the transcript confidential under the Protective Order, to the extent confidential information is coming up.

MR. NESE: Okay.

BY MR. NESE:

Q. What do you mean by SOC's?

A. So SOC is an acronym that means system on a chip. It's a new flavour of graphic -- of chips that we do at AMD where everything from the CPU to the graphic core to multimedia components is integrated into a single silicon device to save costs, and area and power.

Q. Do your current responsibilities as a fellow in hardware, graphics and architecture require the writing of source code?

A. My current responsibilities do not require me to write any RTL source code nor C sim per se;no -- C simulation -- C++.

Q. How long have you had the position of a fellow in hardware, graphics and architecture?

A. I believe, if I am recalling correctly, I was promoted in September 2013.

Q. When did you begin working for AMD?

A. I started working for AMD in September of 2000 right out of school, where I moved to Boston to work for the Marlboro office. Actually, back then it was called ATI Technologies, to be correct.

Q. Thank you, and I understand the two companies merged in 2006. And at that time did you begin working for AMD?

A. So to be completely fair, I was an ATI employee from September 2002 to January --

sorry -- from September 2000 to January 2003, at which point I moved to Montreal and became a contractor who worked for ATI but as an external contractor. And then in September of 2007, if memory serves me right, then AMD, which at the time bought ATI before, rehired me as a permanent employee from Markham.

Q. So you mentioned you began working for ATI right out of school. Do you hold any degrees?

A. I hold two degrees. One of them is an engineering degree in computer science from the Ecole Polytechnique of Montreal right here, and the second degree that I own is a master's degree in computer graphics from the University of Montreal.

Q. And when did you get your engineering degree in computer science?

A. So my degree in computer engineering and computer science would have been granted around June of 1998.

Q. And it that when you got your ring

on the pinky finger of your working hand?

A. That is about when I got my ring, maybe a bit before, because the ceremony occurs a bit before the graduation.

Q. And after June 1998 did you go to grad school full-time, working on your masters?

A. Yes. Right after I was -- I received my degree from Ecole Polytechnique I joined full-time the computer science lab in University of Montreal to work on my masters for two years.

Q. Mr. Lefebvre, when did you first hear about these proceedings?

A. I don't really recall the exact date, to be honest. It -- yes. Probably two, three years ago. Around that time. I cannot be more precise than that. I don't really recall.

Q. How did you first hear about these proceedings?

MR. TUMINARO: I am just going to caution the Witness not to disclose the substance of any communications you may have had with

Counsel. You can answer how you learned about these proceedings.

THE WITNESS: If memory serves, I think it's either Jonathan or Bryan that called a meeting through AMD legal department that I was needed to, you know, depose, to talk about --

MR. TUMINARO: Don't talk about the substance. Just how did you learn about it.

THE WITNESS: Yes.

BY MR. NESE:

Q. You believe it was either Mr. Mechell or Mr. Tuminaro that first told you about these proceedings two to three years ago. Is that right?

A. That is I what I recall; yes.

Q. I would like to talk about the industry you are involved in for a little bit. Do you have an understanding of how graphics processors are designed?

A. I have a very good idea of how graphics processors are designed; yes, given this is what we have been doing for 15 years now.

Q. And can you walk me through how engineers typically design a graphics processor?

MR. TUMINARO: Objection. Form.

THE WITNESS: So the typical flow of development is the writing -- in development starts with the writing of architectural specifications where architects meet and discuss, you know, the various features that are needed in the chip. After specifications are drafted we go and write C++ code to simulate these specifications and start understanding the rules and capabilities, and at the same time RTL developers start to write the register transfer layer code for the chip. This code is then synthesized and turned into what we call gate level net lists, which are sent to physical design for place and route, and finally send to Fab's for fabrication.

BY MR. NESE:

Q. So you mentioned that there are -- there is C++ code for simulation. Is that right?

A. That is correct. AMD uses a C++

model that we call the reference model to be used as a check environment so that the people that write the RTL code can verify for themselves against a golden model.

Q. You also mentioned that there is register transfer level code written, RTL. Is that right?

A. Yes. In order to produce a chip the main element that is needed is the source register transfer layer code.

Q. Will the same engineer or a group of engineers usually write -- let me start over. My apologies. Will the same engineer or engineering team write both the C++ code and the RTL code?

A. At AMD we strongly try not to, so that the same person cannot introduce the same error in both the register transfer layer and the code that is used to validate that code. So typically the architect is the person that's going to write the specification as well as the C simulation version of that block, and another person or another group of persons would write

the register transfer layer.

Q. I would like to know more about this synthesis process that you described. Can you give me some more detail about what's involved in the gate level net list?

MR. TUMINARO: Objection, form.

THE WITNESS: Synthesis is a process -- is an automated process by which a tool takes the register transfer layer source code and transforms it into gates which are, you know, "and gates", or "or gates" or whatever gates are available into the physical library that the PD team will use to fabricate the chip. This flow is fully automated, and there is no human intervention needed to go from the RTL to the gate level simulation. There is also a formal validation tool that is provided by the company that does the synthesis tool to formally validate that the generated gates are equivalent to the RTL code.

BY MR. NESE:

Q. How long does this synthesis process

take?

A. The synthesis time varies based on the size of each block. It can go from anywhere between one hour to 24 hours, depending on really the size of the design that you are trying to synthesize.

Q. And how does a company know that its RTL code has been synthesized properly?

MR. TUMINARO: Objection, form.

THE WITNESS: Like I said, the company that provides the synthesis tool also provides a formal equivalence tool that verifies -- that the generated gates are fully -- formally equivalent to the RTL code. So after the synthesis tool is run, we run this tool and, if any errors or any issues are found, then the tool is flagging them and you know, we file a bug to the company that writes the synthesis tool so that it can be fixed. I mean, once you pass the formal compliance tool you are guaranteed that the gates that you generated are equivalent to the RTL.

BY MR. NESE:

Q. So if the compliance tool finds a bug, who fixes it? Is it the team that wrote the RTL code, or is it synthesis team, the validation team, the Fab?

A. So typically the tool will just file warnings that -- meaning that some gates are -- could be different than what the code really wanted to do. So you can go two ways with this. Once you find one of this, the RTL code team could just rewrite the portion of the code that is problematic in such a way that it's simpler to synthesize and did not raise this issue. We have also found issues with the tool itself, in which case the company that provides the tool needs to go and fix the issue in the tool.

Q. In a typical design process how often does the validation tool find a bug and then that requires the RTL team to rewrite some of the code?

A. I don't really know how often. I know it occurs, but since I was not the person

writing the actual RTL code, I typically don't deal a lot with these types of issues. They are mostly dealt with by the design RTL team. I know these issues occur, but I can't provide you with an actual number as, like I said, this is not my speciality.

Q. So after the synthesis you have this gate level net list. And is there a name for that file?

A. We just typically call it the net list.

Q. Okay.

A. With like a number, so that people know which net list we are talking about.

Q. And is it the net list that get sent to the Fab for reduction to silicon?

A. No. So basically the net list is just the list of gates and the connectivity between those gates. There still needs to be a place and route flow that needs to be provided to them. So these gates present in the net list get placed into the area by a physical design team so

that fabrication can occur. So there is a PD step that is required to place and route those net lists before fabrication. That process is not making any modifications to either the connectivity or the type of cells used. It's really just the process by which the gates present in a net list are placed in a 2D map so that fabrication can occur.

Q. So to put things maybe in a mechanical engineering term that I can understand, would it be like creating a blueprint of the chip?

MR. TUMINARO: Objection. Form.

THE WITNESS: It is creating a 2D representation of the cells in, you know, a map of the chip for which in each coordinates you can place only one gate and so you need -- given the list of gates that you have in the net list, someone needs to pick a place for each of these gates based on the connectivity so that, you know the rule of -- that only one gate can be present in any 2D physical location on the chip is

obeyed.

BY MR. NESE:

Q. And does the Fab use that file to then start production of the chip?

A. Yes. It's my understanding that the fab uses these files that we call them -- and we call them tape out files to start the fabrication of the chip.

Q. What is a GDS2 file?

A. That is, to my knowledge, this tape out file that we just discussed.

Q. And how is it again that the RTL file get converted into a GDS2 file?

A. Like discussed previously, the first step is to run the synthesis tool, which is an automated tool that takes the RTL file and generates gate level net lists for each of the blocks of the file. And these gate level net list files are then turned over to the PD team which are going to place these gates on a 2D map and generate the GDS files to be sent to the fab.

Q. So how does the PD team place the

gate on a 2D map to generate the GDS file?

A. Again, I am not an expert in PD. That's really a job on its own, but my understanding is that given the connectivity and the type of cells that are present in the net list, they have to obey the rule that only one cell can be present in any physical location. So they start with a blank canvas of silicon, and then they have to place all the cells or the gates that are present in the net list onto that canvas so that the GDS file can be generated and sent to the Fab. It's really a process that we really call place and route, because it's really just placing the cells and making sure there is a route between all the connections that are specified in the net list.

Q. Is the PD team manually doing that?

A. The start process is automated, so the PD tools can do a first job of trying to place the cells, but a lot of the PD work is manual, unfortunately, and so they need to go and tweak the placement of the cells to find the

ideal map that gives them the smallest area without having congestion issues, because you cannot do like 20 wires in the same place. There is only one.

Q. How long does that place and route process take typically?

A. Place and route? Again, it depends on the size of the block and the complexity of that block, so the amount of connectivity that needs to be done and the number of cells you are trying to place. I would say, if I recall correctly, it's about a month or two of work for every block. And a lot of the teams work on each block in parallel, so you are talking about one to two months of schedule time to do the full place and route of a chip.

Q. And how many engineers would typically be involved in that process?

A. Again, I am going to talk from my knowledge, and I am definitely not an expert in PD. I don't claim to be one. My understanding is that it's at least one or two persons per

block of the chip.

Q. I would like to talk about your time at ATI now. What was the first project you worked on at ATI?

A. The very first project I worked on at ATI was the R400 chip graphics processor.

Q. And how long were you working on the R400 chip?

A. I started working on the R400 right when I joined ATI and, I would say, worked on -- I mean, that was my sole responsibility all the way until the project was converted into the Xenos program, on which I continued to work on until it got taped out. If memory serves, that's around 2004, 2005, around these dates, but it's far away.

Q. And you first started at ATI in 2001? Am I remembering that correctly?

A. I first started at ATI in September of 2000.

Q. Was there a lead designer on the R400 project?

A. My recollection of this was that responsibility was shared between Steve Morein and Andrew Gruber.

Q. So were Mr. Morein and Mr. Gruber the lead designers for the R400 project?

A. I wouldn't call them "lead designers", but they were definitely the lead architects which would, you know, define the main blocks, the infrastructure and the connectivity.

Q. So during 2001 what percentage of your time working at ATI was spent working on the R400 project?

A. 100 percent.

Q. Is that true for 2002 as well?

A. Like stated previously, R400/Xenos was my sole responsibility at the time. So I did not participate in any other programs during that time.

Q. At what point did you start working on the Xenos chip?

A. I believe it was some time in late 2003, 2004. I mean, for us the transition was

seamless because it was the exact same design, so I don't have a firm recollection on the transition, because for me, as the SQ lead, the design was the exact same, identical. So there was nothing really for me to do to accommodate for this new design, for this new chip. I just continued working on the sequencer.

Q. How is the Xenos chip different from the R400 project?

MR. TUMINARO: I am going to object. To the extent it call for third-party confidential information, I will caution the Witness not to disclose any third-party confidential information.

THE WITNESS: I can't really answer that question without giving confidential information from our external customer.

BY MR. NESE:

Q. Okay. Fair enough. So I believe you mentioned earlier that 100 percent of your time at ATI was spent working on the R400 project and the Xenos project between 2000 and, let's

just say, 2004. Is that right?

A. Yes, that is right. Obviously, I mean, there are some training tasks that we are required to take as employees or HR tasks, but all of my technical time was spent on working on the R400/Xenos during that time.

