

Declaration Of Jeffrey Fischer
U.S. Patent No. 9,798,967

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

SML INTELLIGENT INVENTORY SOLUTIONS LLC
and SML (USA) INC., Petitioners,

v.
ADASA INC., Patent Owner.

Patent No. 9,798,967

DECLARATION OF JEFFREY FISCHER

I declare that all statements made herein on my own knowledge are true and that all statements made on information and belief are believed to be true, and further, that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code.

Nov 1, 2024

Date

By:



Jeffrey Fischer

TABLE OF CONTENTS

	Page
Table of Exhibits	xi
I. Introduction.....	1
II. Background and Qualifications	1
III. Relevant Legal Standards	5
A. Claim Construction	5
B. Obviousness.....	7
IV. Summary of Conclusions.....	12
A. Ground 1A.....	12
B. Ground 1B	12
C. Ground 1C	12
D. Ground 2A.....	13
E. Ground 2B	13
F. Ground 2C	13
V. THE '967 PATENT.....	13
A. Technology Background	13
1. IP Addressing and Subnetting.....	20
2. Computer Memory Allocation.....	29
3. ISO 15693-3	33
4. The Auto-ID Center Built on These Concepts and Methods to Develop the EPC standard	33
B. The Alleged Invention.....	37
C. The Effective Filing Date of the '967 Patent Is May 21, 2008.....	40
D. The Prosecution History	40
E. Ex Parte Reexamination History	41
F. Litigation History	43
G. Level of Ordinary Skill in the Art.....	44

Declaration Of Jeffrey Fischer
U.S. Patent No. 9,798,967

VI.	Claim Construction.....	44
A.	How the Challenged Claims Are To Be Construed	44
1.	“An Allocated Block of Serial Numbers”.....	46
2.	“Uniquely corresponding to” Limitation	47
VII.	Overview of the prior art	48
A.	IBM200 (Ex-1007).....	48
B.	Kuhno (Ex-1008).....	55
C.	Bowers (Ex-1009)	61
D.	Tilson (Ex-1010)	65
E.	EPC Tag Data Standards Version 1.3 (Ex-1011).....	67
F.	Nitzan (Ex-1022).....	69
G.	Forster (Ex-1023)	70
VIII.	The Challenged Claims Of The '967 Patent Are Unpatentable	71
A.	Ground 1A: Challenged Claims 1, 6-8, 10-12, and 20 Are Unpatentable as Obvious over IBM200 in View of Kuhno.....	71
1.	A POSITA would have been motivated to combine IBM200 and Kuhno	71
2.	Independent Claim 1	73
	[1pre] An RFID transponder comprising:	73
	[1A] a substrate;	74
	[1B] an antenna structure formed on the substrate;	74
	[1C] an RFID integrated circuit chip which is electrically coupled to the antenna structure;	75
	[1D] wherein the RFID integrated circuit chip is encoded with a unique object number, the unique object number comprising an object class information space and a unique serial number space, 76	

	[1E] wherein the unique serial number space is encoded with one serial number instance from an allocated block of serial numbers, the allocated block being assigned a limited number of most significant bits, and.....	79
	[1F] wherein the unique serial number space comprises the limited number of most significant bits uniquely corresponding to the limited number of most significant bits of the allocated block and of remaining bits of lesser significance that together comprise the one serial number instance.....	86
3.	Claim 6: The RFID transponder of claim 1, wherein the allocated block is defined by at least 3 most significant bits.....	88
4.	Claim 7: The RFID transponder of claim 1, wherein the allocated block is one of a plurality of allocations intermittently received by an encoder.....	89
5.	Claim 8: The RFID transponder of claim 1, wherein the RFID integrated circuit chip is comprised of silicon.....	90
6.	Claim 10: The RFID transponder of claim 1, wherein the substrate, antenna structure, and RFID integrated circuit chip are covered by a face stock material.....	90
7.	Claim 11: The RFID transponder of claim 1, wherein the RFID integrated circuit chip is disposed on the substrate.	91
8.	Claim 12: The RFID transponder of claim 1, wherein the limited number of most significant bits that correspond to the allocated block further correspond to an encoder that encodes the RFID transponder.....	92
9.	Claim 20: The integrated circuit of claim 16, wherein the limited number of most significant bits that correspond to the allocated block further correspond to an encoder that encodes the memory of the integrated circuit.....	93
B.	Ground 1B: Claims 2-5, 13-16, and 17-19 are Unpatentable Over IBM200 in view of Kuhno and TDS.....	93

Declaration Of Jeffrey Fischer
U.S. Patent No. 9,798,967

1.	A POSITA Would Have Been Motivated To Combine TDS with IBM200 and Kuhno.....	93
2.	Claim 2: The RFID transponder of claim 1, wherein the object class information space is encoded with information comprising a company prefix, an item reference, a partition value, and a filter value.....	94
3.	Claim 3: The RFID transponder of claim 1, wherein the unique object number is a portion of a serialized global trade item number.	96
4.	Claim 4: The RFID transponder of claim 3, wherein the serialized global trade item number is an EPC SGTIN-96 encoding.	96
5.	Claim 5: The RFID transponder of claim 1, wherein the unique serial number space is at least 38 bits.	97
6.	Independent claim 13:.....	98
	[13pre] A RFID transponder comprising:.....	98
	[13A] a substrate;	98
	[13B] an antenna structure formed on the substrate; and	98
	[13C] an RFID integrated circuit chip which is electrically coupled to the antenna structure,	98
	[13D] wherein the RFID integrated circuit chip is encoded with a global trade item number including a unique object number, the unique object number comprising an object class information space including a block of at least 50 bits and a unique serial number space including a block of at least 38 bits,.....	98
	[13E] wherein the unique serial number space has one serial number instance from an allocated block of serial numbers, the allocated block being assigned a limited number of most significant bits, and	99

	[13F] wherein the unique serial number space comprises at least 3 most significant bits uniquely corresponding to the limited number of most significant bits of the allocated block.	100
7.	Claim 14: The RFID transponder of claim 13, wherein the unique serial number space further comprises remaining bits of lesser significance that together with the at least 3 most significant bits comprise the one serial number instance.	100
8.	Claim 15: The RFID transponder of claim 13, wherein the at least 3 most significant bits that correspond to the allocated block further correspond to an encoder that encodes the RFID transponder.	100
9.	Independent Claim 16	101
	[16pre] An integrated circuit, comprising:	101
	[16A] a radio frequency circuit configured to backscatter a return link signal for processing by an RFID interrogator;	101
	[16B] control logic; and	101
	[16C] a memory,	102
	[16D] wherein the memory is configured with a unique object number space, the unique object number space comprising an object class information space and a unique serial number space,	103
	[16E] wherein the unique serial number space is encoded with one serial number instance from an allocated block of serial numbers, the allocated block being assigned a limited number of most significant bits, and,	103

	[16F] wherein the unique serial number space comprises the limited number of most significant bits uniquely corresponding to the limited number of most significant bits of the allocated block and remaining bits of lesser significance that together comprise the one serial number instance.....	103
10.	Claim 17: The integrated circuit of claim 16, wherein the object class information space is encoded with information comprising a company prefix, an item reference, a partition value, and a filter value.....	103
11.	Claim 18: The integrated circuit of claim 16, wherein the unique object number is a portion of a serialized global trade item number.	104
12.	Claim 19: The integrated circuit of claim 18, wherein the serialized global trade item number is an EPC SGTIN-96 encoding.	104
C.	Ground 1C: Claim 9 is Unpatentable over IBM200 in view of Kuhno and Nitzan.....	104
1.	Claim 9: The RFID transponder of claim 1, wherein the RFID integrated circuit chip is comprised of polymer.	104
D.	Ground 2A: Challenged Claims 1-8 and 11-20	105
1.	A POSITA would have been motivated to combine Bowers, Tilson, and TDS with a Reasonable Chance of Success	105
2.	Independent Claim 1	110
	[1pre] An RFID transponder comprising:.....	110
	[1A] a substrate:	110
	[1B] an antenna formed on the substrate:	112
	[1C] and an RFID integrated circuit chip which is electrically coupled to the antenna structure:	113

- [1D] wherein the RFID integrated circuit chip is encoded with a unique object number, the unique object number comprising an object class information space and a unique serial number space, 115
- [1E] wherein the unique serial number space is encoded with one serial number instance from an allocated block of serial numbers, the allocated block being assigned a limited number of most significant bits, and..... 117
- [1F] wherein the unique serial number space comprises the limited number of most significant bits uniquely corresponding to the limited number of most significant bits of the allocated block and of remaining bits of lesser significance that together comprise the one serial number instance..... 122
3. Claim 2: The RFID transponder of claim 1, wherein the object class information space is encoded with information comprising a company prefix, an item reference, a partition value, and a filter value..... 125
4. Claim 3: The RFID transponder of claim 1, wherein the unique object number is a portion of a serialized global trade item number. 126
5. Claim 4: The RFID transponder of claim 3, wherein the serialized global trade item number is an EPC SGTIN-96 encoding. 126
6. Claim 5: The RFID transponder of claim 1, wherein the unique serial number space is at least 38 bits. 127
7. Claim 6: The RFID transponder of claim 1, wherein the allocated block is defined by at least 3 most significant bits..... 128
8. Claim 7: The RFID transponder of claim 1, wherein the allocated block is one of a plurality of allocations intermittently received by an encoder..... 129
9. Claim 8: The RFID transponder of claim 1, wherein the RFID integrated circuit chip is comprised of silicon..... 130

10. Claim 11: The RFID transponder of claim 1, wherein the RFID integrated circuit chip is disposed on the substrate.130
11. Claim 12: The RFID transponder of claim 1, wherein the limited number of most significant bits that correspond to the allocated block further correspond to an encoder that encodes the RFID transponder.....131
12. Independent Claim 13:.....132
 - [13pre] An RFID transponder comprising:.....132
 - [13A] a substrate;132
 - [13B] an antenna structure formed on the substrate; and132
 - [13C] an RFID integrated circuit chip which is electrically coupled to the antenna structure,133
 - [13D] wherein the RFID integrated circuit chip is encoded with a global trade item number including a unique object number, the unique object number comprising an object class information space including a block of at least 50 bits and a unique serial number space including a block of at least 38 bits,.....133
 - [13E] wherein the unique serial number space has one serial number instance from an allocated block of serial numbers, the allocated block being assigned a limited number of most significant bits, and134
 - [13F] wherein the unique serial number space comprises at least 3 most significant bits uniquely corresponding to the limited number of most significant bits of the allocated block.....134
13. Claim 14: The RFID transponder of claim 13, wherein the unique serial number space further comprises remaining bits of lesser significance that together with the at least 3 most significant bits comprise the one serial number instance.134

14.	Claim 15: The RFID transponder of claim 13, wherein the at least 3 most significant bits that correspond to the allocated block further correspond to an encoder that encodes the RFID transponder.....	134
15.	Independent Claim 16	135
	[16pre] An integrated circuit, comprising:	135
	[16A] a radio frequency circuit configured to backscatter a return link signal for processing by an RFID interrogator;	135
	[16B] control logic; and	135
	[16C] a memory,	136
	[16D] wherein the memory is configured with a unique object number space, the unique object number space comprising an object class information space and a unique serial number space,	137
	[16E] wherein the unique serial number space is encoded with one serial number instance from an allocated block of serial numbers, the allocated block being assigned a limited number of most significant bits, and.....	137
	[16F] wherein the unique serial number space comprises the limited number of most significant bits uniquely corresponding to the limited number of most significant bits of the allocated block and remaining bits of lesser significance that together comprise the one serial number instance.....	137
16.	Claim 17: The integrated circuit of claim 16, wherein the object class information space is encoded with information comprising a company prefix, an item reference, a partition value, and a filter value.....	137
17.	Claim 18: The integrated circuit of claim 16, wherein the unique object number is a portion of a serialized global trade item number.	138

Declaration Of Jeffrey Fischer
U.S. Patent No. 9,798,967

18.	Claim 19: The integrated circuit of claim 18, wherein the serialized global trade item number is an EPC SGTIN-96 encoding.	138
19.	Claim 20: The integrated circuit of claim 16, wherein the limited number of most significant bits that correspond to the allocated block further correspond to an encoder that encodes the memory of the integrated circuit.	138
E.	Ground 2B: Claim 9	138
1.	Claim 9: The RFID transponder of claim 1, wherein the RFID integrated circuit chip is comprised of polymer.	138
F.	Ground 2C: Claim 10	139
1.	Claim 10: The RFID transponder of claim 1, wherein the substrate, antenna structure, and RFID integrated circuit chip are covered by a face stock material.	139

Table of Exhibits

Exhibit	Description
Ex-1001	U.S. Pat. No. 9,798,967 (“’967 Patent”)
Ex-1002	Ex Parte Reexamination Certificate for the ’967 Patent
Ex-1003	’967 Patent Prosecution File History
Ex-1004	Excerpts of ’967 Patent Ex Parte Reexamination File History
Ex-1005	Declaration of Expert Witness, Jeffrey Fischer (“Fischer”)
Ex-1006	Curriculum Vitae of Jeffrey Fischer
Ex-1007	U.S. Pat. No. 7,827,200 (“IBM200”)
Ex-1008	U.S. Pat. No. 7,857,221 (“Kuhno”)
Ex-1009	U.S. Patent No. 6,025,780 (“Bowers”)
Ex-1010	U.S. Patent No. 7,777,628 (“Tilson”)
Ex-1011	EPCglobal Tag Data Standards Version 1.3 (March 8, 2006) (“TDS”)
Ex-1012	Affidavit of Elizabeth Rosenberg
Ex-1013	RFID for Dummies (2005) (“ <i>Dummies</i> ”)
Ex-1014	David L. Brock, White Paper: The Electronic Product Code (EPC): A Naming Scheme for Physical Objects, Auto-ID Center, Massachusetts Institute of Technology (January 1, 2001) (“ <i>MIT White Paper</i> ”)
Ex-1015	Excerpts of David Tarnoff, Computer Organization and Design Fundamentals: Examining Computer Hardware from the Bottom to the Top (Revised 1st Ed., July 2007) (“ <i>Computer Fundamentals</i> ”)
Ex-1016	Excerpts of Bill Glover, et al., RFID Essentials (February 28, 2006) (“RFID Essentials”)
Ex-1017	Klaus Finkenzeller, RFID Handbook (May 23, 2003) (“RFID Handbook”)

Petition for *Inter Partes* Review of
U.S. Patent No. 9,798,967

Ex-1018	RFID Handbook - English Edition RFID Handbook - English Edition [http://web.archive.org/web/20030604032620/http://www.rfid-handbook.de/english/index.html]
Ex-1019	3Com Corporation, Understanding IP Addressing: Everything You Ever Wanted To Know (2001)
Ex-1020	Department of Defense, Radio Frequency Identification (RFID) Policy (July 30, 2004)
Ex-1021	Manohar Potdar, et al, Applications of RFID in Pharmaceutical Industry (Jan. 2007)
Ex-1022	U.S. Pat. No. 7,394,382 (“Nitzan”)
Ex-1023	U.S. Pat. No. 7,571,862 (“Forster”)
Ex-1024	Specification from U.S. Pat. App. No. 11/465,712
Ex-1025	Specification from Provisional U.S. Pat. App. No. 60/709,713
Ex-1026	Specification from Provisional U.S. Pat. App. No. 60/939,603
Ex-1027	ISO/IEC FCD 15693-3:2001 (E), Identification cards—Contactless integrated circuit(s) cards—Vicinity cards—Part 3: Anti-collision and transmission protocol (April 1, 2001) (“ISO 15693-3”).
Ex-1028	<i>Avery Dennison</i> Claim Construction Order
Ex-1029	Excerpts of EPCglobal Architectural Framework, EPCglobal Final Version, 1 July 2005
Ex-1030	<i>Adasa Inc. v. Avery Dennison Corp.</i> , 6:17-cv-01685, Excerpts of Trial Transcript, Dkt. 572 (D. Or. July 13, 2023)
Ex-1031	Declaration of Maria P. Garcia
Ex-1032	<i>Adasa Inc. v. Avery Dennison Corp.</i> , 55 F.4th 900 (Fed. Cir. 2022)
Ex-1033	<i>Adasa Inc. v. Avery Dennison Corp.</i> , 6:17-cv-01685, Patent Owner’s Opening Brief, Dkt. 57 (D. Or. Sep. 27, 2018)
Ex-1034	<i>Adasa Inc. v. Avery Dennison Corp.</i> , 6:17-cv-01685, Claim Construction Order, Dkt. 68 (D. Or. Jan. 22, 2019)

I. Introduction

1. My name is Jeff Fischer. I have been retained by counsel for SML Intelligent Inventory Solutions LLC and SML (USA) Inc. (“SML” or “Petitioners”) as an expert witness in this *inter partes* review to offer my opinions concerning the validity of U.S. Patent No. 9,798,967 (“the ’967 Patent”), which is assigned to ADASA Inc. (“Patent Owner” or “PO”).

2. I am being compensated for my work in this matter at my standard rate of \$500 per hour for consulting services. My compensation in no way depends on the outcome of this proceeding or the particular opinions I express, or the testimony I give.

3. My opinions are based on my reviews of the materials referenced in the body of this Declaration and the Table of Exhibits (attached above).

II. Background and Qualifications

4. Attached to this Declaration as Appendix A is a true and correct copy of my current CV, which describes my education, patents and publications, employment and research history, and professional activities and awards.

5. I have Bachelor of Science degree in Electrical Engineering (1979) and a Master of Engineering degree in Electrical Engineering (1980), both from Cornell University.

6. I have been working as an Electrical Engineer in the wireless

communications field for 40 years. I have published refereed papers, conference papers and I am a named inventor on more than 12 patents. I am currently an engineering consultant actively engaged in product design for wireless systems. My product design work includes the design of digital, analog, and radio frequency (RF) circuits and systems for wireless communication products. I design full product solutions, wireless hardware, software, low-level firmware, algorithms, and entire wireless system architectures.

7. I was employed at MIT's Lincoln Laboratory in the Analog Device Technology Group for 6 years between 1980 and 1986, and for five of those years I was a Senior Staff Member running a project to design the most advanced packet radio available. Lincoln Laboratory is a federally funded research and development center administered by the Massachusetts Institute of Technology, with specialties in advanced radio communications and radar technology.

8. After leaving MIT Lincoln Laboratory, I worked at MICRILOR, a small company in Wakefield, MA, which aimed to commercialize advanced radio technology. I spent the next 14 years at MICRILOR developing high performance, low cost radio modems, as well as other technology for communications and radar including an RFID reader design for the US Government. Techniques I developed in MICRILOR evolved into what is today the IEEE 802.11b standard, part of what is commonly referred to as Wi-Fi. I am a co-inventor on the fundamental design of

this technology.

9. MICRILOR was purchased by Proxim, Inc., and I continued to work on wireless products at Proxim for the next 3 years until 2003. During this period I worked on a wide range of radio designs from the higher level air protocols out to the antennas.

10. In 2004, I helped start Reva Systems, a company which quickly became the leader in RFID infrastructure. From 2004 to 2007, while at Reva, I served as the technical co-chair of the standards body working group to develop a new international standard for RFID. We developed the standard known worldwide as “Gen2”, which was designed for UHF and was subsequently modified to operate at HF. Gen2 is still the prevailing worldwide standard used by GS1, the international standards body for RFID and Barcode numbering, as well as ISO. During this period I also served as a liaison to ISO, ETSI, MIC (Japan) and the FCC to get UHF RFID adopted worldwide. I was one of three referees for an EPC Global “bakeoff” of 200 commercial RFID tag company’s products, including LF, HF and UHF. In this bakeoff, I critically evaluated and wrote summaries of the enormous range of technologies and products that were available. At Reva, I built a measurements laboratory and made numerous measurements on RFID readers, tags and antennas and the air-protocol performance. I also advised numerous customers how to do the same.

11. Additionally, as co-chair of the working group that developed the EPC Gen 2 standard, I was liaison to the EPC Tag Data Standard group to convey information about the EPC Gen 2 air protocol we were developing for both HF and UHF RFID tags to enable the EPC Tag Data Standard group to develop the EPC Tag Data Standard for the content and format of the memory in those tags.

12. In recent years I have also provided technical consulting services. Some examples of that work include developing near-field HF tag products, in-home wireless sensor networks, networked video multiple drones, frame-based backhaul radio for LTE systems, transmit beamforming in LTE and Wi-Fi systems, developing an HF near-field transmission system for transmission to the human eye, a Bluetooth wireless microphone system, and wireless control for robotics.

13. In light of the foregoing, I consider myself to be an expert in the fields of RFID in general, and in particular, Gen2, as well as modulation, air-protocol design, antennas, reader and tag technology and more generally, product design, wireless networking, and wireless communications systems.

14. I am named inventor on at least 12 U.S. patents. Several of these concern RFID technology.

15. At the relevant time of 2008, I qualified as a POSITA in this field (see § V.G).

III. Relevant Legal Standards

16. I have been asked to provide my opinion as whether claims 1-20 of the '967 Patent would have been obvious to a person of ordinary skill in the art ("POSITA") at the time of the alleged invention, in view of the prior art.

17. I am an engineer by education and profession. The opinions I am expressing in this Declaration involve the application of my engineering knowledge and experience to the evaluation of certain alleged prior art with respect to the '967 Patent. Aside from my experience in litigation support, my knowledge of patent law is no different than that of any lay person. Therefore, I have requested that the attorneys from Jones Day who represent SML provide me with guidance as to the applicable patent law in this matter.

A. Claim Construction

18. It is my understanding that in determining whether a patent claim is anticipated or obvious in view of the prior art, the Patent Office must apply the Phillips standard to construe the claim by giving the claim its plain and ordinary meaning, consistent with the specification and prosecution history.

19. I understand that the claim language, specification of the patent, and prosecution history are relevant to the meaning of a patent claim. I understand that the prosecution history of a patent provides the record of the examination of a patent application before the U.S. Patent and Trademark Office ("PTO" or the "Office").

The prosecution history provides evidence of how the patent examiner and the inventors understood the patent application and the claims, and can therefore be instructive on how to interpret the claims. My understanding is that extrinsic evidence may also be used in understanding the meaning of a claim term. Extrinsic evidence includes dictionaries, treatises, expert testimony, and prior art. But it is my understanding that one should first look to the intrinsic evidence in construing claims.

20. My understanding is that there are at least two circumstances where the words in a patent may differ from and not be given their plain and ordinary meaning. One circumstance is when the applicants act as their own lexicographer by clearly setting forth a definition of a claim term that may differ from the plain and ordinary meaning it would otherwise possess. Another circumstance is when the applicant includes or provides an intentional disclaimer, or disavowal, of claim scope. My understanding is that an applicant may act as their own lexicographer, or disclaim or disavow claim scope, in either the specification or the prosecution history of the patent. My understanding is also that the applicant may act as a lexicographer, or disclaim or disavow claim scope, by making amendments to the claims during prosecution, or by making assertions to the PTO about the differences between the claimed inventions and the prior art.

21. Because a patent claim is interpreted based on what it would mean to a

POSITA, the education, experience, and knowledge of a POSITA are also relevant to determining the scope and meaning of the patent claim. It is my understanding the Board will determine the meaning of any disputed claim limitations and that, for the remaining claim language, I should apply the plain and ordinary meaning of the claim language as that language would be understood by a POSITA as of the date of the inventions.

22. I understand the claim terms should be construed consistently when the same term appears in multiple limitations. Similarly, I also understand that, within a patent, different words in claims should generally be interpreted differently.

B. Obviousness

23. I understand that a patent claim is invalid as obvious when the differences between the claimed subject matter and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a POSITA.

24. I also understand that an obviousness determination includes the consideration of various factors such as: (1) the scope and content of the prior art, (2) the differences between the prior art and the asserted claims, (3) the level of ordinary skill in the art, and (4) the existence of secondary considerations such as commercial success, long-felt but unresolved needs, failure of others, etc.

25. I understand that a reference is considered appropriate prior art if it falls

within the field of the inventor's endeavor. In addition, a reference is prior art if it is reasonably pertinent to the particular problem with which the inventor was involved. A reference is reasonably pertinent if it logically would have commended itself to an inventor's attention in considering his problem. If a reference relates to the same problem as the claimed invention, that supports use of the reference is an obviousness analysis.

26. I understand that an obviousness evaluation can be based on a combination of multiple prior art references. I understand that the prior art references themselves may provide a suggestion, motivation, or reason to combine the references. I also understand that the motivation to combine multiple prior art references may come from the common sense of a POSITA.

27. I further understand that the obviousness analysis recognizes that market demand, rather than scientific literature, often drives innovation, and that a motivation to combine references may be supplied by the direction of the marketplace.

28. I understand that when a technique has been used to improve one device and a POSITA would recognize that the technique would improve similar devices in the same way, then using the technique is obvious unless its actual application is beyond the skill of a POSITA.

29. I also understand that practical and common sense considerations

should guide a proper obviousness analysis, because familiar items may have obvious uses beyond their primary purposes. I further understand that a person of ordinary skill in the art looking to overcome a problem will often be able to fit together the teachings of multiple prior art references like a puzzle. I understand that obviousness analysis therefore takes into account the inferences and creative steps that a POSITA would employ.

30. I understand that a particular combination may be shown to have been obvious by showing that it was obvious to try the combination. When there is a design need or market pressure to solve a problem and there are a finite number of identified, predictable solutions, a POSITA has good reason to pursue the known options within his or her technical grasp because the result is likely the product not of innovation but of ordinary skill and common sense.

31. I understand that the combination of familiar elements according to known methods is likely to be obvious when it does no more than yield predictable results. When a work is available in one field of endeavor, design incentives and other market forces can prompt variations of it, either in the same field or a different field. When a POSITA can implement a predictable variation, the patent claim is obvious.

32. In sum, I understand that prior art teachings are properly combined where a POSITA having the understanding and knowledge reflected in the prior art

and motivated by the general problem facing the inventor, would have been led to make the combination of limitations recited in the patent claim. The prior art references themselves, or any need or problem known in the field of endeavor at the time of invention, can provide a reason for combining the elements of multiple prior art references in the claimed manner.

33. It is further my understanding that a proper obviousness analysis focuses on what was known or obvious to a POSITA, not the patentee. Accordingly, I understand that any need or problem known in the field of endeavor at the time of invention and addressed by the patent can provide a reason for combining the prior art references in the manner claimed.

34. I understand that a proposed obviousness combination does not need to be the preferred, or most desirable, combination available, in order to render a patent claim obvious. Instead, there must be something that suggests the desirability of the proposed combination, not necessarily something that suggests that the proposed combination is the most desirable combination available.

35. Similarly, I understand that for purposes of obviousness, when a motivating benefit comes at the expense of another benefit, that does not nullify the proposed obviousness combination or modification. Likewise, a proposed obviousness combination does not fail because it may be inferior to other solutions used for the same use.

36. I also understand that a determination of obviousness based on teachings from multiple prior-art references does not require an actual, physical substitution of elements between the multiple prior-art references. Likewise, the test for obviousness is not whether the features of a secondary prior-art reference may be readily incorporated into the structure of the primary prior art reference. Instead, the test is whether a claim is rendered obvious by the teachings of the prior art as a whole.

37. I understand that a claim can be obvious in light of a single, prior-art reference, without the need to combine multiple prior-art references, if the limitations of the patent claim that are not found explicitly or inherently in the prior-art reference can be supplied by the knowledge or common sense of a person of ordinary skill the art.

38. I understand that in analyzing obviousness, one may consider what are called secondary considerations of non-obviousness. I understand that such secondary considerations may include: (1) a long felt but unmet need in the prior art that was satisfied by the patent claim; (2) commercial success of the claimed subject matter; (3) unexpected results achieved by the invention of the patent claim; (4) praise of the invention by others skilled in the art; (5) taking of licenses under the patent by others; (6) deliberate copying of the invention; (7) failure of others to find a solution to the long felt need; and (8) skepticism by experts. I

further understand that contemporaneous and independent invention by others is a secondary consideration supporting an obviousness determination.

39. I also understand that there must be a relationship, or nexus, between any such secondary considerations and the claimed invention. I further understand that evidence of licensing indicates non-obviousness when the licenses were taken due to the merits of the claimed invention, as opposed to a desire to avoid incurring litigation costs, as a result of prior business relationships, or other economic reasons.

IV. Summary of Conclusions

A. Ground 1A

40. In my opinion, claims 1, 6-8, 10-12, and 20 of the '967 Patent are unpatentable as obvious over US Pat. 7,827,200 ("IBM200") in view of US Pat. 7,857,221 ("Kuhno").

B. Ground 1B

41. In my opinion, claims 2-5, 13-16, and 17-19 of the '967 Patent are unpatentable as obvious over IBM200 in view of Kuhno and EPCglobal Tag Data Standards Version 1.3 ("TDS").

C. Ground 1C

42. In my opinion, claim 9 of the '967 Patent is unpatentable as obvious over IBM200 in view of Kuhno and U.S. Pat. 7,394,382 ("Nitzan").

D. Ground 2A

43. In my opinion, claims 1-8, and 11-20 of the '967 Patent are unpatentable as obvious over US Pat. 6,025,780 ("Bowers") in view of US Pat. 7,777,628 ("Tilson") and TDS.

E. Ground 2B

44. In my opinion, claim 9 of the '967 Patent is unpatentable as obvious over Bowers in view of Tilson, TDS, and Nitzan.

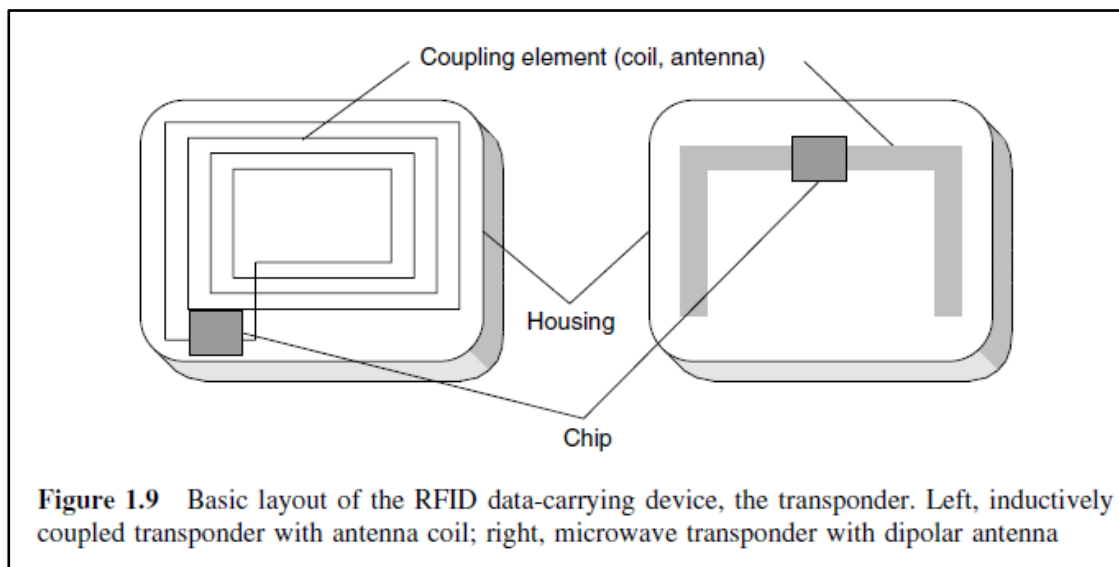
F. Ground 2C

45. In my opinion, claim 10 of the '967 Patent is unpatentable as obvious over Bowers in view of Tilson, TDS, and US Pat. 7,571,862 ("Forster").

V. THE '967 PATENT

A. Technology Background

46. Radio Frequency Identification (RFID) systems are useful for locating and identifying objects. RFID systems are made up of "two components: the transponder, which is located on the object to be identified" and is "the actual *data-carrying device*," and the interrogator or reader, "which is a data capture device." Ex-1017, 7 (emphasis original). An RFID transponder, or tag, consists of an antenna integrated with a microchip, or integrated circuit ("IC"). *Id.* The antenna and IC are connected through and held together by a substrate. *Id.*, 58. The figure below, *id.*, 9, depicts the basic layout of an RFID tag.



47. The RFID tag's IC chip contains a memory area that is programmed with information that uniquely identifies the tag, as well as storing and processing data. Ex-1017, 298-99, 296, 273. The communication between the microwave RFID tag and reader occurs through a process called backscatter modulation. Ex-1017, 13, 23. The IC modulates the tag's reflection of the reader's RF signal, encoding the tag's data in the reflected signal. *Id.* The communication between the HF RFID tag and the reader occurs through field modulation. The reader then decodes this modulated signal to retrieve the tag's unique identifier and any other stored information. *Id.*

48. Many of the tags produced at the time of the invention were passive tags. Ex-1012, 20. Passive RFID tags do not have their own power source. *Id.* Instead, they receive energy from the electromagnetic or magnetic field created by

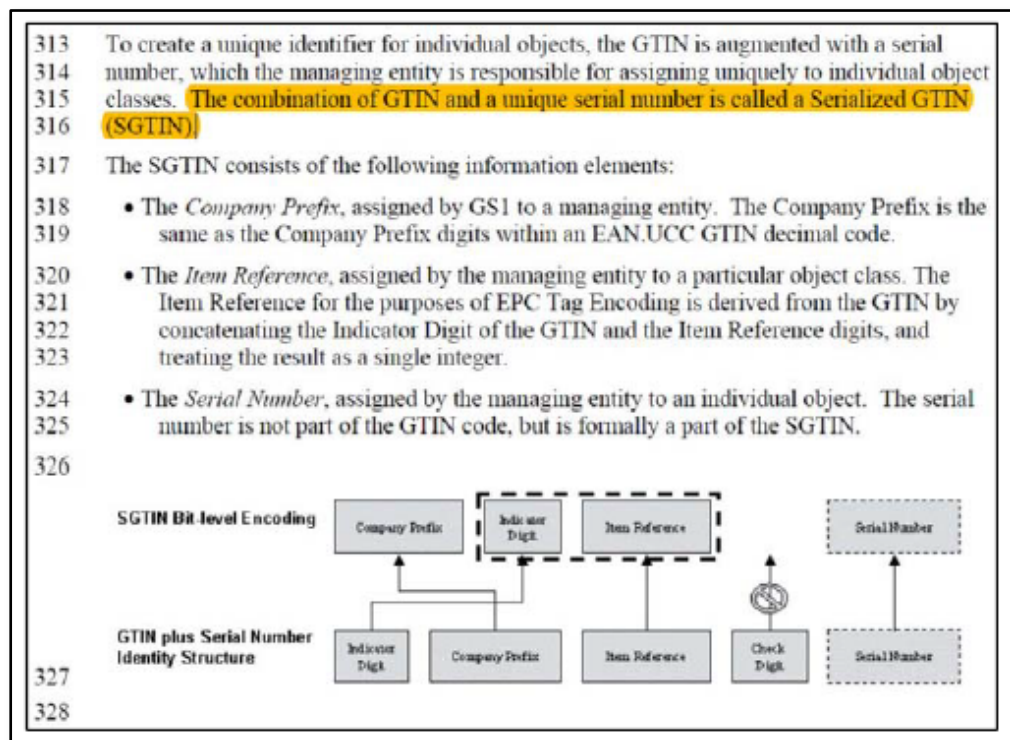
the interrogator and backscatter information when requested. *Id.*

49. RFID tags, including passive tags, store data that uniquely identifies the object to which the tag is affixed. Ex-1014, 6. Various methods have been used to identify assets and track them in transit or as they are being purchased. Such identification forms include coded numbers, such as Uniform Product Codes (“UPC”) and Serialized Shipping Container Codes (“SSCC”), that are stored physically on RFID tags. Ex-1014, 5-6. These tags identify some property of the asset which may be used to track, collate, manage or convey the assets in an asset or inventory management/control system. *Id.* The identification data can be stored in memory in a standard format for RFID numbers under the umbrella description of electronic Product Codes (“EPC”). Ex-1011, § 3.2, Line 522-28.