Q. And how do you know that for sure?

A. Like I said before, I was hired for the R400, so that's my recollection of what I did. I mean, it's a long time ago. I don't remember working on anything else, and the whole office at the time was also pretty much working on the R400 as well, because the other chips were done on the West Coast, which was a completely different team. And so I am pretty confident that the R400 and the Xenos chips were the only things I worked on during that time frame.

Q. Are you relying on anything else other than your own recollection to be confident those were the only projects you were working on at the time?

A. So I am also relying; yes, on the

log files from our perforce depot server, which is a file revision control tool which logs all of the activity on any given files of the project, logs who wrote any document, what changed within that document, and the time and date of every entry.

Q. Did you look for perforce log files from 2000 to 2004 other than the log files associated with the R400 Xenos project?

MR. TUMINARO: I am going to object to the extent it calls for work product. Don't disclose anything you may have done at the direction of Counsel.

THE WITNESS: I don't recall looking at perforce log files other than during, you know, the log files that are presented here in my deposition. After I transitioned to other programs, then obviously I started working on other projects, but no, I don't recall working on any other log files other than those pertaining to the R400 project.

BY MR. NESE:

Q. Did you take any vacations from 2000 to 2004 while you were at ATI?

A. Yes, I did about three -- 15 days of vacation every year.

Q. And when were those vacations in 2001?

A. In 2001 those vacations were around the spring timeframe where I went to Hawaii.

Q. How long were you in Hawaii?

A. If I recall correctly, about a week and a half or so.

Q. Was that your only vacation in 2001?

A. That was the longest that I recall. Like I said, I typically tried to take all my vacations, so I must have taken other days here and there. I don't recall exactly where or when. Probably some at Christmas.

Q. What about in 2002? Did you take a vacation then?

A. 2002? I don't recall exactly when. Probably in the summer but, like I said, I tried to take all my vacations yearly, so about 15 days

every year.

Q. And did you take 15 days vacation in 2003 as well?

A. I don't recall otherwise. Like I said, I mean, yes. I think -- I think I try to take them all all the time or else I lose them.

Q. Do you remember when your vacations were in 2003?

A. Probably. I mean, at least a couple of days at Christmas, around Christmas, and either in 2002 or 2003, I know that I went to Europe, so that would have accounted for another week around September. Also in 2002 my first girl was born in February, so I would have taken four days then.

Q. Do you know whether Steve Morein was working on the R400 project 100 percent of the time?

A. I cannot attest that he was working on it 100 percent of the time because I don't know his full schedule. I do know he was working on it a lot because that was his project and he

was really working on it a lot. I can't say for sure it was 100 percent of the time or not. I don't know his schedule.

Q. Do you know whether he had other projects at the time?

A. I do not know.

Q. And what about Mr. Andy Gruber? Do you know whether Mr. Gruber was working on the R400 project 100 percent of his time from 2000 to 2004?

A. Mr. Gruber -- it is the same thing than for Mr. Morein. I know he was working a lot on the R400, and we collaborated a lot on the sequencer because he was doing the shader pipe architecture, so we had lots of collaborations, lots of meetings. I don't remember or know if he had any other projects that he was tied to other than the R400.

Q. Was Andy Skende working on any projects other on the R400 project at the time?

A. I believe that Andy Skende very early was working on the R200 project, which was

the chip that ATI was doing before I joined, but once Andy transitioned to the R400 shader pipe, which I don't remember exactly when that occurred, I believe this was his full-time work task.

Q. What makes you believe that this was his full-time work task?

A. Again, I cannot guarantee his schedule because, you know, I can only tell you what I have been doing. But we had very, very many interactions with Andy, because he was the shader pipe designer, and helped him a lot with the shader pipe C code implementation, so the verification model. And based on these interactions, I know if Andy was not working on R400 full-time, he was working on it a lot of the time.

Q. But it's possible he was working on other projects too; right?

A. Yes, it is possible that he was working on other things that I was not aware of.

Q. And it's possible Mr. Morein was

working on other projects at the time too, isn't it?

A. Yes. I am not, you know, keeping track of the calendar of Mr. Morein, but I can tell you again that, given the number of interactions we have had -- I have had with Andy Skende, Steve Morein and Andy Gruber, they were dedicating a lot of their time to the R400 project, to the extent I would be pretty comfortable saying if it wasn't the sole program they were working on, it was very close to be.

Q. It's also possible that Mr. Gruber was working on other projects at the time, isn't it?

MR. TUMINARO: Objection. Asked and answered.

THE WITNESS: Yes. Like I stated before, I do not control the calendar of Mr. Gruber, but I had several interactions with Andy Gruber, and he was sitting right next to me across the hall. I have been in very many interactions and, if the R400 program was not his

sole responsibility, he was certainly close to be.

BY MR. NESE:

Q. Let's take a break.

BY THE VIDEOGRAPHER: Going off the record at 10:01 a.m.

BY THE VIDEOGRAPHER: Going back on the record at 10:11 a.m.

BY MR. NESE:

Q. Okay. Welcome back.

A. Thank you.

Q. We have been talking about the R400 project. What is the R400 project?

A. The R400 project is a GPU project that ATI was working on at the time, trying to create the next GPU for the company graphics processor unit.

Q. Was there an R300 project?

MR. TUMINARO: Objection. Outside the scope. You can answer the question.

THE WITNESS: Yes, there was an R300 project that was being developed by the West

Coast team.

BY MR. NESE:

Q. Do you know anything about the R300 project?

A. I know some very little details. The one thing I do know is that we re-used the base architecture for the sequencer as a starting point, so I know a little bit about that part of the R300 design but, outside of that, not a lot more.

Q. How is the R300 sequencer design like the R400 sequencer design?

MR. TUMINARO: Objection, form.

THE WITNESS: The R300 design is using clauses like the ones presented in figure -- on page six of the 871 Declaration.

BY MR. NESE:

Q. So I am sorry. You are referring to Exhibit 2?

A. Exhibit 2, yes.

Q. Okay.

A. The pixel shader of the R300 was using the same trickle-down flow where pixels would flow down a set of clauses and FIFO's in between. And so we re-used that concept when we established the R400 architecture of the sequencer as a base line.

Q. How did you modify that base line for the sequencer for the R400 project?

A. So the changes were fairly extensive, because we had to modify the sequencer which was used in a non-unified shader architecture to be operable in a unified shader architecture. So we had to have several layers of arbitration to be able to send both vertex data and pixel data to that sequencer. We also extended the design to support more clauses than the R300 did, and I believe we also extended the control flow, which is the program that runs the shader to be more flexible and have more options than the R300 would have had.

Q. How many clauses did the R300 support?

A. I don't recall the exact number, but I believe it to be around three or four.

Q. Do you know whether the R300 was ever commercialized?

A. Yes, the R300 was commercialized.

Q. Do you know the name of the chip?

MR. TUMINARO: Objection. Relevance.

THE WITNESS: I have terrible, terrible knowledge of the commercial names of our chips even today, so I do not know the commercial name of that chip. I just know it was commercialized.

BY MR. NESE:

Q. That's okay. You mentioned a moment ago a unified shader. What is a unified shader?

A. A unified shader is a novel concept that was introduced in the R400 chip, and the idea behind the unified shader is to reuse the most expensive resources of the chip, which are the shader core shader pipeline as well as the texture infrastructure -- sequencer register

files. And because there is so much cost into the shader portion of the chip, create only one version of that shader core and be able to reuse it for both vertices and pixels instead of having two different implementations of the vertex shader and the pixel shader like were done in previous chips.

Q. So what does "shader" mean?

A. Shader typically refers to the program that we run to either compute position, attributes for vertices or colour for vertices or, in the case of pixel, is a case we run on every pixel to generate the colour of the pixels once they are on the screen.

Q. Do you know what a partial unified shader is?

A. I would have no idea what -- I don't have any idea what this would mean, no.

MR. NESE: I would like to mark as Exhibit 4 Exhibit ATI 2040 in IPR 2015 325.

Exhibit 4 was marked for identification.

BY MR. NESE:

Q. Mr. Lefebvre, have you seen Exhibit 4 before?

A. Yes, this is the R400 architecture proposal from Steve Morein.

Q. And Steve Morein wrote this document; correct?

A. It is my understanding that Steve Morein wrote this document, yes.

Q. Would you turn to page nine of Exhibit 4, please, and let me know when you are there.

A. I am on page nine.

Q. There is a line on page nine that reads:

"The unified shader is based on the R300 partially unified shader."

Do you see that?

A. Yes, I see the line.

Q. Okay. Reading that line and the surrounding text, do you have any understanding

of what a partially unified shader refers to?

A. Well, not having any real knowledge of the R300 design, I do not understand what the author is meaning by the R300 partially unified shader. As far as I knew, the design of the unified shader is based on the pixel shader portion of the R300, which may or may not have been used for other things. I wouldn't know.

Q. You can put Exhibit 4 aside, sir. I would like you to take out Exhibit 1, which is your Declaration in the 325 proceeding. And please turn to paragraph four, which is on page one, I believe.

A. Yes.

Q. The first line of paragraph four you say: "The R400 includes many different functional blocks." Did I read that right?

A. Yes.

Q. What do you mean by "functional blocks"?

A. I mean blocks that perform different functions, such as the sequencer which is

operative to run the shader of the pixels and the vertices; the shader pipe itself, which is the block that does the math. The primitive assembly, which is the block that assembles the primitives, et cetera.

Q. So does each functional block have a separate code file in RTL?

A. Each functional block has a complete hierarchy of RTL code files associated with it, yes.

Q. Do any of the functional blocks share code files?

A. Some RTL files are used across different functional blocks. One example of that, for example, is an FIFO. We have a generic RTL module file that implements a FIFO design. So typically, when the block needs an FIFO, instead of re-implementing the FIFO, they use the generic version of the FIFO that ATI has to implement the FIFO, so they initiate a common FIFO design.

Q. How many functional blocks were in

the R400 project?

A. Quite a bit, and you can have a small view of those blocks on the picture on page two of the Exhibit 1 where, you know, the CP, PA/SC, RBBM, MH, memory controller, SP/SQ. Those are all functional blocks.

Q. Were there more functional blocks in the R400 project other than what's shown on page two of Exhibit 1?

A. I think those were the main ones. There could have been more minor ones that I don't recall, but those were the main blocks of the R400.

Q. Which of those blocks, if any, did you work on?

A. So I started working in 2000 on the SC, scan converter, before it got transferred to Orlando, and I also worked on the sequencer and shader pipe blocks and also, towards the end of the project, worked on the SX block. This is the shader export block.

Q. So which of these blocks are

involved in the unified shader?

MR. TUMINARO: Objection, form.

THE WITNESS: The blocks inside the unified shader are the SP/SQ, SX, TC/TP blocks, but all of the blocks that you are seeing on this slide interact with the unified shader in some way, shape or form.

BY MR. NESE:

Q. What do you mean by the blocks inside the unified shader are the SP/SQ, SX, TC/TP blocks?

A. To the best of my recollection, those are the blocks that are re-used for both vertices and pixels, so those are the blocks that I personally consider as part of the unified shader complex. Other people may have different opinions, depending on the interactions of the blocks. Like I said, the R400 is really a whole, and any blocks that really connect to the unified shader need to be aware of it, because they need to modify their interactions based on the fact that it's a unified shader and not two shaders

back to back. So my definition of the unified shader is the blocks that process both pixels and vertices.

Q. Are any of these other blocks on page two of Exhibit 1 involved in processing both pixels and vertices?

A. The -- it depends on your definition.

Q. Well, let's take your definition.

A. Right.

MR. TUMINARO: Objection, form.

THE WITNESS: The XP, for example, is the control processor and that's -- it's the first block on the left. It's not aware of pixels or vertices. It processes packets and sends them to the chip. Similarly, the memory hub or memory controllers, they only process memory transactions, and they don't make the difference between pixels and vertices. To them memory transactions are memory transactions. Of the blocks that actually do 3D rendering, the PA is processing the primitives, so that would be

the vertices. The SC is converting those primitives into pixels. And then you have the blocks of the shader complex that can run shader on either type of data: Pixels or vertices.