50. The EPC is an object encoding scheme for object identification created by the MIT Auto-ID Center. *See generally* Ex-1014. It used to be managed by EPCglobal but is now managed by GS1. *Id.*; *see also* Ex-1001, 7:61-65. The EPC is like an object’s license plate in that it functions as an identifier that uniquely identifies an associated object. *See* Ex-1014.

51. TDS describes the details of the codes stored in the EPC memory space. *See* Ex-1011. The TDS enumerates several types of standardized codes that can be used in the EPC memory. One in particular is the SGTIN, which the EPC TDS describes as two independent codes. *Id.*, § 2.1.2.1. The first code is the GTIN (Global

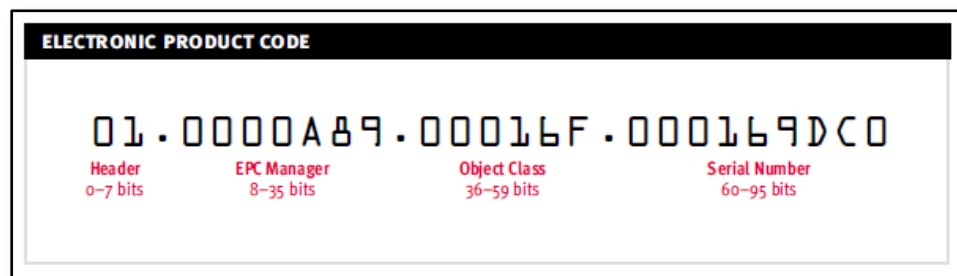
Trade Item Number), “which is a particular kind of product of SKU.” *Id* § 1.1.2.1, Line 343. For example, the GTIN may indicate the item is a 32 oz. box of Kellogg’s Corn Flakes. The second code is the serial number. Concatenating the serial number with the GTIN results in a number identifying a particular box of Corn Flakes that the tag is attached to. The TDS explains that the “Serial Number” is “assigned by the managing entity to an individual object...the serial number is not part of the GTIN code, but is formally a part of the SGTIN.” *Id.*, Line 360-61. In other words, an SGTIN identifier consists of two codes: a GTIN that identifies a product class or type concatenated with a serial number that uniquely identifies an individual instance of the product. As such, worldwide use of SGTIN requires uniqueness. The EPC is stored in an RFID transponder/tag as a string of bits. *Id.*



52. It was well known in the RFID art that the ICs used in RFID tags store binary data in memory. Ex-1011, § 2.2.1, Line 572.

53. The EPC is built around a basic hierarchical idea that has been used to express a wide variety of different numbering systems. *See* Ex-1014. The EPC structure is composed of a fixed number of bits organized into specific data fields. Each field is assigned a portion of the EPC's total bits to effectively categorize and track items, with the goal of being human-readable and capable of handling large quantities of items. The EPC structure is divided into four fields: the header, EPC manager, Object Class, and Serial Number. The header signifies that the data represents an EPC rather than another type. The EPC manager field identifies the

manufacturer or organization overseeing the remaining fields. The Object Class designates the type or category of the tagged product, while the Serial Number provides a unique identifier for each individual item. For example, the figure below depicts the data fields a 96-bit EPC, with 28 bits assigned to the EPC manager, 24 bits to the Object Class, and the remaining 36 bits to the Serial number. *Id.*



54. In 2001, David Brock, co-director of the MIT Auto-ID Center team tasked with creating a naming scheme to identify physical objects, published a white paper (“MIT White Paper”) that clearly demonstrates that the EPC was modeled on established hierarchical numbering systems in various fields that had previously addressed similar serialization challenges. The MIT White Paper details that during the development of the EPC standard, the MIT team evaluated existing product and shipment numbering conventions, as well as other successful schemes from different domains. *See* Ex-1014. These included well-established numbering systems such as vehicle registration numbers, postal addresses, and telephone numbers, which were recognized for their widespread adoption and effectiveness. For instance, the team analyzed automobile license plates and noted that they are managed by individual

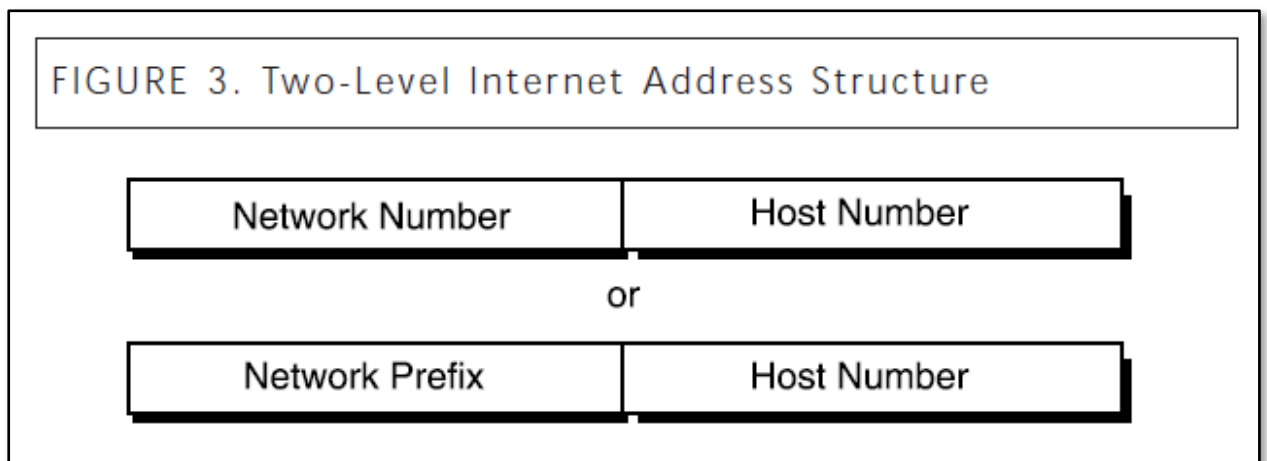
states and typically consist of six or fewer alphanumeric characters. *Id.*, 8-9. The MIT team noted that license plates are engineered for easy readability and memorability, and are therefore often organized into one or two segments containing three or four characters each. *Id.* The MIT White Paper acknowledged the flexibility of these partitioned systems, suggesting that more stringent numbering formats, such as three letters followed by three numbers, could be implemented as necessary. *Id.* This adaptability ensures that the numbering partitions can be modified to meet evolving requirements. *Id.*

55. Furthermore, the MIT White Paper shows that the MIT team valued human readability in the EPC design. *Id.*, 8-11. Their examination and discussion of telephone numbers, Social Security numbers, and International Standard Book Numbers (ISBNs) highlighted the role of concise and easily recallable partitions. Telephone numbers, for example, are divided into segments that correspond to network regions, subregions, and individual extensions. This hierarchical structure allows users to quickly identify and recall the different components of a telephone number, making it easier to remember and communicate. *Id.* The team concluded that any new numbering system intended for widespread use must adhere to similar principles of concise, easily recallable partitions. *Id.*, 12, 16. Accordingly, the last seven digits of a phone number may find duplicates around the country, but the last seven digits with the area code makes the phone number globally unique.

56. The MIT White Paper also analyzes the Internet Protocol (“IP”) address encoding scheme, which the MIT team considered particularly relevant to their serialization efforts, as it tackled a very similar problem: the identification of individual units on a global scale through a decentralized system. *Id.*, 11.

1. IP Addressing and Subnetting

57. An “IP address is a globally accepted method for identifying computers attached to the internet. The address is a standardized, compact binary address that makes computations like routing decisions efficient. Specifically, an IP address encodes the identification of the network to which a host attaches as well as the identification of a unique host on that network.” Ex-1014, 11. Thus, “the first part of an Internet address identifies the network on which the host resides, while the second part identifies the particular host on the given network.” Ex-1019, 3. As such, this creates a “two-level addressing hierarchy,” *id.*, as depicted in the figure below.

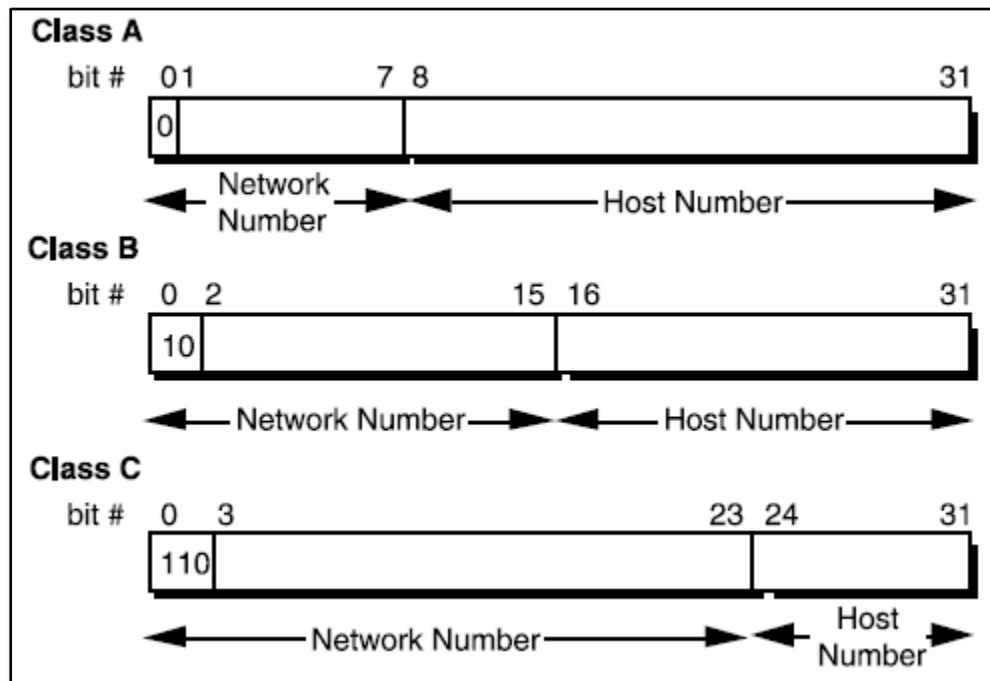


Ex-1019, 3.

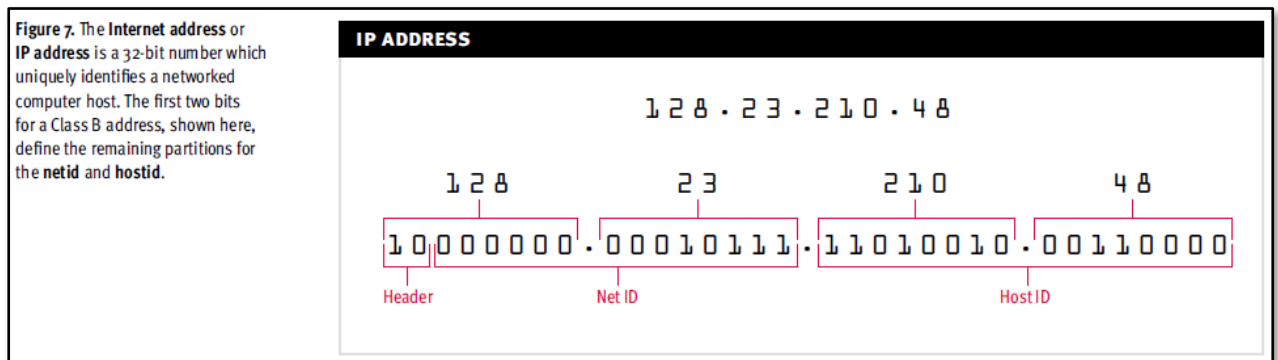
58. IP addressing applies a hierarchical approach to guarantee the uniqueness of an address. It does so by designating a Network Prefix such that, in a given network, “[a]ll the hosts share the same network prefix but must have a unique host number. Similarly, any two hosts on different networks must have different network prefixes but may have the same host number.” Ex-1019, 3. Thus, the host number, or “host id[,] allows the computer to be uniquely identified among all of the computers on the subnet.” Ex-1015, 169. One corollary is that the most significant bits of an address are used to identify the allocated block from where a specific address originated.

59. The IPv4 addressing scheme supports three classes: Class A, Class B, and Class C. These classes can support networks of varying sizes: “Each class fixes the boundary between the network prefix and the host number at a different point within the 32-bit address.” Ex-1019, 3. The number of possible networks is determined by 2^n , where n is the number of bits of the network prefix. Thus, a larger network prefix means that more bits are being used to define the network, which means that you can have more networks. The figure below depicts the address formats of the three classes. Moreover, each address “contains a self-encoding key that identifies the dividing point between the network prefix and the host number.” *Id.*, 4. For example, each Class A network has an 8-bit network prefix, with the highest order bit set to zero and a 7-bit network number, followed by a 24-bit host

number. *Id.*, 4. Similarly, each class B address has a 16-bit network prefix, with the two highest order bits set to 1-0 and a 14-bit network number; and each class C address has a 24-bit prefix, with the three highest order bits set to 1-1-0 and a 21-bit network number. *Id.*



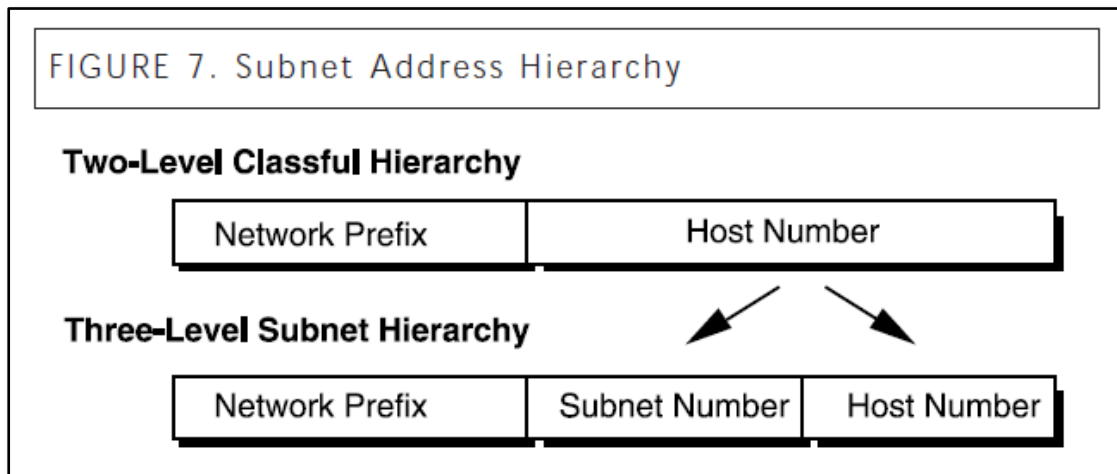
60. Notably, “[f]or readability, IP addresses are typically reduced to four decimal integers, separated by dots,” as shown in the figure. Ex-1014, 11-12. The figure below shows an IP address with its dotted-decimal representation. Thus, the dotted-decimal notation divides the 32-bit address “into four 8-bit fields” (an octet) “and specifies the value of each field independently as a decimal number.” Ex-1019, 5.



Ex-1014, 12.

61. An organization can also further subdivide its network among its different divisions in a decentralized way by dividing the network into sub-networks, or subnets. The subnetting, “or division, of a single Class A, B, or C network number into smaller pieces,” “add[s] another level of hierarchy to the IP addressing structure,” in order to allow organizations to “deploy additional subnets without obtaining a new network number from the Internet.” Ex-1019, 8. As depicted in the figure below, “the basic idea of subnetting [] is to divide the standard classful host number field into two parts—the subnet number and the host number on that subnet.”

Id.



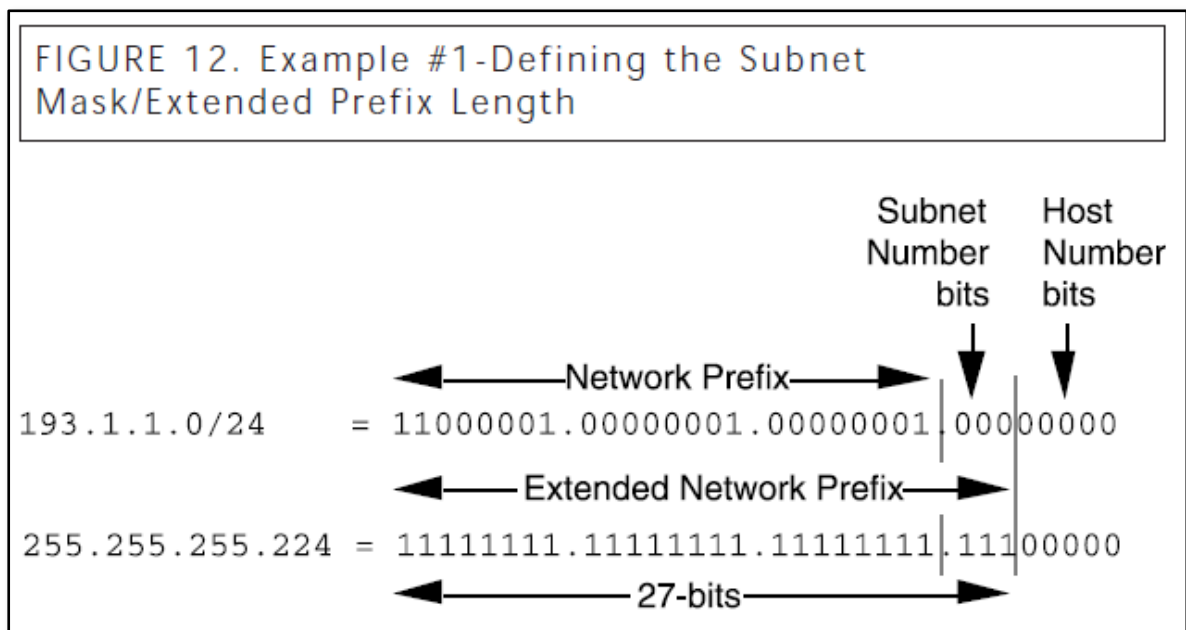
62. Although the 32-bit address is sometimes represented in dotted-decimal notation, it operates at the binary level—the key decisions and operations are undertaken at the binary-bit level. A “subnet mask” is a 32-bit number that “identifies the bits that represent the subnet id.” Ex-1015, 169. In the subnet mask of an IP address, bits set to “1” correspond to the network portion, while bits set to “0” correspond to the host portion. When subnetting, the first n bits of the IP address are fixed to define which subnet or network a particular address belongs to. For example, the subnet mask 255.255.255.0 (11111111 11111111 11111111 00000000 in binary) indicates that the first 24 bits of the address represent the network portion. *Id.* The network portion of an IP address is fixed for all devices within that network; the bits in the final octet are used to differentiate individual devices within that network.

63. The key decisions of the subnetting process are determined by bit-level

operations. For example, to create two subnets in the network 192.168.1.0/24, the network portion borrows one bit from the host portion. As such, the subnet mask changes from 11111111 11111111 11111111 00000000 (/24) to 11111111 11111111 11111111 10000000 (/25). The result is a pair of subnets with 7 bits for identifying the host portion. Thus, one subnet spans the range of addresses from 192.168.1.0 to 192.168.1.127, and the second subnet spans the addresses 192.168.1.128 to 192.168.1.255. In binary, the first subnet spans all binary numbers with the format 11000000 10101000 00000001 0xxxxxxx. The second subnet spans all binary numbers with the format 11000000 10101000 00000001 1xxxxxxx. The red portion of the IP address represents the new network portion of the subdivided subnets, and it is in binary form. *See generally* Ex-1019, 10-13.

64. To further illustrate, imagine that a company is assigned the network number 193.1.1.0/24 and needs to define six subnets, with the largest subnet able to support 25 hosts. Ex-1019, 11. “The first step in defining the subnet mask is to determine the number of bits required to define the six subnets. Since a network address can only be subnetted along binary boundaries, subnets must be created in blocks of powers of two.” *Id.*, 12. Because six is not a power of two, the company “must define a block of 8 (2^3) and have two unused subnets that can be reserved for future use.” *Id.* As such, “three bits are required to enumerate the eight subsets in the block.” *Id.* Here, the company is subnetting a “/24” network number, so it will need

three more bits, or a “/27”, to define the subnets. *Id.* It gets those three bits by “borrowing” them from the host number bits, effectively partitioning the “host number field” into a host-instance identifier and a subnet, or block, identifier. For ease of reading, a 27-bit network prefix “can be expressed in dotted-decimal notation as 255.255.255.224,” and is illustrated in the figure below.



Id., 12.

65. As the figure shows, the 27-bit network prefix leaves five bits to define host addresses on each subnet. *Id.* As such, “each subnetwork with a 27-bit prefix represents a contiguous block of 25 (32) individual IP addresses.” *Id.* Finally, the company defines “Subnet #N” by “plac[ing] the binary representation of N into the bits of the subnet number field.” *Id.* Thus, “to define Subnet #6,” the company “simply places the binary representation of 6 (110) into the 3 bits of the subnet

number field.” *Id.*

66. Consequently, IP addressing teaches the use of the most significant bits (“MSBs”) of the host portion to define the a subnet, which is a subdivision of the larger network; meanwhile, the remaining bits specify the individual host within that network. This is visually demonstrated in the figure below.

The eight subnet numbers for this example are listed in the following code sample. The underlined portion of each address identifies the extended network prefix, while the bold digits identify the 3 bits representing the subnet number field:

Base Net: 11000001.00000001.00000001.00000000 = 193.1.1.0/24
Subnet #0: 11000001.00000001.00000001.000 00000 = 193.1.1.0/27
Subnet #1: 11000001.00000001.00000001.001 00000 = 193.1.1.32/27
Subnet #2: 11000001.00000001.00000001.010 00000 = 193.1.1.64/27
Subnet #3: 11000001.00000001.00000001.011 00000 = 193.1.1.96/27
Subnet #4: 11000001.00000001.00000001.100 00000 = 193.1.1.128/27
Subnet #5: 11000001.00000001.00000001.101 00000 = 193.1.1.160/27
Subnet #6: 11000001.00000001.00000001.110 00000 = 193.1.1.192/27
Subnet #7: 11000001.00000001.00000001.111 00000 = 193.1.1.224/27

Ex-1019, 13.

67. Thus, for instance, the company can assign 2^5 , or 32, individual host

addresses in Subnet #2,¹ and by identifying the MSBs of the last octet, the company knows that the host belongs in Subnet #2. The figure below depicts host addresses for Subnet #2, as well as the 3-bit field for Subnet Number bits. The “bold digits identify the 5-bit host number field” within the subnet. *Id.*, 15. As shown, Subnet #2 is effectively a block of host addresses defined by the 3-bit Subnet Number field, *which are the most significant bits of the host number space.*

Subnet #2:	<u>11000001.00000001.00000001.010</u>	00000	= 193.1.1.64/27
Host #1:	<u>11000001.00000001.00000001.010</u>	00001	= 193.1.1.65/27
Host #2:	<u>11000001.00000001.00000001.010</u>	00010	= 193.1.1.66/27
Host #3:	<u>11000001.00000001.00000001.010</u>	00011	= 193.1.1.67/27
Host #4:	<u>11000001.00000001.00000001.010</u>	00100	= 193.1.1.68/27
Host #5:	<u>11000001.00000001.00000001.010</u>	00101	= 193.1.1.69/27
.			
.			
Host #15:	<u>11000001.00000001.00000001.010</u>	01111	= 193.1.1.79/27
Host #16:	<u>11000001.00000001.00000001.010</u>	10000	= 193.1.1.80/27
.			
.			
Host #27:	<u>11000001.00000001.00000001.010</u>	11011	= 193.1.1.91/27
Host #28:	<u>11000001.00000001.00000001.010</u>	11100	= 193.1.1.92/27
Host #29:	<u>11000001.00000001.00000001.010</u>	11101	= 193.1.1.93/27
Host #30:	<u>11000001.00000001.00000001.010</u>	11110	= 193.1.1.94/27

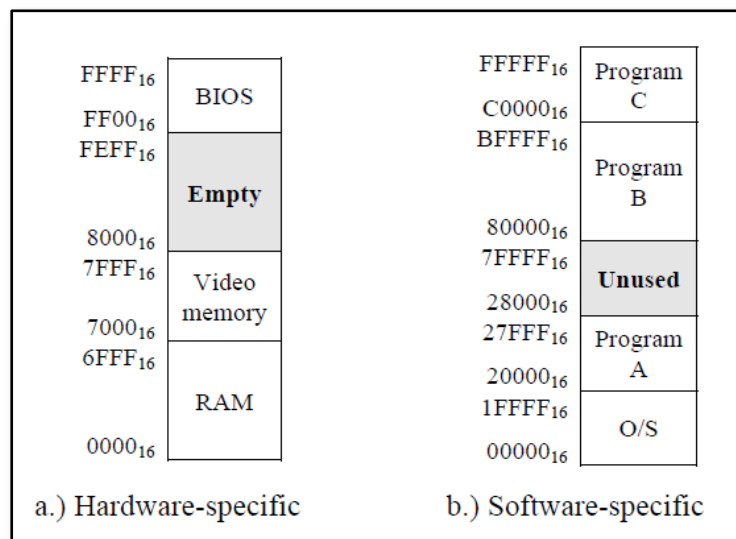
68. Other areas of computer science also apply these hierarchical principles

¹ Technically, it can assign only 30 host numbers two of them are reserved. The network 0.0.0.0 is reserved for future use and the network 127.0.0.0 is reserved for the “loopback” function. *Id.*, 4.

and MSB-based data structures to allocate binary-bit identifiers to a set of entities.

2. Computer Memory Allocation

69. The IP addressing scheme studied by the Auto-ID center employs the same methods as computer memory allocation methods—“A subnet within the entire IP network is equivalent to a memory device within the memory space of a processor.” Ex-1015, 254. Multiple memory devices and different types of memory may be present in a single system. Thus, it is necessary to show what addresses correspond to which memory devices and how the memory is being used. *Id.* To that end, a memory map is used to describe the use of memory. A memory map models the memory array where the rows correspond to the memory locations. *Id.*, 249. Within the full range of addresses are smaller partitions where the individual resources are present, as depicted in the figure below, which depicts a generic example of a memory map. *Id.*, 249-50.

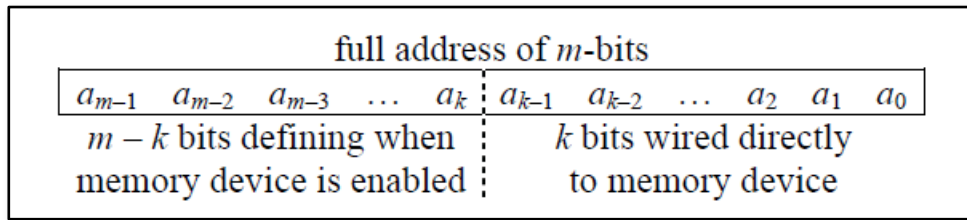


70. The memory map represents the address range of a processor, and this address range is referred to as the processor's memory space. *Id.* Its size is represented by the number of memory locations in the full range, i.e., 2^m , where m equals the number of address lines coming out of the processor to a section of memory. The size of the BIOS partition, for example, can be determined by subtracting the low address from the high address. The range of the BIOS starts at $FF00_{16} = 65,280_{10}$ and goes up to $FFFF_{16} = 65,535_{10}$. This means that the BIOS fits into $65,535 - 65,280 + 1 = 256$ memory locations. *Id.*, 249-50.

71. A POSITA would have known that each of the 256 locations is uniquely identified by virtue of the MSBs segmenting that part of memory, and that the LSBs with the MSBs create a unique address. This is because "there is an exact method to selecting the upper and lower addresses for each of the ranges in the memory map." *Id.*, 250. In the binary representation of the ranges in the figure above, the upper five bits of the two bounding addresses are identical while the remaining bits go from all zeros in the low address to all ones in the high address. *Id.* These "most significant bits are used to define which memory device is being selected." *Id.*

72. Accordingly, "[a]ll address decoding schemes have one thing in common: the bits of the full address are divided into two groups, one group that is used to identify the memory device and one group that identifies the memory location within the selected memory device...The division of the full address into

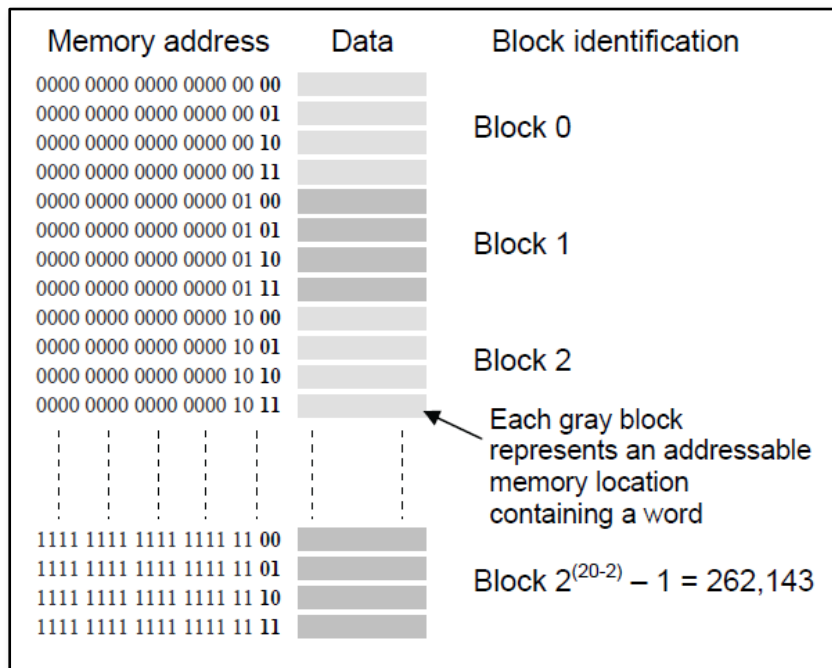
two groups is done by dividing the full address into a group of most significant bits and least significant bits,” as depicted in the figure below. *Id.*, 250-51. Doing so forces uniqueness on the combination of MSBs and LSBs despite repeating LSB values.



73. Thus, for example, given a 20-bit address such as 0011 1110 1001 0001, we can hide or mask the last bit of the memory address. The resulting address could refer to one of two possible locations, 0011 1110 1001 0001 or 0011 1110 1001 0000. Similarly, hiding the last two bits means that the system now has four potential locations it could be referring to, as the last two bits of the address could be 00, 01, 10, or 11. Therefore, the address could be referring to one of the following four memory locations: 0011 1110 1001 0000, 0011 1110 1001 0001, 0011 1110 1001 0010, 0011 1110 1001 0011.

74. As seen, the group of memory location share a common prefix and vary only in the last two bits. This group of memory locations is defined as a “block.” To organize memory into blocks, a memory address is divided into two logical sets of bits, one to represent the block number and one to identify the word’s position or

offset within the block. In the figure above, the memory address uses “*the most significant 18 bits to identify the block and the last two bits to identify a word’s position within the block.*” Ex-1015, 288. The figure below depicts the blocks of memory addresses.



75. This computer allocation method is familiar. A POSITA would have known that enabling a specific memory device within a processor’s memory space is equivalent to the method for resolving the subnet of an IP address. Ex-1015, 254 (“A subnet within the entire IP network is equivalent to a memory device within the memory space of a processor.”). A POSITA would have been aware of the similarities and the common organizing principles between the two methods. This includes the use of the most significant bits of an address (i.e., the memory location

identifier) to identify the block and the remaining lesser bits of significance to define a word's position in the block (i.e., the specific instance or memory location).

3. ISO 15693-3

76. In 2000, ISO published an Identification card RFID standard (ISO/IEC 15693-3). That standard specified uniquely identifying vicinity cards by Unique Identifier or "UID." Ex-1027, §4.1 (pg. 2-3). The UID space was partitioned into blocks. Those blocks were allocated to different manufacturers. The UID space was defined to have a fixed set of most significant bits for the manufacturer code, followed by serial number bits to be assigned by the manufacturer. This example further illustrates that it was well known in the RFID field to partition or sub-divide identifiers in a hierarchical manner.

MSB				LSB			
64	57	56	49	48			1
'E0'		IC Mfg code		IC manufacturer serial number			

Figure 1 — UID format

Id.

4. The Auto-ID Center Built on These Concepts and Methods to Develop the EPC standard

77. The examples above "show common approaches and general directions which" Brock and his team used in developing the EPC standard; and, importantly, they recognized that "it may even be possible to accommodate some of these

standards under the common framework of the EPC code.” Ex-1014, 12. By drawing on these examples, the Auto-ID Center developed the EPC standard to provide unique identification of physical objects.