BY MR. NESE:

Q. So sticking with your definition of unified shader, is the memory controller block part of the unified shader?

A. Like I said before, because the memory controller is processing memory transactions, which is just an address and data, and does not care if it's coming from a vertex or pixel, I would not consider the memory controller piece to be part of the unified shader complex. However, it is needed and is connected to the unified shader complex in order for it to do memory operations.

Q. And what about the control processor? I understand that that may interact with the unified shader, taking again your definition, but is the control processor part of the unified shader?

A. So again, the control processor is part of the R400 chip and is definitely needed for the unified shader to see any data, because it is the entry point of the chip that's the only block that really talks to the driver. But I would not consider it to be part of the unified shader complex.

Q. Let's talk about the sequencer block. What are the functions of the -- excuse me. What are the functions performed by the sequencer block?

A. So the sequencer can be seen as the heart of the chip. It is the block that will arbitrate between pixels and vertices to decide which ones get to go next into the unified shader complex. It's also storing all of the state information between all of the vertices and pixels that are currently executing their shader program, and is responsible for executing or actually sending the instructions to either the SP or the texture to run either the vertex shader programs or the pixel shader programs on the

collective pixels or vertices in the unified shader complex. It allocates and schedules also rights to temporary storage that are written to by the vertex shader to store intermediate data between vertices and pixels, and also writes to the output buffers of the chip so that pixels can be sent to the frame buffer.

Q. Are any of the functional blocks on page two of Exhibit 1 arbitrating between pixels and vertices?

A. As best I can remember, the sequencer is the only block that really arbitrates between pixels and vertices. Like I said, I mean, the memory controller, not knowing what it's being sent to, will perform arbitration between memory transactions, but I would not call this arbitrating between pixels and vertices, but just ordering memory transactions of some sort.

Q. I think you said another function of the sequencer block is to send instructions to the shader pipe for executing programs. Did I get that right?

A. So the sequencer has a memory inside it called the instruction store, which collects all of the instructions that are needed to run a particular shader, whether it is a pixel shader or a vertex shader, and one of its roles is when that instruction is an ALU instruction, is to send that one instruction, one by one, to the shader pipes along with the control relative to that information meaning, you know, the GPR addresses that conform to that instruction so the shader pipe can execute the ALU instruction.

Q. Okay. Do any of the other functional blocks on page two of Exhibit 1 perform that function that you just described?

A. To my knowledge, the sequencer is the only block that has an instruction store, and reads that and sends that information to the shader pipe.

Q. I would like to talk about the shader pipe block. What is the function -- what are the functions being performed by the shader pipe?

A. So the shader pipe is the block that is responsible to store all the data that any given pixels or vertex would need. Like that's where the GPR's are, the general purpose registers where the data is stored to operate on. So it is a repository of that data, and one of its main roles is to perform ALU operations on that data. And ALU operations could mean: Add, subtract or any ALU operations that, you know, are specified in the R400 shader pipe specification.

Q. You said one of the main roles of the shader pipe is to perform ALU operations; correct?

A. Yes. One of the main roles of the shader pipe is to perform the ALU operations. Its other role, because it has the register files is to, when instructed by the sequencer, submit the addresses needed for a texture pipe operation or a vertex operation to said texture pipe, and collect that data when it comes back from memory.

Q. So the shader pipe is performing

both ALU operations and texture operations?

A. No.

MR. TUMINARO: Objection. Form.

THE WITNESS: The shader pipe only performs ALU operations, but because the register files are inside the shader pipe, the sequencer can instruct the shader pipe to perform an ALU operation or to read its GPR in order to perform a texture operation.

BY MR. NESE:

Q. Are any of the other functional blocks on page two of Exhibit 1, apart from the sequencer and the shader pipe block, responsible for performing ALU operations?

A. The ALU, the shader pipe is the only block, to my knowledge, that can do programmable ALU operations. There are other fixed function blocks, for example, the PA, which is the fixed function logic block that has fixed function logic to perform some ALU operations like computing a reciprocal in order to be able to generate the -- based on output from the vertex

shader, compute the position of the vertices on the screen. But this is not a programmable piece of ALU. It's just fixed function, not logic that happens to do ALU operations so, to my knowledge, the SQ and SP is the only complex that can do programmable ALU operations.

Q. And the PA only deals with primitives. Is that right?

A. So the PA receives vertices from the vertex shader and creates primitives based on that data, yes.

Q. I believe you said you also did some work on the shader export block. Is that right?

A. Yes. Towards the end of the program I also did some work on the shader export block.

Q. What are the functions of the shader export block?

A. The shade export block is the block that receives either intermediate data from the vertex shader, such as positions that are computed in the vertex shader or attributes, and stores them while pixels are generated, because

the pixels will source that data as part of their input. And it's also the block that is responsible to store data before it goes to the render backends, which is the RB's in this block diagram.

Q. I would like to talk about the work you were doing on the sequencer block. Did you write the emulator code for the sequencer block?

A. Yes. Part of my function as being an architect for the sequencer, I had the responsibility of both writing the specification and the C code for the sequencer. We were a team of people, so some other people might have done some contribution in terms of when they find bugs, or letting me know what various issues, but so I was responsible for writing most of the code. There could be some other people that added, you know, a few words here and there during the course of the program.

Q. How many other people were working on the emulator codes with you for the sequencer block?

A. I was the only one officially assigned to the emulator code for the sequencer. But part of the validation -- so the people that wrote the tests for the chip which -- in order to energize it and to test, sometimes, when they find an issue that is very easy to fix like a typo, they would go in and fix that block. But they are not really the owners or they wouldn't really contribute greatly to the code. I was really the sole person writing new features and making main updates to that source code base.

Q. So if there are any fixes that were needed beyond something like a typo, someone on your team might tell you about it and you would personally write the code to fix that. Is that right?

A. Yes. Typically what would happen was that I would be made aware of a test that would fail, and I would re-run it on my local machine, reproduce the failure, and then debug it and eventually fix the program in the code.

Q. Did you write any of the RTL code

for the sequencer block?

A. Like I said, at AMD we don't like to mix and match the people that write the emulator and the people that write the RTL. So in that respect I did not write any of the RTL code, but in the same way, as part of debugging, whenever we found issues in the chip and the issue was very simple to fix and I could identify it, I may have done some minor tweaks to the RTL to fix those issues when they were very, very simple or typo-related kind of issues.

Q. So can you program in RTL?

A. Yes, I can program in RTL. I can do fixes in RTL, and I can program in RTL. It's part of my engineering formation, yes.

Q. But you didn't write any of the RTL code for the R400 project?

A. That is correct. I did not write any of the RTL code for the R400 project.

Q. Let's talk about the shader pipe block. What -- were you also writing the simulation code for the shader pipe block? I am

sorry. I misspoke. The emulation code.

A. I -- if I recall correctly -- again, this is a long time ago. I started from a code base that was provided to me by Andy Skende, and I helped him whenever there were issues or bugs to fix those issues in the SP code. So in the emulation -- because the shader pipe does not have any control, it's just a block that is solely controlled by the SQ -- the shader pipe was represented as a library of functions that could be performed of ALU functions, and that code resided into the sequencer repository, if you will and, in that manner, whenever issues were found with the precision of said ALU operations, I would fix them directly and Andy Skende would be the person really more working on the RTL side of things.

Q. How about for the shader export block? Did you write the emulation code for the shader export block?

A. I remember writing some code, but I do not remember if it was in the time period that

we are talking about here.

Q. Do you know who else was writing emulation code for the export block?

A. If I remember correctly, Andy Skende also contributed to the shader export block emulation code.

Q. Other than Mr. Skende, was there anyone else?

A. To the best of my memory, no, but we could check the perforce logs to make sure.

Q. How many engineers were working on the entire R400 project?

A. There was the whole Boston office or Marlboro office, and also quite a few people from Orlando. I would say at least 100 persons. I cannot tell you. There is probably some blocks and, you know, that I know exist, but I don't know how many people were working on these blocks because they were in Orlando and I was not located next to them. So I know that, you know, most of, if not all of the people in the Boston Marlboro office were working on the R400 program.

Q. Were you responsible for managing any of these engineers that were working on the R400 project?

A. No, I did not manage anyone. I was a technical person with nobody under my responsibility.

Q. You didn't supervise anybody?

A. I did not directly supervise anybody, but during the course of my work I interacted a lot with other people on a technical side. So I did not -- I was not responsible for any performance evaluation or, you know, management types of interactions with anyone, but I had many technical interactions with very many people in both Orlando and in Boston.

Q. How many of these engineers did you personally work with on the R400 project?

A. Do you mean on site or do you mean -- because I had like several meetings with people in Orlando which I didn't see quite as much as the people that I saw in Boston. So can you be more precise into the question?

Q. Sure. You mentioned earlier that you did have -- you interacted with a lot of people on the technical side. I am just looking for a rough estimate of how many people on the R400 project you did interact with.

A. So I would say I interacted with at least ten or so people from the Orlando office. And on the Marlboro side interactions were probably maybe 40 or so people.

Q. So roughly half of the engineers that were working on the whole R400 project?

A. Yes. I mean, those are the people that I worked more, a lot with. I mean, there were other interactions with, you know, more -- whenever a problem arose in another block and it affected the sequencer or the shader pipe, I would interact with these people. That was more of a point-based interaction. So this is like the core group of people I would interact with very often.

Q. Do you know how many engineers were writing the RTL code for the sequencer block?

A. For the sequencer block we started with one engineer, if I recall, and then expanded to two locally in Boston, and after that it grew to three or four toward the end of the program.

Q. And when you say toward the end of the program, what time frame are you referring to?

A. Say around 2003, 2004 when we were trying to finish up and close all the miscellaneous corner cases that we needed to fix before we could have a productizable design.

Q. Was the R400 project ever taped out?

A. The R400 project got repurposed as Xenos, and that project got taped out.

Q. What do you mean by "repurposed"?

A. During the course of the R400 development, my understanding is that we were approached by an external customer that needed a semi-custom chip, and so because we had a design that was already supporting the DX10 version of the API that that customer wanted, the company decided to repurpose the program and create the

Xenos chip instead.

Q. What changes were made to the R400 project when it was repurposed to the Xenos chip?

MR. TUMINARO: I am going to object to the extent is calls for third-party information, and caution the Witness not to disclose any third-party confidential information in answering this question.

THE WITNESS: I can't really answer that question without divulging confidential information pertaining to Microsoft.

BY MR. NESE:

Q. Okay. What if I -- let's limit the discussion to the sequencer block. Did the sequencer block change from the R400 project to the Xenos project?

MR. TUMINARO: Same caution with respect to confidential information.

BY MR. NESE:

Q. I am just looking for a yes or no at this point.

A. I can't give you that information.

Again, it will divulge confidential information between Microsoft and AMD.

Q. To say whether the sequencer block changed at all from R400 to Xenos chip is third-party confidential information?

MR. TUMINARO: You can answer yes or no, if there is.

THE WITNESS: Yes, there would be confidential information in that; yes, because you would divulge configuration information.

BY MR. NESE:

Q. I don't want to know the specifics. I just want to know yes or no, did the sequencer block change from the R400 to the Xenos chip?

A. I would say in configuration only.

Q. Again without getting into the specifics; what do you mean by "in configuration only"?

MR. TUMINARO: Again I will caution you. If you can answer that without disclosing third-party confidential information, you can answer.

BY MR. NESE:

Q. It might be I can help a little bit. What does "in configuration only" mean?

A. It means, as with any design, you can scale the design to process more or less elements to increase or decrease its performance. And so that's what I mean by "in configuration only."

Q. I see. So for example, moving from the ability to process 16 clauses to an infinite number of clauses, is that what you mean by "configuration only" or would that be an example?

A. Not really. That's more a design change. What I would mean by configuration change is processing 16 pixels per clause versus 32 or 100.

Q. I see. But the R400 project itself was never taped out. I understand that the Xenos chip was, but was the R400?

A. To my knowledge, the R400 project itself never got taped out; correct.

Q. Was the R400 project ever

synthesized?

A. Yes. The R400 project was synthesized many times, if I recall correctly.

Q. Do you recall when the first time the R400 project was synthesized?

A. An Exhibit, part of the Declarations that I wrote specifies that date to be around August of 2002, if memory serves. That's the first time that the program was fully synthesized. I mean all the blocks.

Q. Do you recall what Exhibit number that is in your Declaration?

A. It's either -- it's probably 2066 of Exhibit 1.

Q. Did the R400 project ever get to the plan and route phase that we discussed earlier?

A. I don't recall that information. It might. It might not. I don't recall.

Q. Mr. Lefebvre, I am handing you now what I have marked as Exhibit 5.

Exhibit 5 was marked for identification.

BY MR. NESE:

Q. This is Exhibit 2010 to the 325 IPR proceeding. Have you seen Exhibit 5 before?

A. Yes, this is the R400 sequencer specification version 0.4.

Q. And you wrote this document; correct?

A. Yes, I wrote this document.

Q. Would you turn to page three of Exhibit 7, please?

A. I am on page three.

Q. Here under revision four you have a note that reads:

"Added the dynamic allocation method for register file and an example (written in part by Vic) of the flow of pixels/vertices in the sequencer."

Do you see that?

A. Yes, I see that comment.

Q. Who is Vic?

A. Vic is Vick Romaker, which was the RTL lead for the sequencer.

Q. Would you spell his last name?

A. R-O-M-A-K-E-R.

Q. Did Vic write any portions of Exhibit 5 here?

A. I don't recall him writing anything here, but the comment clearly says that I must have been using an example that was coming from him.

Q. Where is that example in Exhibit 5?

A. That would be on page 14.

Q. Okay. Where on page 14?

A. The four pictures, as well as the paragraph write underneath it.

Q. You are referring to the paragraph that begins with:

"Above is an example of how the algorithm works."

Is that right?

A. That is correct.

Q. So Mr. Romaker wrote this paragraph

here?

A. No. What I said is that I probably used his example to write this paragraph.

Q. Let's go back to page five of Exhibit 5, please. And on page five of Exhibit 5 there is an image underneath the text, "Top Level Block Diagram". Do you see that?

A. Yes, I see that image.

Q. Is this referred to as a reservation station?

A. This image is trying to replace the flow of information going through the sequencer as well as the various arbiters, and it also is representing the eight texture reservation stations and eight ALU reservation stations of either the pixels or the vertices.

Q. You mentioned the figure shows various arbiters. How many arbiters are in this figure on page five of Exhibit 5?

A. In this figure you have three main arbiters: The input arbiter which is at the top, which is operative to process or choose between

either pixels or vertices as to which type gets to go next. Then on the right you have the texture arbiter which is looking at all the elements in the reservation stations and gets to pick which one would go next to the texture position. And on the right is a mislabelled arbiter that is labelled texture arbiter in this figure, which should have been called an ALU arbiter that is looking at the eight ALU clauses and seeing which of these clauses contain information so that -- to pick the next program to be sent to the shader pipe.

Q. Just so we are clear, Mr. Lefebvre, the mislabelled arbiter which should be the ALU arbiter, is that the arbiter on the left?

A. Yes, the mislabelled arbiter is the arbiter on the left of the picture.

BY THE VIDEOGRAPHER: Going off the record at 11:07 a.m.

BY THE VIDEOGRAPHER: This begins tape number two in the deposition of Laurent Lefebvre. We are back on the record at

11:20 a.m.

BY MR. NESE:

Q. Welcome back, Mr. Lefebvre.

A. Thank you.

Q. When we last left off we were discussing the arbiters that are on page five of Exhibit 5. I would like to return to that discussion. You mentioned an input arbiter and are you referring to the vertex/pixel vector arbitrator?

A. Yes, this is what I was referring to.

Q. What is the function of the vertex/pixel vector arbitrator in this figure on page five of Exhibit 5?

A. So because we have a unified shader, we need to send pixels or vertices to said unified shader. And in the R400 design there is a single main input bus that can be used to send that information down, and so because we have two types of customers trying to address or go to the shader complex, we needed an input arbiter to

decide in cases where both pixels and vertex would -- wanted to come in at the same time, we needed an arbiter to pick which one would go next.

Q. Is there a difference between the terms "arbiter" and "arbitrator"?

A. These are just caveats of having to deal with somebody that does not speak English really well, and arbitrator was kind of my terminology for arbiter when -- because in French it has a different meaning. That's really all it is. So this should be seen as the exact same thing.

Q. Okay. I may slip and say one or the other today, but can we have an understanding that arbiter and arbitrator are really the same term?

A. Yes, we can have that understanding. Absolutely.

Q. Thank you. So the text below this figure on page five of Exhibit 5 reads:

"There are two sets of the

above figure, one for vertices and one for pixels."

Do you see that?

A. Yes, I see that text.

Q. So does this mean that in the actual sequencer you are describing here there will be six arbiters: Three for the one representation of this figure and three in the other one?

A. The comment is slightly misplaced. We have two sets of reservation stations, so everything below the possible delay for available GPR's is replicated for vertices and pixels, and the main vertex/pixel arbiter there is only a single instance of that one. And so you could think of the two sets are sitting on top of one another and when we refer to the texture arbiter, it really means that it sees all of the texture clauses from either the pixel and vertices and picks the one -- the winning one from both sets, basically.

Q. I see. So there will be only one vertex/pixel arbiter?

A. Yes, that is correct.

Q. And will there be only one -- I am not sure what this box here you have labelled is. It says: "Possible delay for available GPR's". Is there only one instance of that or are there two instances of that?

A. As best as I can tell, and remember, this box is really not a block or an instance. It really represents push back on the main vertex/pixel arbiter to tell it that there is no room in the unified shader complex at that time, so it cannot push one more element inside the complex.

Q. What do you mean by "push back on the main vertex/pixel arbiter"?

A. I mean for every vertex or pixel that you are trying to write to the unified shader, you need to have room in the GPR's, which are the general purpose register files that sits in the memory. And when there is -- there is like an allocator of those GPR's in the sequencer that keeps track of how many GPR's are in use

versus the total number of GPR's that a chip can store. And whenever we try to insert a new work group, or a new vertex wave or a new pixel wave into the unified shader, we need to first make sure that there is room into the GPR's to receive that new work group. So there is an interface built into the arbiter to let it know whenever it can proceed into loading the next group of vertices or the next group of pixels based on the availability of the GPR's.

Q. Do you see just above the text: "There are two sets of the above figure", there is an arrow in the figure pointing downward. Do you see that?

A. Yes, I see that arrow.

Q. Where is that arrow leading?

A. In the case of the above picture, being vertices, that arrow leads to the primary cache, which is the intermediate buffer in the shader export block that stores the vertex parameters. In the case of pixels, that arrow through the export buffer leads to the render

backends. So one could say that this arrow is only connected to the export buffer and then after that they take different routes.

Q. I see. But there is no arbiter down there where that arrow is pointing to, is there?

A. There is no arbiter for that arrow, because the arbitration is already done in the sequencer. As you can see, the arrow is only connected to the ALU side, so you absolutely have to finish your shader with ALU instructions and by definition, because there is only one ALU shader pipe. When the sequencer picks the winning ALU clause it's either a pixel or vertex. It cannot be both at the same time. So the arbitration is done at that level.

Q. And by "that level", you mean what's actually shown in this figure here on page five?

A. Again, the figure is slightly misleading, because there are two sets of this picture.

Q. Of course.

A. But yes, that would be from the

left-hand side, mislabelled "texture arbiter", which in fact is the ALU arbiter which picks between all of the vertices and all of the pixels that are present in the unified shader at that time.

Q. So these commands that you are showing here in this figure on page five of Exhibit 5, they are processed further down the pipeline on the shader pipe?

A. The shader pipe is connected to the ALU arbiter. So whoever -- whichever clause the ALU arbiter picks is going to run its instructions on the shader pipe next.

Q. How is the shader pipe connected to the ALU arbiter?

A. The ALU arbiter picks the next clause of ALU, and a clause is a collection of instructions. That clause is sent to a sub-module of the sequencer which is called the AIS -- if I remember correctly, or -- A -- ALU -- A -- A something. ACF or -- the block operative to read all the instructions in that group of

instructions one by one, and sending it to the shader pipe. So all of that state machine that reads all the instructions from the instruction store in the SQ, the output of that state machine which is going to be the instruction to be used as well as the addresses and possible constants, that's what's sent to the shader pipe, and so that's the connection to the shader pipe from the sequencer.

Q. Is there a similar connection for the texture arbiter on the right-hand side of this figure?

A. Yes, there is a very similar mechanism on the texture side where the texture arbiter picks the winning texture clause which is a group of texture instructions, sends that to a state machine which then reads all the instructions one by one and sends them through the texture pipe machine through the shader pipe, because it needs to read the register file on the way out, because that's where the addresses are stored in the register file.

Q. Please turn to page 11 of Exhibit 5.

A. I am on page 11.

Q. Where on this figure, if it is shown, are pixel commands processed?

A. Well, pixel over text commands are coming from the instruction store/cache, and if you look at the arrow that's on the left side of that box, to me that would represent the reading of those instructions and sending them to the ALU pipeline so that they can be processed.

Q. So the ALU's shown in this figure on page 11 of Exhibit 5, they can process both pixel commands and vertex commands?

A. That is correct. The gray section is what we refer to as the shader pipe, and in the unified shader that shader pipe can process either vertex commands or pixel commands.

Q. And how do you know that by looking at this figure?

MR. TUMINARO: Objection, form.

THE WITNESS: I don't know that there is any way to know that by just looking at

this figure. I mean, the figure is just showing connections from an instruction store to the ALU's, and so if the instructions are vertex instructions then the ALU or shader pipeline will process vertex instructions. If the instructions in the instruction store are pixel instructions then the shader pipe will process pixel instructions. There is no way to tell what you are processing without being able to look what's inside the instruction store.

BY MR. NESE:

Q. Let's put Exhibit 5 to the side for now, and I would like you to turn to Exhibit 1 again, which is your Declaration in the 325 IPR proceeding. And I would like you to please turn to paragraph 30, the portion that's on the bottom of page 18.

A. I have paragraph 31 on page 18.

Q. I am sorry. I am looking at the top of page 18, the very last sentence above the figure. And that sentence says:

"To meet Microsoft

specifications, I changed the sequencer's control flow."

Do you see that?

A. Yes, I see that sentence.

Q. And when you are referring to the change in the control flow, do you mean that you were changing the control flow that you were describing in version point 4 of the specification which is Exhibit 5?

A. Yes. So the main difference that -- the main change we had to do to meet Microsoft's requirements for the API was to enable an unlimited number of ALU and texture clauses. In this 0.4 design, as you can see, there is only a limited number of texture clauses you can have. Each shader can only have eight texture clauses and eight ALU clauses, because there is no way for you to recirculate the data and have more ALU clauses if need be. So because Microsoft, as part of their new API, the X10 needed a limited number of texture and ALU clauses, we could not use this design because of the limit of clauses

it had. And so we -- this design would not have been certified as a DX10 chip which obviously, if you are building GPU's you want to be able to say you support the newest and latest API. So to fix this we changed the sequencer design and made it instead of using a FIFO-like approach, used global reservation stations shown in the -- shown as Exhibit 20, 28 on page ten, one that would contain all the vertex waves and one that would take all the pixel waves. And in this design the data is always recirculated to the same memory, and we keep track of where in the program each waves are using status bits. So with this design you can have as much clauses as you want.

Q. I will hand you now what I have marked as Exhibit 6.

Exhibit 6 was marked for identification.

BY MR. NESE:

Q. And please let me know if you have seen this document before.

A. This document is version 2.0 of the

R400 sequencer.