78. For instance, “many codes, particularly license plate and telephone numbers, contain few partitions with few numbers” since they are specifically designed “for instant recognition and simple recollection.” *Id.*, 16. Meanwhile, “IP Addresses are intended for machine use, but include representations for human readability,” which, although not intended for memory, “can be written down and easily typed in manually.” *Id.* Similarly, “readability—as with the current UPC code and IP Address—will be necessary, but human recollection will not be required.” *Id.* As such, the EPC code was designed to “have a simple and consistent representation, which can easily be transcribed, dictated and keyed manually.” *Id.*

79. The EPC is typically represented as a 96-bit binary number divided into four fields and allocates a fixed number of bits to each field. *Id.*, 19-20. The “hierarchical” partitioning of the EPC fields maintains global uniqueness across multiple industries and manufacturers. The Header Field, typically allocated 8 bits, is the first section of the EPC and identifies the type of data being encoded. *Id.*, 19. The EPC manager field, allocated 28 bits, is responsible for identifying the organization or manufacturer that is managing the item’s serialization. *Id.*, 20. The Object Class field, allocated 24 bits, represents the category or type of the item being

tagged. For example, for a retailer, all products of the same type (e.g., bottles of a specific soda brand) would share the same Object Class. *Id.* The final portion of the EPC is the Serial Number field (36 bits), which uniquely identifies an individual item within a product category. *Id.* Every product instance receives a different serial number, ensuring that even if two items belong to the same manufacturer and product class, they can still be uniquely tracked. *Id.* .

80. More fundamentally, a POSITA would have understood that by organizing the EPC code hierarchically, each level narrows down the identification from a broad category to a specific item. This structure allows different manufacturers or entities to independently assign serial numbers within their allocated namespaces ***without the risk of duplication***. The uniqueness is enforced not just by the serial number itself but by its placement within the higher-level categories of the hierarchy. And this hierarchical approach to forcing uniqueness is baked into the EPC code, and a POSITA would have recognized that this method is agnostic to the representation of the serial numbers; in other words, these principles apply equally to decimal, binary, octal, hexadecimal, and other general-purpose encoding schemes. This is the case in the computer memory allocation (binary and hex), IP addressing (binary and decimal), vehicle license plates (alphanumeric, decimal), and the other art that the MIT team studied. *See* Ex-1014.

81. Moreover, a POSITA would have known that this method eliminates

the need for a central issuing authority because the combination of hierarchical fields inherently ensures global uniqueness. POSITAs recognized the drawbacks of relying on a central authority to assign serial numbers and understood that prefixing identifiers with distinguishing or categorical information—such as a manufacturer code or area code—achieved the same goal of uniqueness without requiring a central authority. Ex-1013, 238-40. Allocating ranges of unique identifiers, like serial numbers, to subordinate entities was a common method for maintaining uniqueness within a numbering scheme. *Id.* For example, in 2006, *RFID Essentials* explained that uniqueness “can be accomplished in several ways, but the simplest method is either to assign a prefix to each producer to put at the beginning of their identities or to assign blocks of identities to each producer from a central authority (which accomplishes the same thing).” Ex-1016, at 15.

82. Indeed, even *Dummies* explains that an EPC relies on a “numbering hierarchy” which can be exploited to solve the problem of using a central issuing authority to assign unique EPCs when multiple facilities across different locations are encoding tags for the same product. Ex-1013, 239. *Dummies* suggests “companies can use an intelligent hierarchy imposed on the serial number allocation to decentralize and make feasible the allocation of unique serial numbers across all manufacturing lines.” *Id.* “A simple example of a decentralized, hierarchical approach to allocating serial numbers” involves allocating a range of serial numbers

for each product to each manufacturing facility. *Id.*, 240. Then, the facility allocates “a range of numbers from those allocated to the facility [] to each line producing a particular product. In this way, *the serial number is effectively subdivided into a facility number, line number, and subserial number....*” *Id.* (emphasis added). *Dummies*, therefore, taught the “proven method” of partitioning a serial number field in order to further extend the hierarchical scheme of the EPC.

B. The Alleged Invention

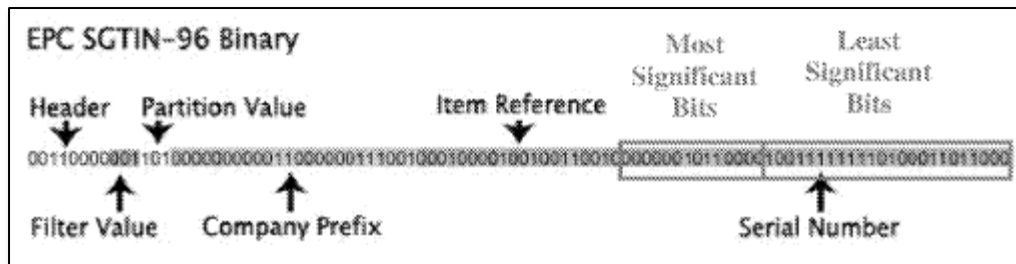
83. The '967 Patent is directed to radio frequency identification (“RFID”) transponders, or “tags,” and systems for “commissioning RFID transponders with unique object call instance numbers without requiring a realtime connection to a serialization database.” Ex-1001, Abstract, 5:65-6:1. RFID tags are used to identify and track objects by encoding a unique Electronic Product Code (“EPC”) associated with each object. The Serialized Global Trade Item Number (“SGTIN”) is an encoding scheme supported by EPC and consists of a GTIN identifying a particular brand and class of the item followed by a serial number uniquely identifying the tagged item within the brand and class. GS1, a central number-issuing authority, allocates SGTIN numbers to member companies in a hierarchical manner. *Id.*, 7:65-66. Each member company can further allocate numbers to lower levels as needed. *Id.*, 7:61-8:3.

84. The '967 patent discloses a method for allocating pre-authorized ranges

of serial numbers to lower level encoders by “subdivid[ing] the entire object class serial number space into sectors that are defined by a limited number of MSB’s (Most Significant Bits) of the serial number field.” *Id.*, 8:13-15. For example, an SGTIN-96 number has “a total of 38 bits used to define the entire serial number space which contains 238 unique numbers.” *Id.*, 20-23; *Id.*, 8:31-32. The “upper 14 bits could be designated as the most significant bits for a particular embodiment. In that case the object class serial number space would be comprised of 16,384 sectors.” *Id.*, at 8:25-26. Because 14 of the 38 bits are MSBs, “there must be 38 minus 14 bits of lesser significance, which equals 24 bits,” and those lower 24 bits “represent 16,777,216 unique serial number values.” *Id.*, 8:27-31. The ’967 Patent explains that this method eliminates the need for “external authorizations or queries” to a company’s central serialization number database, thus achieving a “quasi-autonomous RFID transponder encoding authority.”

85. During re-examination, the Patent Owner (“PO”) explained that the “first implication of this recitation is that the unique serial number space is specifically partitioned into two portions, a first portion that is the most significant bits and a second portion that are the bits of lesser significance.” Ex-1004, 173. The “second implication” is that “this intentional partitioning is based on the...limited number of most significant bits of the allocated block.” *Id.* For a manufacturer, the claimed invention “can ensure uniqueness across its many operations through, for

example, allocation of a block of serial numbers comprising a unique sequence of most significant bits to each product line or operation of the brand owner.” Ex-1030. These lower level encoders “corresponding to each product line or operation can then freely generate and assign serial numbers” comprising the assigned MSBs and the LSBs. *Id.* During re-examination, PO illustrated this serialization scheme in the figure below:



Ex-1004, 96.

86. According to PO: “By partitioning the leading bits of the serial number and assigning fixed MSBs for an object class to encoders from different assembly lines, facilities, or parties, the brand owner can enable each of the encoders to operate semi-autonomously to encode tags for items of a particular class without being connected to the same EPC issuing database, the uniqueness of the serial number is ensured within and across the ranges of numbers that are assigned to each encoder throughout the stream of commerce.” *Id.*

87. These features (“partitioning the leading bits of the serial number and assigning fixed MSBs”) are reflected in the claim limitations: [1E] wherein the

unique serial number space is encoded with one serial number instance from an allocated block of serial numbers, the allocated block being assigned a limited number of most significant bits, [1F] wherein the unique serial number space comprises the limited number of most significant bits uniquely corresponding to the limited number of most significant bits of the allocated block and of remaining bits of lesser significance that together comprise the one serial number instance.

C. The Effective Filing Date of the '967 Patent Is May 21, 2008

88. Based on my analysis, the earliest written description support for the claims is in the application dated May 21, 2008. The earlier applications do not include support for all the limitations of the claims, including for example, the limitation “the allocated block being assigned a limited number of most significant bits.” Ex-1024; Ex-1025, Ex-1026. Accordingly, my analysis here is based on a May 21, 2008 effective filing date for the all claims of the '967 patent.

D. The Prosecution History

89. The '967 Patent issued from Application 15/042,993, which was filed on February 12, 2016, and claimed priority to a series of continuation and continuation-in-part applications that trace back to March 21, 2008. No claims were rejected in view of the prior art, but the Examiner issued a double-patenting rejection, which was overcome by a terminal disclaimer. Ex-1003, 146-49. Ultimately, the '967 Patent issued on October 24, 2017. *Id.*, 301.

E. Ex Parte Reexamination History

90. On November 29, 2017, Avery Dennison filed a reexamination request identifying four references that were not considered during the original prosecution of the '967 patent:

1. U.S. 2007/0040684 A1 (“McAllister”)
2. U.S. Pat. No. 7,827,200 (“IBM200”)
3. U.S. Pat. No. 8,373,545 (“Traub”)
4. EPCglobal Tag Data Standards Vers. 1.3 (“TDS”)

Ex-1004, 4. The Office ordered reexamination on December 20, 2017, under Reexamination Control No. 90/014,052. *See* Ex-1004.

91. On May 14, 2018, the Examiner rejected every claim under § 103 based on McAllister in view of (1) IBM200, or IBM200 and TDS, and (2) Traub, or Traub and TDS. *Id.*, 126.

92. In response, PO amended the independent claims as follows: “wherein the unique serial number space comprises the limited number of most significant bits uniquely corresponding to the limited number of most significant bits of the allocated block and of remaining bits of lesser significance that together comprise the one serial number instance.” *Id.*, 164, 166.

93. PO argued that (1) the method of encoding serial numbers, wherein the allocated block of serial numbers is defined by a limited number of MSBs, is a

“structural limitation of the claims;” and (2) the prior art references only suggest the selection of decimal ranges for serial numbers, without disclosing or suggesting the allocation of blocks based on MSBs. *See, e.g., id.*, 175–76. PO contended that the claimed RFID transponder encoding is not merely a process but a structural limitation because the serial number space is partitioned into two distinct portions: MSBs corresponding to the allocated block, and lesser significant bits. This intentional partitioning is designed to reflect the encoding structure, where the MSBs from the allocated block are precisely what is encoded on the RFID transponder. *Id.*

94. The Examiner argued that converting decimal ranges to binary would have been obvious to a POSITA. PO countered that the cited prior art, particularly the IBM200 and Traub patents, fail to disclose this MSB-based approach. *Id.* According to PO, IBM200 teaches serial number allocation through ranges of decimal values—such as odd/even serial numbers or specific ranges for different divisions—but does not suggest a binary system based on MSBs. *Id.* PO asserted that simply selecting decimal ranges that may convert to specific MSBs in binary is fundamentally different from intentionally allocating blocks based on MSBs. *Id.*, 100. Without the intentional partitioning into MSBs and lesser significant bits, as taught by the ’967 patent, there is no discernible benefit in allocating based on MSBs. *Id.*

95. Further, PO asserted that such conversions in IBM200 do not achieve

the same structural encoding specified in the claims, which requires the unique selection of serial numbers based on significant bits rather than arbitrary decimal ranges. *Id.*, 99. PO also argued that, like IBM200, Traub discloses a centralized EPC server that allocates code blocks according to business rules but relies on decimal allocations; as such, PO concluded that Traub lacks the specific partitioning of serial numbers into MSBs. PO contended that these prior art methods, which are based on decimal allocation, fundamentally differ from the claimed MSB-based encoding scheme. *Id.*, 179.

96. Following PO's arguments and amendments, the Examiner issued a Notice of Intent to Issue an Ex Parte Reexamination Certificate on July 5, 2018. The Reexamination Certificate, issued on July 30, 2018, incorporated PO's amendments to address the prior rejection dated May 14, 2018. *See* Ex-1002.

F. Litigation History

97. I understand that the '967 Patent was previously asserted against another company named Avery Dennison in *ADASA Inc. v. Avery Dennison Corp.*, Case No. 6:17-cv-01685-MK (D. Or.) ("*Avery Dennison*"). I understand that in *Avery Dennison*, a jury found that the '967 patent was (1) not anticipated by Kuhno, and (2) not anticipated by or obvious over *Dummies*, but I also understand that the jury did not consider obviousness in view of Kuhno, either alone or in combination other references.

G. Level of Ordinary Skill in the Art

98. I understand that in *Avery Dennison*, ADASA argued for the following level of ordinary skill in the art:

[A]t least a bachelor's degree in electrical engineering, computer science, or computer engineering, or a related discipline, and at least five years of experience which would consist of, A, designing, developing, evaluating and/or using RFID or other AIDC [automatic identification and data capture] technologies in inventory management, unique object identification, and/or security; or B, studying and conducting research related to RFID technologies and using RFID and other AIDC technologies in inventory management, unique object identification, and/or security.

Ex-1030, 707:3-13.

99. For purposes of this proceeding, I adopt ADASA's description of the level of ordinary skill in the art.

VI. Claim Construction

A. How the Challenged Claims Are To Be Construed

100. I understand that the Board gives claims their ordinary and customary meaning, or the meaning that the term would have to a person of ordinary skill in the art at the time of the invention.

For purposes of this proceeding, with the exception of the terms “an allocated block of serial numbers” and “uniquely corresponding to” discussed in greater detail

Petition for *Inter Partes* Review of
U.S. Patent No. 9,798,967

below, I interpret the language of the '967 Patent claims according to ADASA's constructions, which were adopted in *Avery Dennison* (Ex-1017) (see below table).

Term	Construction
object class information space	data field within the memory of the RFID integrated circuit chip for information identifying the class of an object, such as a company prefix, item reference code, partition value, and/or filter value
unique serial number space	data field within the memory of the RFID integrated circuit for information identifying a unique serial number
being assigned a limited number of most significant bits	includes a limited, predefined sequence of higher order bits at the leading end
remaining bits of lesser significance	the remaining lower order bits at the trailing end
is encoded with one serial number instance	has stored within it one serial number instance

1. “An Allocated Block of Serial Numbers”

101. This term does not have a plain and ordinary meaning, and the intrinsic evidence shows that a POSITA would have understood the term “an allocated block of serial numbers” to mean “a pre-authorized range of *binary* serial numbers.”

102. The '967 patent's specification explains that “[a] preferred method of providing pre-authorized blocks of object class serial numbers is to subdivide the entire object class serial number space into sectors that are defined by a limited number of MSB's (Most Significant Bits) of the serial number field.” Ex-1001, 8:11-15. “Once a sector is allocated to a lower level within an authority hierarchy, it is referred to as a block.” *Id.* at 8:31-32. During reexamination, PO emphasized that “[t]he partitioned serial number approach in the '967 Patent manages assignment of the serial number at the binary bit level, rather than sequential allocation of decimal or hexadecimal number strings from a database.” Ex-1004, 96. “The entire point of the hardware-based solution provided by the '967 patent is that the assignment of the MSBs eliminates the need to be concerned with the decimal versions of the serial numbers. [The prior art fails to] disclose or suggest such a binary assignment method.” *Id.*, 103.

103. Therefore, according to the intrinsic evidence, it is my opinion that the term “an allocated block of serial numbers” should be construed to mean “a pre-authorized range of binary serial numbers.”

2. “Uniquely corresponding to” Limitation

104. In *Avery Dennison*, PO asserted the “uniquely corresponding” term should be construed according to its plain and ordinary meaning or “uniquely and directly correlated.” Ex-1033, 18. Avery proposed: “Each and every serial number in the allocated block and only serial numbers in the allocated block contains the most significant bits.” *Id.* Based on the parties’ apparent agreement that their constructions meant the same thing, the Oregon court construed the term “as having its plain and ordinary meaning.” Ex-1034, 9 (“Defendant Avery acknowledges that the parties’ differing constructions may mean the same thing.”); *see also* Ex-1033, 19 (PO states: “... Defendant’s construction that only the serial numbers in the allocated block contain the same most significant bits, and no other serial numbers, is not necessarily incorrect.”).

105. My understanding is supported by the Federal Circuit, which recognized that the “uniquely corresponding” term had been construed by the district court according to its plain and ordinary meaning. Ex-1032, 908-09. The Federal Circuit observed that this term connotes a one to one correspondence between the MSBs defining a block of serial numbers and the range of serial numbers within that block:

In other words, for any set of MSBs there is exactly one corresponding allocated block, and for each allocated block there is exactly one set of MSBs. In essence, the claimed MSBs function as an additional data field within the serial number space that uniquely identifies the allocated block from which it came.

...

Because the predefined sequence of MSBs in a given serial number uniquely corresponds to an allocated block, and vice versa, serial numbers drawn from different blocks are guaranteed to be unique.

Id. at 909.

106. Accordingly, for this declaration, I am using the plain and ordinary meaning consistent with the articulation of the plain and ordinary meaning in *Avery Dennison*.

VII. Overview of the prior art

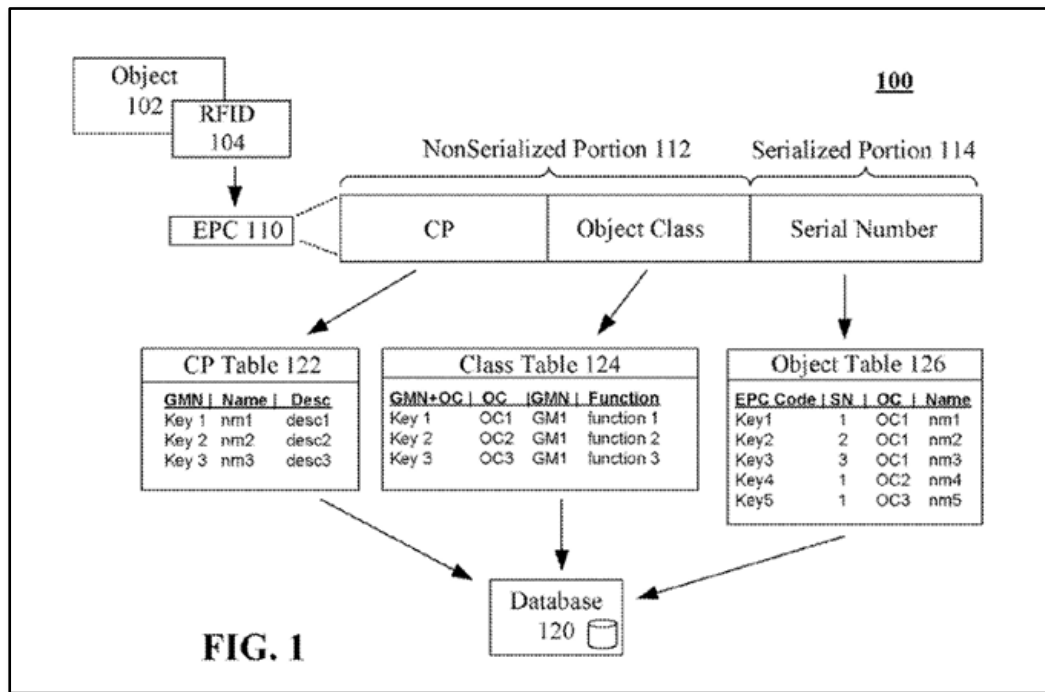
A. IBM200 (Ex-1007)

107. U.S. Patent No. 7,827,200 (“IBM200”) (Ex. 1007) was filed as U.S. Pat. App. 11/457,704 on July 14, 2006. Accordingly, IBM200 qualifies as a prior art printed publication. IBM200 is directed to a method and system for efficiently managing the serialization of EPCs used in RFID tags. IBM200 addresses the challenge of generating and managing unique serial numbers for EPCs, recognizing

that “various retailers including WAL-MART, TARGET, and the US Department of Defense (DoD) are requiring suppliers to tag material with an EPC identifier encoded on a n RFID tag.” Ex-1007, 1:21-25. This, in turn, “requires that EPC identifiers includes a serial number that is unique for each tagged object/location/entity or collection thereof.” *Id.*, 1:33-36.

108. IBM200 is analogous art at least because both patents are in the field of RFID-based product identification and tracking systems. They both focus on managing serial numbers for items in a supply chain or inventory management context; both patents describe systems for generating and managing unique serial numbers.

109. Figure 1 depicts a system for managing serialization of RFID EPCs, where each RFID tag is associated with an object and encoded with a unique EPC identifier. The identifier is divided into a nonserialized portion, which includes manufacturer and object classification information, and a serialized portion, which provides item-specific uniqueness. This structure allows the nonserialized part to identify general attributes, while the serialized portion ensures that each tagged object has a unique identifier. *Id.*, Fig. 1, 4:30-43.



110. As illustrated in Fig. 1, the EPC 110 has a pure identity that includes a nonserialized portion 112 and a serialized portion 114. A pure identity is “an abstract name or number used to identity object 102 that consists of the information required to uniquely identify a specific object 102 and nothing more.” *Id.*, 3:61-64. The nonserialized portion 112 can specify a manufacturer and an object classification of object 102. Each company or manufacturer can be assigned a unique Company Prefix (CP) or General Manager Number (GMN) by EPCglobal. The company can then use the assigned CP to generate new EPC numbers. The other component of the nonserialized portion, the object classification, is unique for a given company. *Id.*, 3:65-4:5.

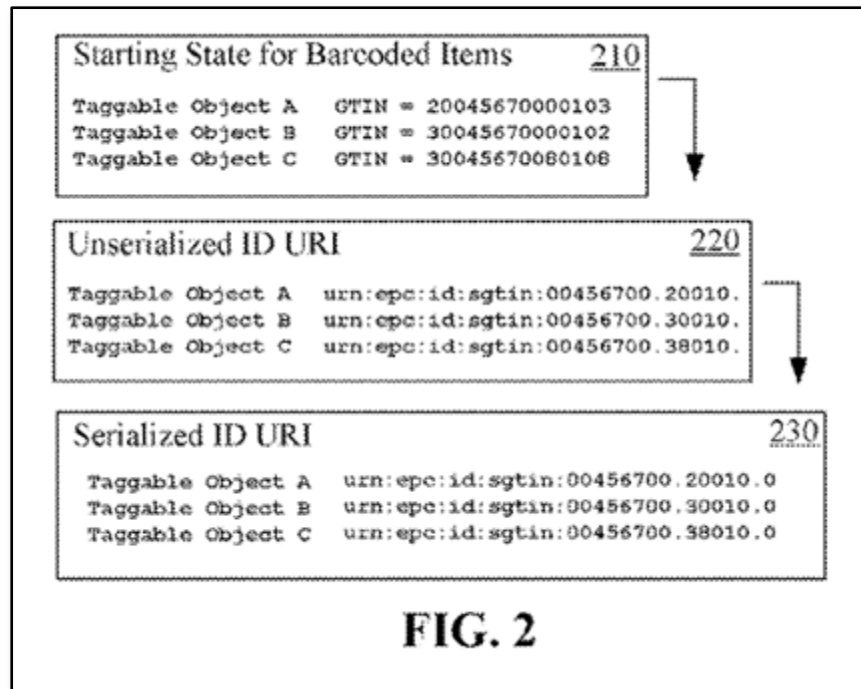
111. To manage the EPC identifiers, IBM200 teaches a relational database

structure with tables normalized to the third normal form. These tables, depicted in Fig. 1, include the Company Prefix (CP) table, the Class table, and the Object table. In this structure, a one-to-many relationship exists between the CP table and the Class table, as well as between the Class table and the Object table. This means that each entry in the CP table can be associated with multiple entries in the Class table, and each entry in the Class table can be linked to multiple entries in the Object table. This relational design facilitates efficient management and retrieval of EPC data. *Id.*, 4:13-20.

112. The Object table 126 stores the serialized values, which are created by “[c]ombining the serialized portion 114 with the nonserialized portion 112.” *Id.*, 4:9-12. The combination leads to “unique identifier for object 102;” in other words, the “serialized portion 114 of EPC 110 can be specifically associated with object 102.” *Id.*, 4:8-12. The relational database maps classification values to serial number generation functions, which allows each object classification to follow distinct serialization rules, such as sequentially ascending or descending serial numbers, or other mathematical patterns. Each classification maintains an independent serial number sequence to prevent duplication. In practice, this could mean that different divisions or product categories have designated serial number ranges or use different mathematical functions for unique number generation.

113. Fig. 2 illustrates how EPCs are generated for three taggable objects:

Objects A, B, and C. These objects are initially barcoded using a Universal Product Code (“UPC”), which use GTINs, as seen in State 210. The TDS specifies how to convert the GTINs into identity Uniform Resource Identifiers (“URIs”) for an EPC. An identity URI is one expression or representation for a pure identity. For any EPC type, “each EPC 110 has a pure identity that includes a nonserialized portion 112 and a serialized portion 114.” *Id.*, 3:56-58.



Id., Fig. 2.

114. The unserialized ID URIs in State 220 are then combined with a serialized portion to finally arrive to each object’s serialized identity URIs. The figure below depicts the form of the serialized ID URI, where “nnnnn” represents a serial number to be generated by the entity.

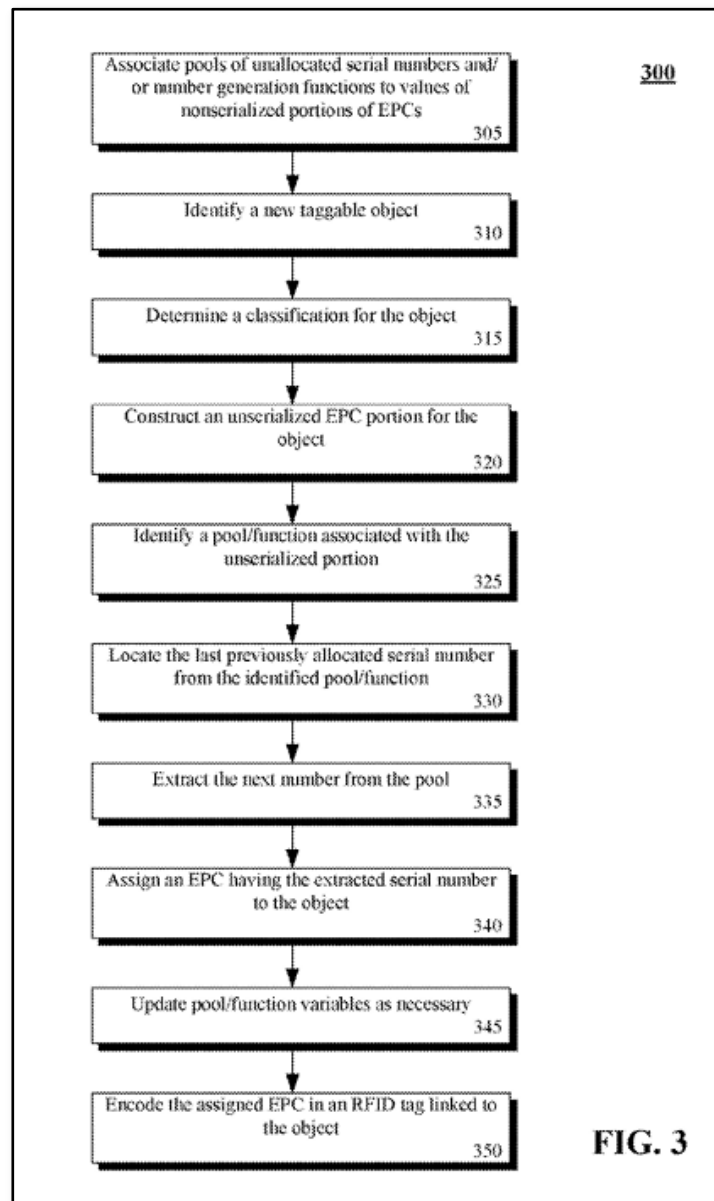
Object A	urn:epc:id:sgtin:00456700.20010.nnnnn
Object B	urn:epc:id:sgtin:00456700.30010.nnnnn
Object C	urn:epc:id:sgtin:00456700.38010.nnnnn

Id., 5:40-45.

115. Encoding the serial number to physical memory means that the “size of a serial number is limited because of physical memory constraints of an RFID tag.” *Id.*, 5:67-6:1. As such, to avoid exhausting all available numbers quickly, IBM200 “teaches that a serialized portion of EPCs are to be managed and allocated on a classification-by-classification basis, where each classification is based upon the unserialized portion of the EPC.” *Id.*, 6:11-14. Therefore, “serialized portions of EPCs can be automatically assigned based upon a taggable classification defined by a particular EPC type. The nonserialized portion of the EPC remains constant for a given taggable class.” *Id.*, 6:24-27.

116. Figure 3 presents a method for serializing EPCs. The method begins at step 305, where the system establishes multiple pools of numbers or that correlate with the nonserialized portions of EPCs. *Id.*, 6:39-42. These pools are structured so that each classification has a unique serial number space. Then, after identifying a new taggable object that needs an EPC and determining a classification for the new object, step 320 constructs the unserialized portion of the EPC based on this

classification data. *Id.*, 6:44-48. This portion includes information such as the manufacturer's identifier and an object classification code.



117. In step 325, the system identifies the relevant pool of numbers or serial number generation function tied to the object's classification. This pool or function is then queried to locate the last assigned serial number (step 330) to determine the

next available number for assignment. *Id.*, 6:47. In step 335, the next unallocated serial number from the identified pool is retrieved. This serial number is then combined with the previously created nonserialized portion, resulting in a complete and unique EPC. *Id.*, 6:49-51. The newly formed EPC is then assigned to the object in step 340, completing the identifier assignment. *Id.*, 51. Finally, in step 350, the system encodes the assigned EPC into an RFID tag, which is then linked to the physical object. *Id.*, 6:52-58.

118. In sum, the “nonserialized portion of the EPC can specify a manufacturer and an object classification of” the object, or “taggable thing.” The serialized portion is “specifically associated with” the object. Together, the nonserialized and serialized portions “result in a unique identifier” for the object. The nonserialized portion is associated with a “pool of unallocated numbers” from which the serialized numbers are drawn. The number extracted from that pool is used as the serialized portion of the EPC. As such, serialized portions of EPCs are assigned based upon the object’s classification, while the nonserialized portion of the EPC remains constant for a given taggable class.

B. Kuhno (Ex-1008)

119. Kuhno (Ex. 1008) was filed on July 28, 2006. Accordingly, Kuhno qualifies as a prior art printed publication. Kuhno discloses a mobile RFID tagging system in the form of a workstation mounted on a cart that includes a general-

purpose computer, RFID printer, and barcode scanner. Ex-1008, 4:53-57; *Id.*, 5:6-29. The cart houses a battery system that allows it to be mobile across different areas in a warehouse or facility. The RFID printer encodes RFID tags that are embedded within labels and can print barcodes containing specific information related to the cartons or pallets on which they are placed. *Id.*, 5:4-8. Crucially, Kuhno teaches:

that while an RFID printer may print alphanumeric and/or barcode characters on a label carrying the encoded RFID tag, the phrase “RFID printer” may encompass a device that *encodes RFID tags without printing either or both of alphanumeric and barcode characters on the tag* and that RFID tags dispensed from an RFID printer need not be disposed on printable label stock. It should also be understood that the terms ‘RFID tag’ and ‘*RFID label*’ refer to a *programmable/encodable radio frequency device and antenna attached to a substrate* so that the device and antenna can be secured to a desired object such as a carton. “Label” refers generally to a substrate capable of attachment to a carton or pallet and upon which information regarding the carton or pallet is disposed through a barcode, an RFID tag, or both. In the presently described embodiment, RFID printer 20 uses printable label stock that contains an embedded RFID tag. RFID printer 20 formats and prints a barcode containing information specific to the cartons and pallets on which the label will be placed and, *at the same time, encodes the embedded RFID tag with the same and, optionally, additional information.* *Id.*, 5:6-29 (emphasis added).

120. The mobile RFID tagging system assigns and manages serial numbers

for cartons and pallets in a manner that ensures unique identification. The system allows configuration of several parameters, including (1) Carton Serial Length, which defines the total number of digits in the carton serial number; (2) Carton Sequence Length, which specifies the number of digits in the carton number portion.

Fig. 19 depicts a screen display of a software application with fields for specifying the Carton Serial Length and Carton Sequence Length. *Id.*, 14:20-35.