Q. Does this reflect the changes to the control scheme that you made to meet Microsoft's specifications?

A. Yes, this specification reflects the changes that we made to support the new DX10 requirement from Microsoft.

Q. By the way, does DX10 refer to Direct X10?

A. Yes, it does.

Q. What did you need to do to make these changes from the scheme described in version 0.4 to the control scheme described in version 2.0?

MR. TUMINARO: Objection, form.

THE WITNESS: So the main change that we had to do was to extend the number of clauses that were available to pixels and vertices from eight to an unlimited number. So to that regard we created a more flexible storage for the reservation station such that we could recirculate the threads or the waves back to the

storage instead of having to move them from one FIFO to the next.

BY MR. NESE:

Q. And looking at the figure on page ten of Exhibit 6 -- and this is the new scheme here; correct?

MR. TUMINARO: Objection, form.

THE WITNESS: Correct. On page ten of that spec is a top level diagram of the new scheme.

BY MR. NESE:

Q. Okay. There is a box labelled VTX RS. Is that the vertex reservation station?

A. Yes, the VTX RS is the vertex reservation station.

Q. And what does that mean: "Vertex reservation station"?

A. Vertex reservation station is a term we used to describe what the sequencer uses to keep track of the state of progress of all the waves in the system. You can think of it as the collection of all of the eight vertex reservation

stations that you had in the previous design put into a single structure, and being able to read them back to and from the same location instead of having to move that state around from reservation station to reservation station. In this spec. there is more details -- if I can find it -- as to exactly what status we keep in those reservation stations.

Q. In the interests of moving things along, I don't need that level of detail. If you feel you have given me a complete answer.

A. I am fine with that.

Q. I don't want to interrupt you and I don't want to cut you off, but just in the interests of moving things along. Looking at page ten again of Exhibit 6, there is a box labelled PIX RS and that's the pixel reservation station; correct?

A. Correct. That box represents the pixel reservation station. And again in the similar way, you can think of it as the collection of the reservation stations of the

previous design with the added capability of having -- of being able to recirculate data back to the same reservation station instead of having to move it along.

Q. I would like to take a look again at page 18 of your Declaration which is Exhibit 1, the same page we were on earlier. And I will be referring to Exhibit again, so keep that handy if you have the room. I know space is tight over there. So on page 18, paragraph 31 of Exhibit 1, you have written:

"In this version there are two reservation stations, one reservation station for vertices, VTX RS and one reservation station for pixels, PIX RS."

Do you see that?

A. Yes, I see that text.

Q. How many -- if this version -- and by this version here in paragraph 31, you are referring to version 2.0 of the specifications?

A. Yes, that's what I mean by it.

Q. If version 2.0 had two reservation stations, how many reservation stations were in version 0.4 which is Exhibit 5?

A. So in Exhibit 5 you had per pixel and per vertex 16 individual reservation stations, eight of them for texture and eight of them for ALU. In the new design we collapsed all of that storage into a single location which we call the pixel reservation station or the vertex reservation station, and you can store the same amount of information that was there except that instead of having 16 different memories or blocks you have just one block with many entries and then a bit saying whether or not this particular vertex or pixel thread needs to be an operation on the texture side or on the ALU side next. So the state is collected in a single location versus 16 different locations in the previous implementation.

Q. So let's keep looking at the figure on page 18 of Exhibit 1, which is the same figure

as page ten of Exhibit 6. There is an input arbiter in the figure on page 18 of Exhibit 1; correct?

A. Correct, there is an input arbiter on that figure.

Q. And is that the same input arbiter as the input arbiter you described for version 0.4 of the specification?

A. It's an arbiter that performs the exact same functions; yes. It probably had to be modified slightly to be able to talk to the new reservation station structure. I don't remember the details, but the functions that it performs are the same. It's to pick the next work to be submitted to the shader core, either vertices or pixels.

Q. And version 2.0, again referring to page 18 of Exhibit 1, has an execution arbiter. Is that right?

A. Yes. This figure has an execution arbiter, and that's very similar to the arbiters that were present in version 0.4. It's looking

at all the vertices and pixels that are ready to process and, based on the status bits of each, is going to pick one to the shader pipe or the ALU and one to go to the texture, which is exactly the same function that the previous arbiters were doing.

Q. But there were two arbiters performing those functions in version 0.4 of the specification? One was the ALU arbitrator labelled -- or what should have been labelled -- and the other was the texture arbitrator. Is that right?

A. That is correct. And you know, while there is only one box here saying exec arbiter, saying because there are two resources, there are two decisions being made on every clock, one for the ALU side and one for the texture side. I think here, for the sake of simplicity and illustration, it is shown as a single box with two arrows coming in and two arrows coming out, but in fact this box is performing two types of arbitration, one for ALU

and one for textures, which is the same thing as the previous implementation, just shown differently.

Q. So this box in the figure on page 18 of Exhibit 1 that is labelled "execution arbiter", there are actually two arbiters inside that box?

A. Yes, two functional arbiters. I don't recall exactly how it got implemented in RTL and if it turned out to be two different physical blocks, but there is two arbitration functions that are performed by this exec arbiter. One of them is to pick an ALU, vertex or pixel wave winner, and the other side is to pick a texture vertex or pixel thread winner.

Q. And again on the figure on page 18 of Exhibit 1 I see there is an ALU box and also a texture box, and are those the -- are those performing the processing of the commands that are coming through this execution arbiter?

MR. TUMINARO: Objection, form.

THE WITNESS: Yes. The ALU box is

effectively the shader pipe which is performing all the ALU operations as instructed by the sequencer, and the texture box is the texture pipeline which is performing all the texturing operations, again as instructed by the sequencer in a very similar form that we had in version 0.4.

BY MR. NESE:

Q. And the ALU is processing both pixel commands and vertex commands?

A. The ALU is processing either vertex commands or pixel commands, as instructed by the exec arbiter, not both at the same time. In the exact same way as version 0.4, you got to pick only one winner between pixels and vertices. You cannot pick both at the same time, because it's a unified shader. There is only one shader processor, so you can only pick one of vertices or pixels to be sent to that. In a very similar manner there is only one texture pipeline, so you need an arbiter to pick either a vertex wave or a pixel wave to send to the texture unit.

Q. But in the case of the ALU, it has the capability to perform both pixel commands and vertex commands; correct?

A. Both the ALU or shader pipe and texture have the ability to perform either vertex commands or pixel commands. They can, you know -- these blocks don't make the difference. They are just doing what they are told to do by the sequencer, and so they just receive an instruction and a stream of data, and they perform that instruction and -- on that data regardless if it's a pixel or vertex. The sequencer is the one that knows whether it's a vertex or pixel and that makes all the choices as to what instruction and from what group gets to go next in either the ALU or the texture side.

Q. And do all ALU's function that way? In other words, they just perform the instruction that comes in on the stream of data, as you said, this ALU does here?

MR. TUMINARO: Objection, form.

THE WITNESS: I would like to know

what you describe as all ALU's. Are you referring to CPU's? GPU's? What do you mean by ALU in the context of your question?

BY MR. NESE:

Q. I mean ALU's as the way you are describing it in these specifications that we have been talking about.

A. An ALU is just an engine able to perform mathematical operations, just like adds, or subtracts or multiplies. And so any ALU pipeline can perform all of these arithmetical operations on whatever data you give it. It's just a simple pipeline. It does not need to know about anything. It just needs to know the operation you wanted to be done, and the data that it needs to operate on and that's it. So in that sense, I guess, any ALU's would be able to do these types of operations if properly instructed, yes.

Q. Let's look at Exhibit 6 again, and please turn to page 22.

A. I am on page 22 of Exhibit 6.

Q. Okay. Under the heading 5.6: "Constant Waterfalling", there is a sentence three lines into this paragraph that reads:

"There is a small synchronization issue related with this."

Do you see that?

A. Yes, I see that issue.

Q. What is the synchronization issue that you are referring to here?

A. Okay. In the sequencer of that time there is a constant buffer memory which is not shown here, but that stores all of the constants that the shader can operate on. And by "constant" I mean any variable that is shared across the collectiveness of the pixels of a program, and it would be the same for all the collectiveness of pixels. So because it's a separate memory and that the shader is going to use those constants, the sequencer needs to make sure, before it can allow the execution of either the vertices or the pixels that will use that

constant, that the data actually was written to the memory that is going to be used by the shader later before it can start the execution of that shader.

Q. Would the sequencer function properly with this synchronization issue that you just described?

MR. TUMINARO: Objection. Form.

THE WITNESS: If you -- you mean if you run into a synchronization issue during the course of a program?

BY MR. NESE:

Q. Yes.

A. So if you ran into that synchronization program in the course of a program, what would happen is that you would read either an old or corrupted constant from that constant memory and use it in a later ALU operation and so, doing that would give you anything to corruptions -- from hangs to corruptions, depending on what you are trying to do with that constant in the pixel or vertex

shader program. Constants are just an input to the operation, and so if you don't make sure that the right values are sent to the ALU, you might get the wrong data out.

Q. Let's turn to the last page of Exhibit 6, and that's page 58. Here you are describing some open issues; correct?

A. Yes, the top of the page is actually the end of an execution example, and then section 25 is listing a very terse list of open issues.

Q. What is an open issue?

A. Typically at -- any specifications list what the issues that are currently being worked on or not completely solved at the time and that need more work. Unfortunately that's at the discretion of the author and, depending on the work we were doing or -- our busyness with other aspects of the R400, this section may or may not be populated correctly.

Q. How do you know whether the section was populated correctly?

MR. TUMINARO: Objection, form.

THE WITNESS: There is not really any way for you to be able to tell if this section was populated correctly. It's really depending to the author of that particular spec, whether or not that person took the time to list all of the issues present at the time.

BY MR. NESE:

Q. Who was the author of this particular spec? And by that I mean Exhibit 6.

A. This one is myself.

Q. Okay. So given that you were the author of this spec, is this open issue list on page 58 of Exhibit 6 populated correctly?

A. I don't recall. I am sorry. It's too far away.

Q. You state in section 25:

"Need to do some testing on the size of the register files as well as on the register file allocation method. (Dynamic versus static)."

Do you see that?

A. Yes, I see that sentence.

Q. What does that mean?

A. So the register file as stated previously can -- is really used to store all the data that either the pixels or vertices can use during their shader execution. The size of said register file and, depending on how much use per wave is read -- how much space per wave is being used in the register file, will determine how many groups of pixels or how many groups of vertices we can put in the system at any given time. It is important to have enough pixel and vertex waves in the system so that the sequencer can always pick something to execute while it's waiting, for example, for some data to return to memory, in which case those waves would be in the GPR's but dormant because they cannot operate their next instruction because they are waiting for data to come back to memory. That latency can be quite long. So if you have many pixels and many vertex waves in the system at any given time, we can do what we call hide the latency by

executing work items from other pixel or vertex waves in the system while we wait for the data return for any given number of vertex or pixel waves. So the size of the register file is really -- determines the performance of the chip and its ability to hide the memory latency. As far as the allocation method, we envision two ways to allocate the GPR's. One of them was a static allocation method where the schedule file is split into two sections, one section that can be used by pixels and one section that can be used by vertices, and that boundary is fixed and never moves. Another method that we investigated during the R400 project is what we call the dynamic allocation where that boundary or the number of GPR's that are on the chip that can be used by either pixels or vertices can move, and so it would give you -- it would allow you to have access to more GPR's for vertices or pixels, depending on the workload on the chip.

Q. So when you say you need to do some testing on the size of the register file on page

58 of Exhibit 6, what sort of testing are you referring to?

A. Again, it's been a long time so I don't really recall, but if you are asking me that question today, I would say that the testing that you need to do is really more on a performance aspect of it, to make sure that we have enough GPR's to hide the latencies of the shaders or the applications we care about. And we need these applications to work well because we are going to be benchmarked on these applications, so that's what I would think of if you ask me the question today. I don't remember what I was thinking of at the time.