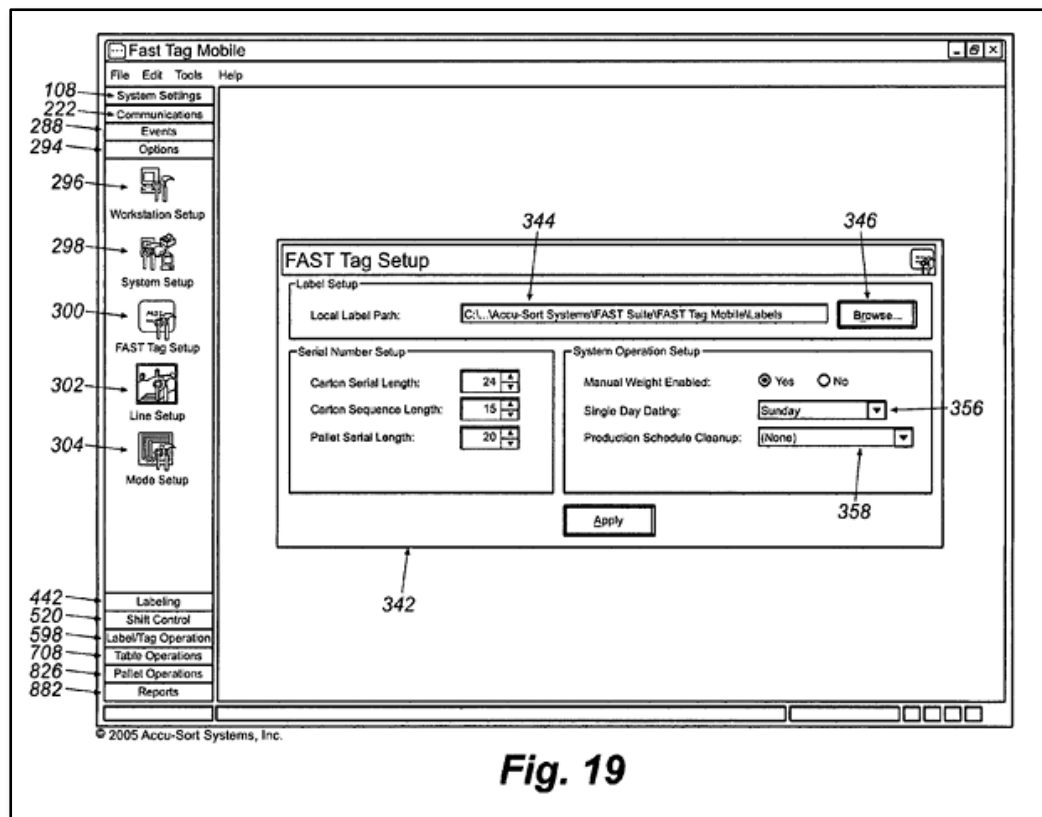


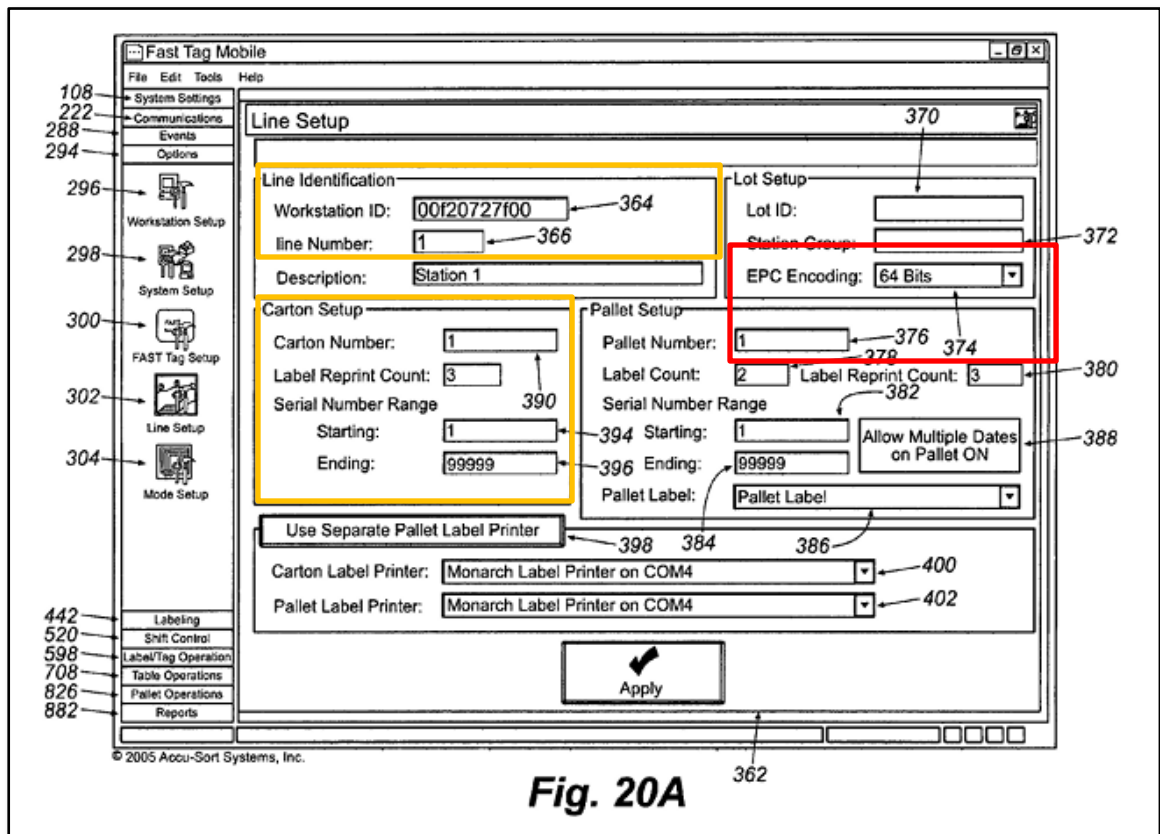
Fig. 19

121. Carton serial numbers are generated by combining a predetermined number, the production line number, and an end portion within a user-defined range.

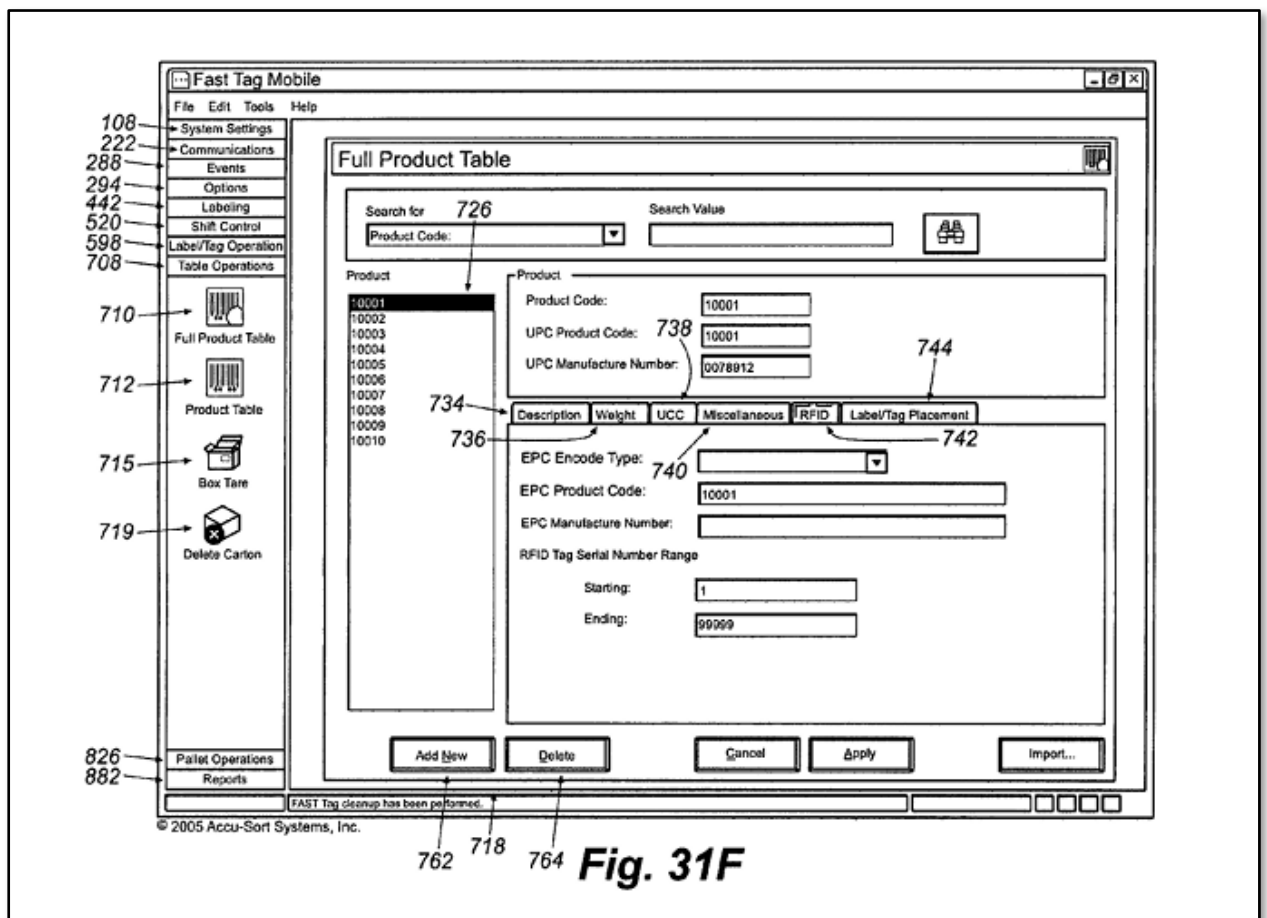
Id. The system increments this end portion for each processed carton. Once the

sequence reaches the maximum end value, it resets to the starting number while maintaining a local database that stores serial numbers it has already generated to avoiding duplication within a production line. Pallet serial numbers are generated similarly. Fig. 20A depicts a screen display from the FAST TAG Mobile software that shows fields for specifying the line number (366) and serial number range (394, 396) for both cartons and pallets. The software also includes a drop-down box 374 is “used to select the encoding type of the RFID tags used in the system: 64 Bits or 96 Bits. This configuration indicates to the application what types of tags are loaded in the RFID printer and how the information should be encoded to properly fit in the tag’s memory space.” *Id.*, 15:65-16:3.

122. The subdivided carton serial number is encoded in binary. *Id.*, 15:65-67 (“A drop-down box 374 is used to select the encoding type of the RFID tags used in the system: 64 Bits or 96 Bits.”); *see also* Figure 20A below.



123. Fig. 31F shows the RFID tab 742, which “provides fields for EPC encode type (SGTIN, SSCC, SGLN and GRIA), EPC product code, EPC manufacturer number, and RFID tag serial number range.” *Id.*, 23:41-44.

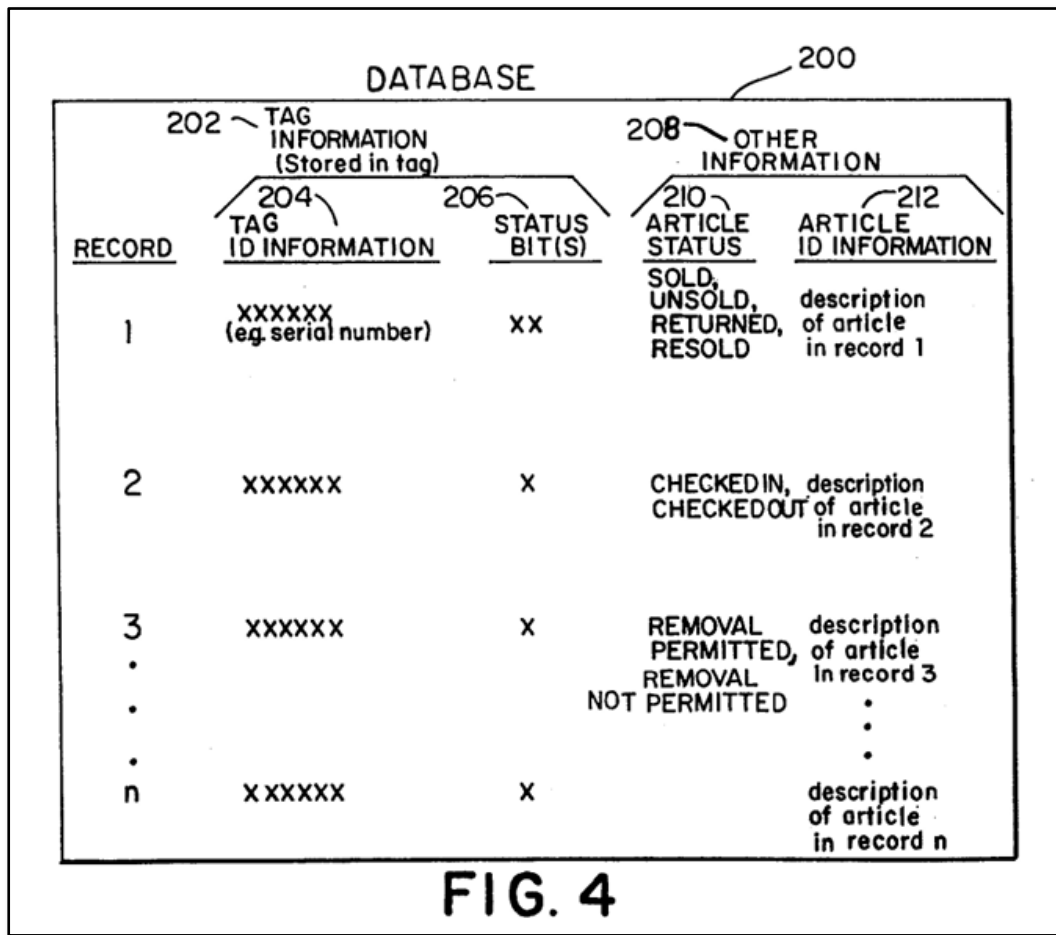


124. Kuhno is analogous art to the '967 Patent at least because both patents are in the field of RFID-based product identification and tracking systems. They both focus on managing serial numbers for items in a supply chain or inventory management context; both patents describe systems for generating and managing unique serial numbers. Moreover, both patents use composite serial numbers that include multiple components, such as Kuhno's combination of elements like Julian calendar date, production line number, and carton number.

C. Bowers (Ex-1009)

125. U.S. Patent No. 6,025,780 (“Bowers”) was filed on July 25, 1997 as U.S. Pat. App. No. 08/900,266. Accordingly, Bowers qualifies as a prior art printed publication. Bowers is directed to inventory control using RFID tags. Bowers teaches a “predefined set of tags,” where “each tag is associated with, and attached to, an article....” Ex-1009, 2:54-56. In other words, a predefined set of tags are applied to a predefined set of articles.

126. In one embodiment, each RFID tag is assigned “unique tag information” and an associated database is then established for the set of tags, depicted in Fig. 4. As shown, “each numbered record in the database 200 includes a field for tag information 202 comprising *tag identification information* 204 and variable information in the form of *one or more status bits* 206.” Ex-1009, 6:31-34. The database may also contain additional fields for other information, such as article status 210 (*see* Fig. 4). Notably, it is the tag information 202 that is stored in the tag. *Id.*, 6:37-38. However, “optionally, some or all of the other information 208 is also stored in the tag 10.” *Id.*

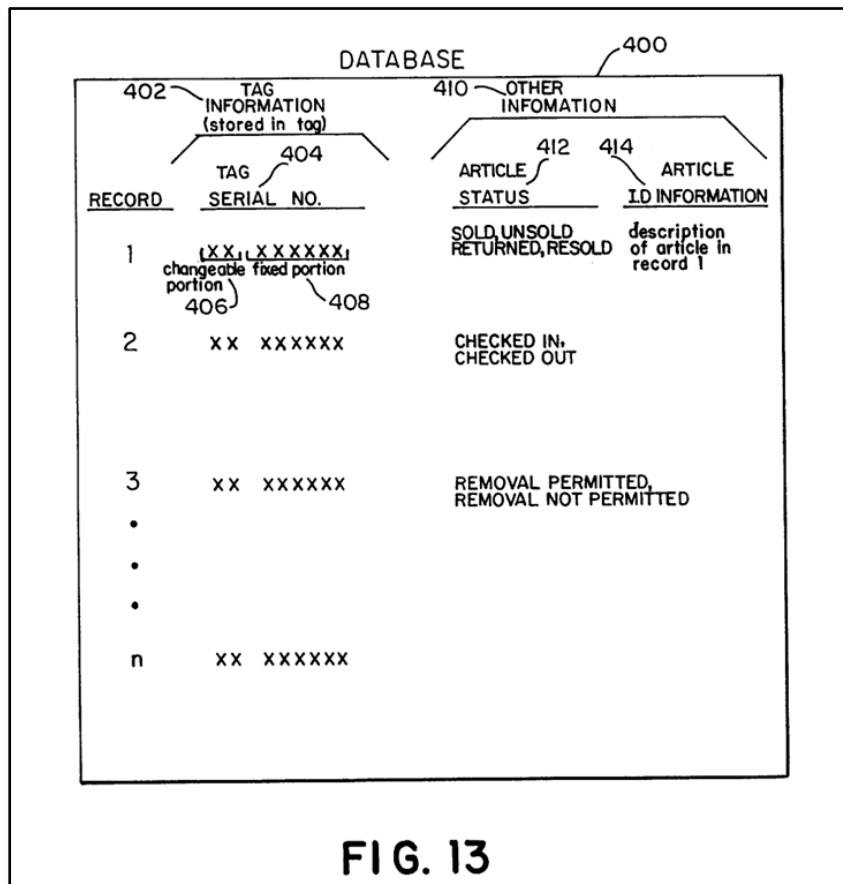


127. The status bits serve as dynamic indicators of the current state of the article to which the tag is attached. In a retail environment, for instance, the tag ID could have a two-bit status code where “00” indicates a sold item, “01” an unsold item, “10” a returned item, and “11” a resold item. This allows for up to four distinct states, though the system could be expanded to include more bits for additional status options if required. Because the status bits are dynamic, users can electronically change or reprogram the status bits as the article moves through various stages of the retail process. For example, when an item is purchased, its status can be quickly

updated from “unsold” to “sold” at the point of sale. *Id.*, 7:20-30.