Q. Do you know whether you ever did that testing?

A. I don't recall, honestly.

Q. You have got a note on this same page, that is page 58 of Exhibit 6: "Saving power". What do you mean by that?

A. So I mean, power is always a top concern whenever you build a GPU, and so

basically I think it's a remark of letting me remember that as we implement more and more features, is there anything we can do to make them more power-effective or use the energy in a more efficient way. I can't think of anything else.

MR. NESE: I think this is a good spot to break for lunch.

BY THE VIDEOGRAPHER: Going off the record at 12:09 p.m.

BY THE VIDEOGRAPHER: Going back on the record at 1:19 p.m.

BY MR. NESE:

Q. Welcome back, Mr. Lefebvre.

A. Yes.

Q. I want to talk about Mr. Steve Morein again. In the 2001 to 2003 time frame, how often did you work with Mr. Morein?

A. I would say probably on a daily to two days basis. Either I see him every day or every other day.

Q. And was he your supervisor?

A. Andy Gruber was my supervisor, not Steve Morein.

Q. What position did Mr. Morein hold in 2001 to 2003?

A. I don't really recall. We always referred to him as an architect, but I don't recall what the actual technical position he was holding at the time.

Q. Do you know whether Mr. Morein is still working for ATI?

A. I don't think so. I mean, I know he left some time ago. I am not aware of him ever coming back but, you know, I don't -- I am not in all the offices at any given time. I have seen some people come and go without me knowing, so it's possible that he came back working for us as a contractor or not. But I know he left the company some time after the Xenos project ended.

Q. Do you recall what year that was?

A. I don't recall the year, but I would say, if my memory serves, it would have been after 2005.

Q. Do you know where he is working today?

A. I have actually no idea.

Q. What about Mr. Andy Gruber? How often would you work with Mr. Gruber in the 2001 to 2003 time frame?

A. I would say about the same cadence as I was working with Steve Morein. Maybe a little bit more, because Andy Gruber was my manager and, as such, we had other discussions. But yes, about daily or over two days basis.

Q. Is Mr. Gruber still working for ATI?

A. My understanding is that Mr. Gruber is now working for Qualcomm.

Q. Do you know when Mr. Gruber left ATI?

A. That would have been when ATI sold part of their business, probably around the 2006, 2007 time frame. But my memory is a bit fuzzy on the exact dates he left after the Xenos program.

Q. You said that it's your understanding that Mr. Gruber is now working for

Qualcomm. How do you have that understanding?

MR. TUMINARO: I caution the Witness not to disclose the substance of any communication you may have had with Counsel in answering this question.

THE WITNESS: So as you know, AMD has an office in Boxborough, Massachusetts. And I go there fairly often, and it so happens that Qualcomm also has an office in the same building in Boxborough, Massachusetts, and so I run into Andy Gruber quite often when I am in Boxborough on my way in or out of the office.

BY MR. NESE:

Q. You still keep in touch with Andy Gruber?

A. I do not. I just run into him from time to time, but he is not an acquaintance or anybody that I keep in touch with.

Q. Do you know whether Mr. Gruber knows about these proceedings?

MR. TUMINARO: Objection, form.

THE WITNESS: I think divulging that

information would kind of give you Attorney privileged kind of internal view discussions. Maybe not in that case, but.

BY MR. NESE:

Q. Okay. How often did you work with Andy Skende in the 2001 to 2002 time frame?

A. At the beginning, like in early 2001 not so much but then, as we were going into more of the implementation phase and the debug phase which is around mid 2002, then I had like lots of interactions with Andy Skende, because we had lots of things to discuss and debug, with him being the owner of the RTL of the SP and myself being responsible, or at least co-responsible for the emulator of that block. So during that time period let's say multiple times a day, and at the beginning maybe once a week or so.

Q. Is Mr. Skende still working for ATI?

A. It is my understanding that Mr. Skende is now working for Invidia.

Q. How did you come to have that understanding?

A. I had several discussions with other people in the Boxborough office, and was also informed of his departure and they told me that's where Andy Skende went to.

Q. When did he depart ATI?

A. I don't recall exactly. Not too, too long ago. Maybe five years ago or so.

Q. I would like to turn back to Exhibit 1, which is your Declaration in the 325 IPR proceeding, and let's please turn to page three, paragraph seven.

A. I am on page three.

Q. Paragraph seven reads.

"No later than August 24, 2001 Steve Morein, Andy Gruber and I collectively conceived of the graphics processing system in the '053 patent."

Did I read that correctly?

A. Yes, you read that correctly.

Q. Now, I understand you didn't write this sentence. Is that right?

A. That is correct.

Q. What is your understanding of the word "conceived" in this sentence?

A. My understanding is it would mean invented the graphics processing system or, you know, specified enough so that it would be -- that the idea would be put on paper and we would be able to proceed with it.

Q. Proceed with it how?

A. Proceed in the sense of being able to initiate discussions with the design team, as well as starting to write the emulation portions of the design so we can start stimulating it and understanding more the caveats and corner cases of that particular design.

Q. Okay. Sir, I am handing you now what's been marked as Exhibit 7 and also what's been marked as Exhibit 8.

Exhibits 7 and 8 were marked for identification.

BY MR. NESE:

Q. Do you recognize Exhibit 7?

A. This looks like the patent on the multi-thread processing system that has been somehow redacted up or something.

Q. Yes, and I apologize. I just noticed this too. This is the same issue we had, Mr. Tuminaro, I think, at Dr. Wolfe's deposition. There is some shading, I guess. There is some sort of problem with the file at the bottom there.

MR. TUMINARO: Yes, the page that contains column one and two, there is some shading or text that -- shading that covers up some of the text on the bottom file.

BY MR. NESE:

Q. And my apologies for that. It was -- again, I understand this is an error with the file. But what I have also handed you Exhibit 8, which I will represent to you is the claims of the '053 patent that are at issue or the independent claims of the '053 patent that are at issue in the 325 proceeding?

MR. TUMINARO: I am going to object

to Exhibit 8 on relevance and outside the scope.

MR. NESE: You believe that the claims of the challenged patent are irrelevant and outside the scope?

MR. TUMINARO: Well, I haven't matched these up to the claims in the '053. I haven't seen this Exhibit before.

MR. NESE: Understood.

BY MR. NESE:

Q. So I will represent to you, and for Counsel as well, that these are a word-for-word copy of the claims one and five in the '053 patent. And Mr. Lefebvre, if you want to do a quick comparison to satisfy yourself that that's the case.

A. Yes. That seems to be the case, yes.

Q. As we go through this, if you notice anything that seems like a mismatch, please point it out to me.

A. Okay.

Q. So looking at Exhibit 8, in claim

one of the '053 patent, which aspects of claim one did you conceive of?

MR. TUMINARO: Objection. Relevance, and outside the scope.

THE WITNESS: I am not a Lawyer, nor very knowledgeable in law terms or anything. So I can talk to you about the arbiter that we conceived in the R400 design and that we submitted to the law office for the patent draft, but I cannot really comment on the text in this -- in these claims, because I don't have the right law background knowledge to do so.

BY MR. NESE:

Q. So when you stated in paragraph seven of Exhibit 1 that:

"Steve Morein, Andy Gruber and I collectively conceived of the graphics processing system in the '053 patent."

Are you saying that you don't have the right law background to make that statement?

MR. TUMINARO: Objection, form.

THE WITNESS: When I stated that Steve Morein, Andy Gruber and I collectively conceived the graphics processing system in the '053 patent I mean that we specified and created the invention as used in the R400 device and that we also filed a patent on that device called the '053 patent.

BY MR. NESE:

Q. Okay. And I would like to know which parts of claim one you, yourself, had thought of and specified.

MR. TUMINARO: Objection. Outside the scope.

THE WITNESS: Again, reading claim one would imply that I know with the -- within the scope of the law what claim one means, and I don't have the right knowledge to do so. I am not a product -- I am not a patent Attorney, nor do I have a law background.

BY MR. NESE:

Q. So you are not able to say which aspects of the graphics processing system in the

'053 patent you conceived of?

MR. TUMINARO: Objection.

BY MR. NESE:

Q. Is that right?

MR. TUMINARO: Objection, form. Sorry.

THE WITNESS: I can tell you which portions of the sequencer in the R400 I conceived, but I cannot really tell you how that maps to the claims of the patent, because again that would mean a knowledge of the law and interpretations of the words of that claim within the scope of the law, which I don't have the right tools or the right background to do.

BY MR. NESE:

Q. And would your answer be the same for the '871 patent that's involved in the 326 proceeding?

A. Yes, my answer would be the same for any patent, actually.

Q. So it would be the same for the '369 patent in the 330 IPR proceeding as well;

correct?

A. Yes, it would be the same.

Q. I would like to go back to Exhibit 1, which is your Declaration in the 325 proceeding and please turn to page 19, and I would like to look at paragraph 34.

A. I am on page 19.

Q. Okay. So paragraph 34 reads:

"After conceiving of the design for the R400 my colleagues and I worked to implement it."

Did I read that correctly?

A. Yes, you read that correctly.

Q. Who are you referring to when you -- by "my colleagues" here?

A. By my colleagues and I, I meant the people that were working on the R400 design inside the Boston and Orlando and Toronto sites.

Q. So does that include Mr. Steve Morein?

A. Yes, that would include Mr. Steve

Morein, although he is more of an architect and not an RTL designer. So he participated in the sense of making modifications to the specification based on the update received by the implementation team, if any.

Q. And by "my colleagues" here in paragraph 34, are you referring to Mr. Gruber as well?

A. Yes, I would be referring to Mr. Gruber as well.

Q. And also Mr. Skende? Is he included in what you mean by "my colleagues" in paragraph 34?

A. Yes, although, you know, he is also co-inventor of other patents.

Q. And would "my colleagues", as you use the term in paragraph 34, also include all the people whose names appear on pages 23 through 34 of Exhibit 1?

MR. TUMINARO: Objection. Form.

THE WITNESS: So it would definitely include at least the subset of these people. We

need to go in the details of exactly all the work that these persons did to understand whether or not they were working on actual implementation of the R400, or testing aspects of it or libraries that are needed to support the design.

BY MR. NESE:

Q. So just looking at this list, you are not able to say who was involved with implementing the R400 project?

A. No. I would have to go deeper and study this list a lot more to really detail out which persons worked on the actual implementation versus which persons worked on libraries and tests that are needed for the implementation.

Q. Let's flip ahead please to page 31 of Exhibit 1. What are you showing on page -- I guess it's really page 31 through 56 of Exhibit 1.

A. Your question is what I am showing in this?

Q. Yes.

A. This Exhibit is meant to show the

amount of work that went into the R400 depot, to show that, you know, which days and which parts of the file structures got touched by who, to show that we were collectively diligently working every day, every open day any way, on the R400 design or some aspects related to the R400 design and testing.

Q. And were you, yourself, working on the R400 on every single day in this calendar?

MR. TUMINARO: Objection, form.

THE WITNESS: I was working on the R400 every single day that I was at the office but, as you know, we have holidays and I also took vacations. But every day that I was present in the office; yes, that's what I was working on.

BY MR. NESE:

Q. So if I were to look at any given reference here in this calendar, for example, on August 28th I see "Lib., 434. Arch page, 44. Test page 510." If I go and look on those documents for any given day in this calendar, will I find your name on one of those entries?

A. So you would definitely find my name on some of those entries, but these are -- these dates and Exhibits show the actual perforce submissions of the documents. There is a lot of work that occurred in between those submissions and, you know, even though I could have been working on the sequencer spec. or the emulator every day, it -- depending on the scope, I would not have checked in my work at the end of the night, every day possibly. Some of the reasons for not doing that, for example, is if I am working on an emulator problem and I know that if I were to check it in to the depot it would break the depot because it's incomplete, then I would have refrained from checking it in, and in that case you would not have seen any mark. Similarly, when I am working on the spec, editing it and adding or removing data, we only try to check in the spec when a substantial amount of work has been done so that people can go and read about it, and so again, you will not see submissions from me every day because of that

reason, but that doesn't mean that I was not working on either the spec. or the emulator or other aspects relative to the R400 program.

Q. I see. So this diligence calendar in Exhibit 1 is showing dates that you and your colleagues were checking in documents; correct?

A. Yes, that is correct. It's showing you the dates that are recorded into our perforce server which only records whenever somebody makes a submission to the depot. So checking in.

Q. These are check-in dates, not work dates; correct?

MR. TUMINARO: Objection, form.

THE WITNESS: These are check-in dates, which means that in between each check-in dates a certain amount of work occurred.

BY MR. NESE:

Q. And how do you know that in between these check-in dates a certain amount of work occurred?

A. From this calendar you wouldn't know, but again because of the nature of the

perforce depot, you can go on any of these check-in dates and verify exactly what changed in which files and who made those changes, and so that would give you an idea of the amount of work that occurred for that check-in to occur.

Q. By looking at what was changed in the files, how would that give you an idea of the amount of work that occurred for that check-in?

A. Well, for example, if you look at the C code that gets produced in the emulator, the tool will give you line by line which lines of codes were added, edited, removed, so that doesn't tell you how much time the designer put into making those changes, by just looking at the amount of lines of code added. You could infer a theoretical amount of work that went in for that check-in to occur. Similarly, for the specifications you have traceability within the file that allows you to see how much of the specification actually changed from the previous version to the next.

Q. So if I have a file that was checked

in, say, on Tuesday August 28th and then checked in again on Wednesday September 5th, does the perforce system show me that that individual was working on that document every single day between the two check-in dates?

A. No. The perforce tool only records dates and sizes of files whenever you do a check-in. So it will let you know how much -- sorry -- each file changed between the check-ins. It is not going to let you know when any given person has been working on that file or been doing something else. That's the amount of traceability we have.

Q. I would like to flip to page 32 of Exhibit 1, please. And there are no entries for Tuesday September 4th. Is that right?

A. Yes, that appears to be the case.

Q. And there are also no entries for Thursday September 6, 2001. Is that right?

A. Yes, that also appears to be the case.

Q. And again there are no entries for

Wednesday September 12, 2001; right?

A. This is correct, but note that these are normal working days at AMD and again, the tool only collects the check-ins and not the work that goes in between the check-in, and so it doesn't mean if there is no entry on these dates that no work actually was going on on the R400 design. It just means that we have no traceability on those days, not that work did not occur.

Q. But there are no documents to show that work did occur on, for example, Wednesday September 12th; correct?

A. There are no log-in's from perforce, but you have documents. For example, if you look at Arch 43 or 14 and again, Arch 43 of -- on Tuesday the 11th, the file changed in between those days. It means some work occurred either on Wednesday or Thursday of that week.

Q. Or it could be just Friday morning?

A. Or it could have been just Friday morning. Yes; sorry.

Q. So there is no way to tell that work was being done on each of those days in between; right?

MR. TUMINARO: Objection, form.

THE WITNESS: The perforce logs we have would not tell you that information; correct.

BY MR. NESE:

Q. Let's go back to page 31, please, of Exhibit 1. At the top of page 31 there are six different file names, and does each of these file names point to a different folder or history on perforce?

A. Yes, that's correct. Each different file name points to a different folder history on perforce.

Q. Are different types of work being performed in each of these folder histories?

MR. TUMINARO: Objection to form.

THE WITNESS: So these are different perforce directories that have -- that collect different files. I wouldn't say -- I would like

to define the different work. I didn't quite understand what you mean by that.

BY MR. NESE:

Q. Let me ask it this way: Why would something end up in, say, the sequencer emulator folder history as opposed to the parts folder history?

A. So the sequencer emulator folder history is where the emulator files and changes are stored, and so any changes to the emulator would show up in this Exhibit versus the parts directory is where the RTL files are stored, so any changes to the RTL files would show up in that Exhibit 2049.

Q. So the changes made to -- excuse me. The changes that show up in this parts folder history, Exhibit 2049, those relate to the RTL files?

A. That is correct. Those relate to the RTL files of the sequencer.

Q. And I am going to go out on a big limb here and I am going to ask you if the files

that end up in the emulator folder history, are those the emulator files for the R400 sequencer?

A. Yes. Exhibit 2048 are -- is listing all the changes that went through the emulator files for the sequencer.

Q. All right. I got a lucky guess there. What kind of files are contained in the document library history, Exhibit 2050?

A. So the document library folder history lists all the changes that are going into the libraries that are used to either build the chip or energize it. For example, when we write a test and we want to run that test on the RTL simulator, we need to use a library that allows us to talk to the graphics chip, and that -- the changes that would show in that folder are representative of the changes that are going into these library files that allows us to either run tests or stimulate the design.

Q. The functional blocks that we were talking about earlier this morning, one of them was the sequencer functional block; correct?

A. The -- you mean the part of the blocks of the chips?

Q. Yes, and I am sort of referring to the functional blocks that are on page two of Exhibit 1.

A. Yes. One of the functional blocks in that would have been the sequencer, yes.

Q. Okay. So this library, does it interact with the sequencer functional block? And by "this library" I am referring to the library we were just discussing in reference to Exhibit 2050.

MR. TUMINARO: Objection, form.

THE WITNESS: So the library does interact with the sequencer in the sense that in order to energize the sequencer you need to provide it with shaders, and in order to compile these shaders in a form that the sequencer understands them in a binary format you need to use the libraries. And so the libraries are tied to the sequencer in that regard, and they are an inherent part of the design in that way.

BY MR. NESE:

Q. And what do you mean by "energize the sequencer"?

A. By "energize" I mean stimulate the chip with external output such as, you know, a simple triangle, or geometry or an application so that you get to see the chip -- how the chip is processing that information and how it reacts to external stimulus from the driver or the libraries.

Q. So could you test the sequencer emulation code without this library?

A. No, you could not test the sequencer emulation code without the library, because you needed the library to compile the shaders and submit them to the emulation code so that you can run a test on the emulator to see what it would do.

Q. And do you need that library for any of the other functional blocks that are on page two of Exhibit 1?

A. Some of the blocks that you are

seeing on page two are fairly simple in nature, like the PA and the SC and, because of that, they had what we call a block level test bench which allowed them to be stimulated on their own, in abstraction of the rest of the chip. The sequencer and the unified shader, because of its interconnectivity and connection to lots of components of both the libraries and the blocks surrounding it, did not have such a test bench. So we had to rely on what we call the GC test bench, which is graphics core, and that particular infrastructure was all encompassing of all the blocks that you are seeing on page two. We needed all of these blocks in order to energize the sequencer because of the inability for us to have a block test bench around just the sequencer because of its intricate nature.

Q. Let's go back to page 31 of Exhibit one. There is another folder here called Architecture Folder History Exhibit 51. What sort of files are in there?

A. The files in this folder would

represent all of the R400 architectural specification, such as the R400 sequencer specification that we discussed, but they would also include all the blocks of the chip, as well as the main specification as written by Steve Morein of the R400 architecture proposal.

Q. And that main specification that Steve Morein wrote, that deals with the R400 project as a whole?

A. That -- yes, that specification tries to explain and -- how the R400 project is supposed to work and be connected as a whole; yes.

Q. And other than that main specification that we just mentioned and the sequencer specifications that you drafted, were there any other specifications in this architecture folder history?

A. In this folder you would find all the specifications for all of the blocks that you had on page two, as well the description of all the interfaces and register definitions. Yes,

that's about what you would find. It would include, for example, the SP specification, the PA specification of all the blocks.

Q. And that would include a specification for the memory controller? And I am referring now to page two of Exhibit 1.

A. Without checking the actual log, I couldn't be sure, but I would believe that yes, it would include that block as well. But the memory controller is a system block. It's not a graphics block, so I am not positive that it was put into that same directory. And in that directory you would get all of the specifications of the graphics block which are the PA, SC, RB, and all of these types of blocks.

Q. Okay. Let's look at the GFX testing folder history. What files are contained in that folder?

A. So this directory structure contains all of the tests that were written to energize or stimulate the graphics core of the R400. So they include all of the tests that were written for

what I was referring to as the GC test bench, which is the test bench that surrounds the entire graphics core. So you would see there any tests that pertained to the sequencer, the shader pipe, any scenarios of -- that one could think of. So to stimulate the chip, like one single triangle test, many triangle tests, tessellations with one texture, with many textures. All of the scenarios to both test the chips in a performance environment so to try to infer the performance of the chip or to try to functionally validate the chip as a whole.

Q. Would this include tests on individual functional blocks?

A. If I recall correctly, the GFX testing folder history only included the test for the whole chip, so while you can focus your testing on activities that a block is performing, you would not find the tests solely for that one block in this suite. They are all tests that are run on the whole chip.

Q. And the last folder on here is the

R400 shader pipe parts folder history. What files would be in there? What types of files would be in there?

A. So that folder contained the RTL files related to the SP block or the shader pipe, the block operative to do ALU operations.

Q. Now, I notice that there is -- I am sorry. Did you have more?

A. No.

Q. I notice that there is -- there is a folder history for the RTL for the sequencer; correct? That's Exhibit 2049?

A. That is correct.

Q. And then there is a folder history for the RTL for the shader pipe which you just mentioned; correct?

A. Yes, that's correct.

Q. But I also see that there is an emulator folder history for the sequencer. Is there an emulator folder history for the shader pipe?

A. Like I was saying earlier, the -- in

the emulator side the shader pipe is really represented as the library of ALU functions that is maintained under the sequencer directory, so the SP files would have been in the sequencer emulation directory, because in the emulator the shader pipe is really represented just as a library of ALU functions, and in that sense there is no real folder for those files. They would leave with the sequencer files.

Q. Let's take a short break.

BY THE VIDEOGRAPHER: Going off the record at 2:04 p.m.

BY THE VIDEOGRAPHER: This begins tape number three in the deposition of Laurent Lefebvre. We are back on the record at 2:13 p.m.

BY MR. NESE:

Q. Welcome back, sir.

A. Thank you.

Q. Throughout the day we have been discussing testing that engineers will do on graphics processors, and I want to talk a little bit more about that. Were there ever any tests

performed on the R400 project?

A. Yes, there were like a great many amount of tests performed on the R400 design. On the simulator and on the emulator we ran quite a bit of tests and, as with any projects, even today, one of the first tests that we tried to make pass is the one triangle test, because it energizes the whole shader pipeline from top to bottom -- actually energizes the whole graphics core from top to bottom, and is a very good indicator as to when things are taking shape and are functioning like we want them to be functioning.

Q. You said there are a great many amount of tests performed on the R400 project. Are we talking hundreds? Thousands?

A. I would say, if my memory is right, in the thousands.

Q. And were those tests performed on the emulator code? The RTL? Both?

A. All of the tests are typically run first on the emulator where we collect what we

call trace files, vec. files which are recording the activity of all the interfaces, and the order in which these activities occurred and what data transitioned through each of the interfaces. Then we run the exact same test on the RTL and we compare the data present into the interfaces of the RTL versus the data present in the files that we got from the emulator, compare them and check for mismatch and differences, and we also generate for every test an image to make sure that the image that's produced by the emulator matches the image produced by the RTL simulation.

Q. Are those images saved?

A. Yes, typically these images -- the emulator images are saved, which we call the golden images. Typically the simulator images are not, because we just compare the data from the simulator as soon as it gets out of the simulator, and then what gets recorded on the server is a passing or a failing notification for that test.

Q. So these golden images in the course

of the R400 project, these thousands of tests that were done on the system, how many of these golden images were generated?

A. You should see about the same amount of golden images as there are tests.

Q. So in the thousands?

A. Yes, it should be in the thousands as well. Typically one per test.

Q. Did you perform any of the tests on the R400?

A. Can you -- by "perform", what -- can you specify what you want -- I mean, what do you mean by "perform"?

Q. Well, were you the person hitting "go", to put it very crudely?

A. I see. I see. I did. I mean, typically the tests are first run in what we call the regression, because the tests are so long to run, we have a batch file that collects all the tests, and nightly or weekly a server and a script would go and pick up all the tests and run them overnight and in the morning give us

notifications of which tests passed and which tests failed. Then, depending on the error, I or somebody else would pick up a given test and run it on their local machine to debug it. And during the course of the R400 project; yes, I ran a great many tests on my personal machine, specifically the ones that failed from the regressions to find solutions and find why these given tests were failing.

Q. And were you testing the entire graphics core or only particular functional blocks?

A. So like I said, the sequencer did not have a local test bench. So in order to run a test, any test, I had to energize the whole chip. But I was focusing on the bugs that pertained to the sequencer or the shader pipe. So typically I would only run those tests that were failing inside the sequencer, so at the sequencer interface boundary, if you will. If the test was failing elsewhere in the PA, or the SC or another block, then somebody else would

pick up that test and debug it.

Q. But tests could be run on individual functional blocks; right?

MR. TUMINARO: Objection, form.

THE WITNESS: No, not in -- not -- I guess for some blocks; yes, like the PA and SC you could pick up a test and run them. But for the sequencer you needed to run the test on the whole graphics core, because there was no way to apply the correct stimulus on just the sequencer. You had to use the whole chip to energize that sequencer.

BY MR. NESE:

Q. Earlier you mentioned a single triangle test. Could you describe what a single triangle test is?

A. The single triangle test is, even to this day, probably the most significant milestone that we record in chip development. It consists of one triangle, so three vertices and the pixels that go along with it. Why is it so relevant? It's because it's a test that is simple but also

very complicated in the sense that to achieve that single triangle rendering you actually need to energize all of the blocks of the chip and run both a vertex shader and a pixel shader along with all the memory operations that are pertaining to these -- the pixels or the vertices that create that triangle. So it's a very useful test for what we call a chip ring up, because it's a fairly simple, visually easy to see output, yet it energizes the whole chip. So it allows us to really see if the chip is connected correctly from top to bottom, and it's able to process data all the way from the top of the pipe, which is the command processor, the XP, to the render backend which processes the pixels and push them to the frame buffer.

Q. So is your work done after you pass the single triangle test?

MR. TUMINARO: Objection, form.

THE WITNESS: Our work is far from done when we are done with the single triangle test. The single triangle test is showing us

that the chip is able to perform a simple operation from top to bottom. But, as you know -- you know, our chips have to be working in all conditions and every condition. In order to productize a chip it has to never hang. It has to never show corruptions. So after you are done with the single triangle you need to continue working and find all the corner cases, all the texture formats that the chip could support, all of the programs or the various programs that the chip can support, all of the triangle flavours that we support in the chip, the polygon types, whether there is dotted line. You name it. So the first triangle is a significant milestone in the sense that it shows us that the chip is able to do the operations it's intended to, but it's not a sign that you can productize a chip. In order to have something you can productize, you have to run these thousand other tests to validate all of the corner cases and make sure that the chip is 100 percent solid and 100 percent robust before you tape it up.

BY MR. NESE:

Q. Right. Because most graphics cards need to be able to process more than just one triangle; right?

A. Yes. Absolutely. More than one triangle, but it's -- once you process one triangle all of the other triangles look the same, really. So it's more of a stress testing that you need to do -- after you prove the concept that you are able to process that one triangle, a single triangle or ten thousand triangles in a chip, it's all the same thing. It's just the amount of data that you are sending it. Each of the triangles are really doing the same programs, the same things that the one triangle you were trying to do. So you need to write more tests to make sure you cover all of the corner cases that the single triangle test in its nature might not touch, but in another sense once you prove that you can push one triangle to the chip, you prove that it can work and perform and do the things it's supposed to do.

Q. To do everything it's supposed to do?

MR. TUMINARO: Objection, form.

THE WITNESS: Again, to have a product you need to make sure all the corner cases are hit, and so the single triangle gives you the basics. It's like the chip can do what it's supposed to do at its core. Is it doing all it's supposed to do? Can you validate that with the single triangle test? No, you cannot. You need to do very many other tests to make sure that all of the other corner cases that you are going to encounter in real-life scenarios are going to be covered, but with the one triangle test you get that the core functionality is there, that all the blocks are connected properly and that all the functions can work in the simplest of forms.

BY MR. NESE:

Q. I am handing you now what's been marked as Exhibit 9.

Exhibit 9 was marked for

identification.

BY MR. NESE:

Q. And I will note for the record that, Counsel, your courtesy copies are double-sided just to save my back when I was bringing these documents over, but the Witness's is single-sided. Mr. Lefebvre, can you take a look at Exhibit 9 and tell me if you recognize this?

A. Yes. This is the log file for the R400 sequencer emulator directory.

Q. And this is Exhibit 2048 to the 325 proceeding IPR. And is this the same Exhibit that you are mentioning at the top of page 31 of Exhibit 1?

A. Yes, that would be Exhibit 2048 of page 31 of Exhibit 1.

Q. Okay. I would like you to turn to page 50 of this document, and it's about midway through. There are actually four pages per physical page.

A. Yes.

Q. Okay. At the top of page 50 there is

a change 48140 on August 26, 2002. And based on reading this document here, are you the one who logged that change?

A. Yes, according to this log, I am the one that logged that change.

Q. And how do you know that?

A. Because my handle "LLefebvre" is tagged to the change list number.

Q. And okay. And here you say:

"Fixed three bugs in the HW accurate interpolators."

Does HW stand for hardware?

A. That is correct. So this will mean that three bugs were fixed in the emulation portion of the hardware accurate version, which is trying to have the same exact precision of the hardware for those types of operations.

Q. And further down in this same change, you wrote this fixes both primb -- excuse me, this fixes both: Primlib_template_simple_triangle.cpp and milestone_ tri.cpp.tests. That's a mouthful. Did

I get it right?

A. Yes.

Q. Okay. Can you explain what you meant when you said you fixed these two test files?

A. So this was a while ago, so I don't really recall. Again, I am going to give you the version of what -- if I am reading this today, what I would infer it means. It doesn't mean that it's what I was thinking back in the date. I don't remember.

Q. Sure. I can only ask for your best recollection.

A. So if I am seeing this, is that somehow when the I's and J's, which are the parameters that we use for interpolations, were zero and the alpha was also zero, the de-normalization was occurring in the emulator and needed to be -- so that probably resulted in some kind of corruption of the test because of the emulator not matching the hardware precision somehow. So that is saying that by making this

change in the emulator, we were able to get these two tests to pass.

Q. And were these two tests failing prior to this?

MR. TUMINARO: Objection, form.

THE WITNESS: I can only assume they were. I mean, it doesn't say that they were. But yes, if it says "fixes", the underlying assumption is that they were failing before, but I don't remember.

BY MR. NESE:

Q. Fair enough. Do you recall what each of these two tests refer to? I am not going to try to barf out the file names again. Do you recall what they are referring to?

A. They seem to be referring to some kind of one triangle test. I don't know what the milestone or really what it's trying to do, but they seem to be some flavour of single triangle tests.

Q. And let's look down now at page 51 of Exhibit 9. I would like to direct your

attention about halfway down that page to change 46036 on August 14, 2002. You are the one who made this change; correct?

A. Yes, from the log file it seems that I was the one that made this change.

Q. So you start out this text by writing: "Fixed the clamping problem." Do you recall what you might have meant by that?

A. Unfortunately, I have no recollection of what this could have meant. Sorry.

Q. No, that's okay. You continue:

"Note however that now there is no triangle drawn in the test because the clamping causes the triangle to be degenerate."

What did you mean there?

A. A degenerate triangle is a triangle that has all three vertices along the same axis, so basically transforms the area into a zero area triangle and makes the triangle disappear. I can

only infer that after fixing whatever the clamping problem was, the test or the triangle that used to be drawn became degenerate because it wasn't clamped any more and that made it disappear.

Q. What does it mean to be clamped?

A. Clamped means, for example, if you say clamped to zero and you give me a negative number, a hardware that clamps to zero will transform that negative number to zero. Or typically, in hardware language, clamping is used whenever you change the precision. So if you go from a 32-bit precision number down to a 16-bit precision number, you are going to have to clamp some bits.

Q. Okay. You can put Exhibit 9 aside. Earlier today you mentioned that the R400 project was never taped out. At what point was the -- was work on the R400 project stopped?

MR. TUMINARO: Objection, form.

THE WITNESS: Like I said before, I don't really recall, because for myself and most

of the team that change was really transparent, because the design was really the same across the project. So in my memory this is all a continuation of the same work. I actually don't remember the boundary between the R400 and the Xenos program, when it actually occurred, because it's all the same thing for the shader core team.

BY MR. NESE:

Q. Do you have an understanding for why the R400 project stopped in favour of the Xenos chip?

A. Again, my understanding is that Microsoft came to us with a concept proposal, and we were developing a chip that aligned with their needs. And so we only had one design team or at the time had only one -- sorry. Let me rephrase that. We had two design teams. One was applied already on the chip that was going to market which was the R300 and the R500, but the design team I was working on and the design team applied to the R400, there was only one such design team. So when the company decided to go forward with

that Microsoft project, we had to apply the one design team we had towards that project and, in that sense, that's what cancelled the R400 project, because we just applied the design team towards that new project. Again, for us it was mostly no change. It's just names of directory change. That's it.

Q. Okay. I don't have any other questions, subject to any questions that your Attorneys may ask you.

MR. TUMINARO: Could we just take five minutes?

MR. NESE: Absolutely.

BY THE VIDEOGRAPHER: Going off the record at 2:38 p.m.

BY THE VIDEOGRAPHER: Going back on the record at 2:51 p.m.

MR. TUMINARO: I have no questions. This deposition is closed.

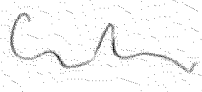
MR. NESE: Mr. Lefebvre, thank you for your time.

MR. TUMINARO: Read and sign.

BY THE VIDEOGRAPHER: This concludes the deposition of Laurent Lefebvre. We are going off the record at 2:52 p.m.

C E R T I F I C A T E

I, C.L. KLEIN, Official Court Reporter, duly sworn as such, DO HEREBY CERTIFY that the foregoing is a true and faithful transcription of the evidence herein, AND I HAVE SIGNED:



C.L. KLEIN, O.C.R.

Laurent Lefebvre c/o
STERNE, KESSLER, GOLDSTEIN & FOX P.L.L.C.
1100 New York Avenue, NW
Washington, DC 20005

Case: LG Electronics, Inc. v. ATI Technologies ULC
Date of deposition: November 13, 2015
Deponent: Laurent Lefebvre

Please be advised that the transcript in the above referenced matter is now complete and ready for signature. The deponent may come to this office to sign the transcript, a copy may be purchased for the witness to review and sign, or the deponent and/or counsel may waive the option of signing. Please advise us of the option selected. Please forward the errata sheet and the original signed signature page to counsel noticing the deposition, noting the applicable time period allowed for such by the governing Rules of Procedure. If you have any questions, please do not hesitate to call our office at (202)-232-0646.

Sincerely,
Digital Evidence Group

Digital Evidence Group, L.L.C.
1726 M Street, NW, Suite 1010
Washington, D.C. 20036
(202) 232-0646

SIGNATURE PAGE

Case: LG Electronics, Inc. v. ATI Technologies ULC
Witness Name: Laurent Lefebvre
Deposition Date: November 13, 2015

I do hereby acknowledge that I have read and examined the foregoing pages of the transcript of my deposition and that:

(Check appropriate box):
( ) The same is a true, correct and complete transcription of the answers given by me to the questions therein recorded.
( ) Except for the changes noted in the attached Errata Sheet, the same is a true, correct and complete transcription of the answers given by me to the questions therein recorded.

_____ _____
DATE WITNESS SIGNATURE

_____ _____
DATE NOTARY

Digital Evidence Group, LLC
1726 M Street, NW, Suite 1010
Washington, D.C. 20036
(202)232-0646

ERRATA SHEET

Case: LG Electronics, Inc. v. ATI Technologies ULC

Witness Name: Laurent Lefebvre

Deposition Date: November 13, 2015

Page No.	Line No.	Change

_____ _____

Signature Date