128. The second embodiment simplifies the tag’s data structure by integrating item status information into the serial number itself. Unlike the first embodiment, “there are no separate status bits.” Ex-1009, 10:39-40. Instead, a portion of the RFID tag’s serial number, the “changeable portion,” is “used to perform some of the functions performed by the status bits of the first embodiment.” *Id.*, 40-41.

129. Fig. 13 depicts a sample database for the second embodiment. In this configuration, the tag information 402 comprises only “a tag serial number 404.” *Id.*, 10:58. This “serial number 404 includes a variable or changeable portion 406 of one or more bits and a fixed portion 408 of a plurality of bits.” *Id.*, 10:59-61. Like the first embodiment, the “tag serial number 404 is stored in the tag 10,” but “optionally, some or all of the other information 408 is also stored in the tag 10.” *Id.*, 10:64-66.



130. In the context of electronic security, the changeable portion can consist of one or more digits at the beginning of the serial number, which are electronically modified based on the item's current state. For example, an initial serial number of 01234567 might be altered to 00234567 upon sale of the item, or to 10234567 if the item is returned. This alteration allows the serial number to reflect the item's status directly without the need for separate status bits.

131. Bowers is analogous art to the '967 Patent at least because both inventions are in the same field of managing serial numbers stored in RFID tag memory and associated with physical items. Additionally, both integrate additional

information in the serial number space so as to have similar data structures. Moreover, both patents integrate additional information within the serial number space, which results in comparable data structures.

D. Tilson (Ex-1010)

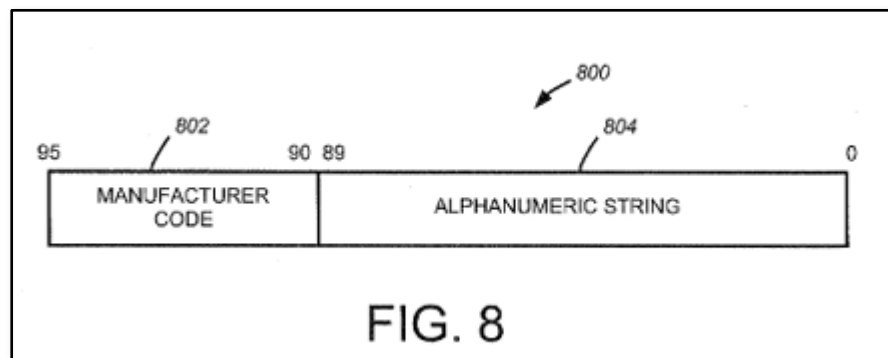
132. Tilson issued on August 17, 2010 from an application filed on September 14, 2007 as U.S. Pat. App. No. 11/855,766. Accordingly, TDS qualifies as a prior art printed publication.

133. Tilson is directed to methods for applying identification information to RFID tags, and its use with cast concrete components. Ex-1010, Abstract. Tilson teaches an “RFID assembly 100 [that] stores a unique serial number that may be read by an RFID reader, entered into a computer database and associated with the particular concrete component being manufactured.” Ex-1010, 4:23-26.

134. Fig. 8 depicts a data structure 800 representing the information stored on the non-volatile memory unity of the RFID tag 104. *Id.*, 7:61-66. The information “stored on the RFID tag 104 is a serial number that uniquely identifies the concrete component.” *Id.* As such, the “data structure 800 is organized to store a serial number. In the preferred embodiment, the serial number has a manufacturer code portion 802 and an alphabetic, numeric or alphanumeric portion 804, which together form the unique serial number for the respective concrete component. For example, bits 90-95 of a 96-bit data structure may store the manufacturer code, and bits 0-89

may store the alphanumeric portion.” *Id.*, 7:65-8:5.

135. Accordingly, Tilson teaches that the alphanumeric string is composed of human-readable symbols displayed on a screen by a software and represented and stored in RFID tag as binary values. A POSITA understood that these two forms—the human-readable string and the binary representation—are equivalent. Put differently, *the alphanumeric string and its binary representation are not two different values or information; rather, they are equivalent and represent the same information.* Moreover, a POSITA would have understood that the information saved in the RFID tag is actually in binary form, and that it is so in all cases.



136. Moreover, Tilson teaches that “each concrete component manufacturing making using an RFID assembly 100 of the present invention is assigned its own manufacturer’s code. This manufacturer’s code, moreover, is used in all RFID assemblies sent to and used by this particular manufacturer.” *Id.*, 8:6-8.

137. However, importantly, Tilson teaches that people “skilled in the art will understand that the data structure 800 may be modified to carry additional

information. For example, it may be configured to carry a name or identifier of manufacturer of the concrete component, a name or identifier of the entity that supplied the concrete used to make the component, an identifier of the concrete batch used to make the concrete component, a name or identifier of the entity that supplied the reinforcement element, if any, used in the precast concrete component, a name of the project or structure for which the concrete component is to be used, etc.” *Id.*, 8:32-41.

138. Tilson is analogous art to the '967 Patent at least because both inventions are in the same field of managing serial numbers stored in RFID tag memory and associated with physical items. Additionally, both integrate additional information in the serial number space so as to have similar data structures. Moreover, both patents integrate additional information within the serial number space, which results in comparable data structures.

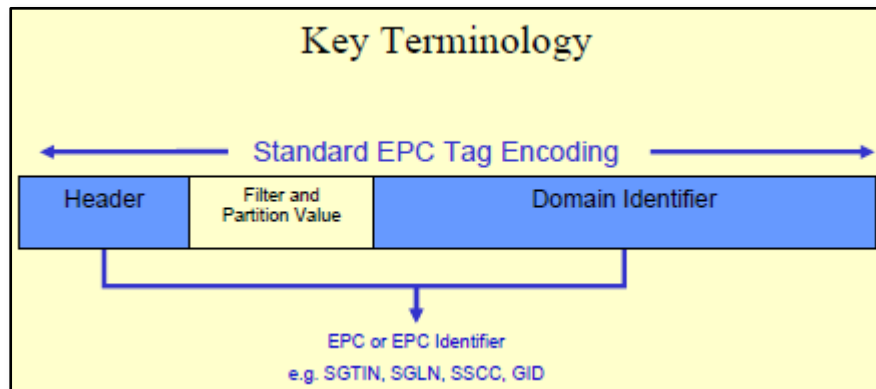
E. EPC Tag Data Standards Version 1.3 (Ex-1011)

139. EPCglobal Tag Data Standards Version 1.3 (“TDS”) was ratified on March 8, 2006 and, the Wayback machine indicates that version of TDS was available at least by January 4, 2007. Accordingly, TDS qualifies as a prior art printed publication.

140. TDS defines the EPC and provides guidelines for its implementation in RFID tags. The standard also specifies various EPC data formats, including SGTIN-

96, which the '967 Patent acknowledges as prior art: “The object class serial number space is defined by the number of serial number bits that are used in a specific standard, such as a particular EPCglobal key type, for example an SGTIN-96 and is defined in a corresponding specification such as the GS1 EPCglobal EPC Tag Data Standard.” Ex-1001, 8:16-21.

141. TDS teaches that the EPC “represents a family of coding schemes,” and the “various coding schemes are referred to as Domain Identifiers, to indicate that they provide object identification within certain domains such as a part particular industry or group of industries.” Ex-1011, § 1, Fig. A.



142. The SGTIN-96 identity structure includes a header, filter value, partition, company prefix, item reference, and serial number, as shown in the figure below. *Id.*, § 3.5.1, Table 4. The figure below illustrates the data fields of an SGTIN-96 number, including a company prefix, an item reference, a partition value, and a filter value. Importantly, the serial number space is defined as a 38-bit sequence of

binary values (0s or 1s); moreover, the “Max. decimal value” shown in the figure is the number of items the 38-bit serial number can identify. *See generally* Ex-1011, § 2.1.2.1.

	Header	Filter Value	Partition	Company Prefix	Item Reference	Serial Number
SGTIN-96	8	3	3	20-40	24-4	38
	0011 0000 (Binary value)	(Refer to Table 5 for values)	(Refer to Table 6 for values)	999,999 – 999,999,999,999 (Max. decimal range*)	9,999,999 – 9 (Max. decimal range*)	274,877,906,943 (Max. decimal value)

*Max. decimal value range of Company Prefix and Item Reference fields vary according to the contents of the Partition field.

Table 4. The EPC SGTIN-96 bit allocation, header, and maximum decimal values.

F. Nitzan (Ex-1022)

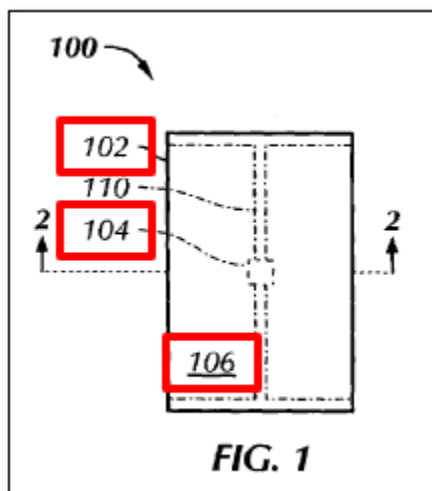
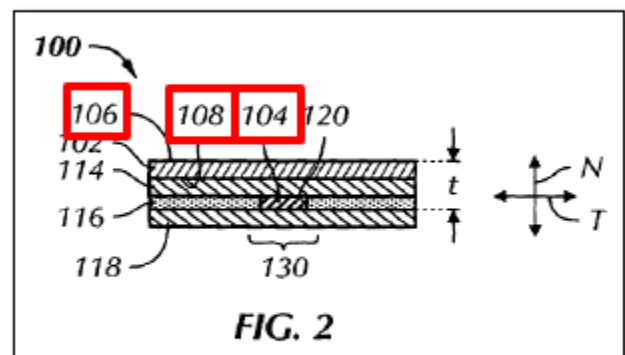
143. Nitzan was filed on June 30, 2005, and is directed to RFID transponders. Ex-1022, Abstract. Nitzan discloses “an integrated circuit (‘IC’), which is arranged to store a code,” comprising “information that is to be transmitted from transponder 28 to reader 32,” such as “an ID number that identifies the package 24.” *See id.*, 7:58-60, 8:67-9:3. Nitzan also discloses that “IC 56 may comprise an organic polymer electronic chip, as known in the art.” *Id.*, 25:64-65.

144. Nitzan is analogous art to the ’967 Patent at least because Nitzan is directed to RFID systems, including RFID transponders and their components. Furthermore, Nitzan teaches that “[a] code comprising information relating to package 24 and/or to transponder 28 can be generated and stored in a memory of

transponder 28. Generally speaking, the code comprises any information that is to be transmitted from transponder 28 to reader 32. For example, the information may comprise an ID number that identifies package 24.” *Id.*, 8:65-9:3. As such, Nitzan teaches RFID transponders that store identification information, and is thus analogous art to ’967.

G. Forster (Ex-1023)

145. Forster was filed on June 2, 2005, and is prior art under § 102(a)(2). Forster is directed to RFID tags including a face stock material. Ex-1023, Abstract. Forster is analogous art to the ’967 Patent because both are directed to the implementation of RFID tags and labels. *Id.* Forster discloses an RFID tag that “includes a face stock 102 and an RFID device 104.” The RFID device is affixed to “the inlay side 108 of the face stock 102.” *Id.*, 2:54-63; Figs. 1, 2.



VIII. The Challenged Claims Of The '967 Patent Are Unpatentable

A. Ground 1A: Challenged Claims 1, 6-8, 10-12, and 20 Are Unpatentable as Obvious over IBM200 in View of Kuhno

1. A POSITA would have been motivated to combine IBM200 and Kuhno

146. IBM200 discloses elements [1A]-[1C], which are generic components typically found in RFID tags, including EPC Gen2 tags. IBM200 also discloses element [1D], and IBM200 in view of Kuhno discloses [1E]-[1F]. IBM200 and Kuhno are analogous art to the '967 Patent as they are directed to methods and systems for commissioning RFID tags with unique identifiers like EPCs.

147. A POSITA would have been motivated to combine IBM200 and Kuhno to solve the problem of allocating unique object identifiers for objects that have the same object classification across multiple manufacturers, and to further associate those objects that may be packaged together, such as in a carton or pallet. *See supra* §V.A.4.

148. A POSITA could begin with IBM200, which teaches a system of allocating blocks of serial numbers across multiple divisions: “different manufacturer divisions that produce objects having the same object classification can be assigned a subset of numbers from a number pool of unassigned numbers to ensure that unique serial numbers are always generated.” Ex-1007, Abstract, 5:7-11. Once those ranges, or blocks, of serial numbers are allocated, POSITAs understood

that the “different manufacturer divisions” could then assign serial numbers within their subset as they wish. *See supra* § VII.A. POSITAS also understood that hierarchical partitioning could be applied within the serial number space, creating additional sub-levels in the existing hierarchy as desired. The notion of hierarchical partitioning by subdividing a number space was well understood, as exemplified by numbering schemes in use at the time for IP addresses, subnet masks, and computer memory allocation as described above. *See supra* §§ V.A.1, 2.

149. Furthermore, IBM200 teaches that the allocation of serial numbers may be achieved through multiple techniques: “the automatic assignment and generation of serial numbers is not limited to simple increasing one-up generated number, but can utilize any mathematical function or serial number generating function that results in unique numbers.” Ex-1007, 4:49-53; *see also id.*, 5:7-22 (providing two examples of serial number allocation among different manufacturing divisions).

150. Accordingly, a POSITA would be motivated to turn to Kuhno because it teaches a technique for allocating unique serial numbers among multiple manufacturing facilities by partitioning a serial number space to further associate objects with a particular production line, pallet, or date of manufacture. *See generally* Ex-1008. Kuhno teaches subdividing the serial number space by partitioning it at the bit level into (1) a more-significant, leading edge partition of bits that identifies, such as a specific Julian calendar date or production line, and (2)

a less-significant partition of remaining bits that contains an incrementing carton number. *See supra* § VII.B.

151. To a POSITA, the date or production line partition at the beginning of the serial number field operates as the claimed “limited number of [MSBs]” within IBM200’s allocated block of serial numbers. In other words, IBM200 in view of Kuhno teaches that the “sectors” of the subdivided “object class serial number space” (each production day being a sector) “are defined by a limited number of [MSBs] of the serial number field,” Ex-1001, 8:14-15, the Julian calendar date or production line prefix being the limited number of MSBs in the serial number space, and the carton number being the remaining LSBs in the serial number space. Because Kuhno describes further subdividing IBM200’s allocated blocks of serial numbers, a POSITA would know that the teachings of the two references could be combined with predictable results and a reasonable expectation of success.

2. Independent Claim 1

[1pre] An RFID transponder comprising:

152. To the extent the preamble is limiting, IBM200 discloses an RFID transponder. IBM200 discloses that an “RFID tag 104 is typically a small object that can be attached or incorporated into object 102 that includes a *transponder*, silicon chips, and/or an antenna.” Ex-1007, 3:19-21 (italics added).

[1A] a substrate;

153. IBM200 renders obvious a substrate, and Kuhno expressly discloses one. Although IBM200 does not explicitly disclose it, a substrate would necessarily exist and would have been known in view of the common knowledge of a POSITA. A POSITA knew that the RFID tag disclosed by IBM200 requires a physical base to be mounted on or printed onto. Ex-1017, 57-58, 134. A POSITA would have understood that a substrate provides this necessary physical structure. *Id.*

154. Moreover, Kuhno teaches the RFID tag comprises “a programmable/encodable radio frequency device and antenna attached to a *substrate* so that the device and antenna can be secured to a desired object such as a carton.” Ex-1008, 5:16-19 (emphasis added).

155. Therefore, IBM200 renders obvious, and Kuhno expressly discloses, element [1A].

[1B] an antenna structure formed on the substrate;

156. The full scope of [1B] would necessarily be present in IBM200, and Kuhno expressly teaches [1B]. IBM200 discloses an RFID tag that “includes a transponder, silicon chips, and/or an antenna.” Ex-1007, 3:19-21. A POSITA understood that the substrate is a base material or surface upon which the electronic components of an RFID tag or transponder are mounted or printed, including the antenna. Ex-1017, 57-58, 134. As such, a POSITA understood that for the antenna

to function or communicate with the other tag components, it must be physically connected to the them via a substrate. *Id.* Therefore, a POSITA would understand that the antenna is formed on the substrate.

157. Kuhno expressly discloses RFID tags with “a programmable/encodable radio frequency device and *antenna attached to a substrate* so that the device and antenna can be secured to a desired object such as a carton.” Ex-1008, 5:15-22.

[1C] an RFID integrated circuit chip which is electrically coupled to the antenna structure;

158. IBM200 teaches element [1C]. IBM200 discloses an RFID tag that “includes a transponder, silicon chips, and/or an antenna,” and notes that RFID tags “can be active devices having internal power sources used to power integrated circuits (ICs) that generate outgoing signals.” Ex-1007, 3:20-29. A POSITA understood that in order to generate those outgoing signals, (and respond to queries, selectively transmit stored data) the RFID IC chip must be electrically coupled to the antenna structure, which is the communicative element of the transponder. POSITAs understood that the “radio frequency device” includes an RFID IC chip, and that to be “encodable,” the chip must be electrically coupled to the antenna to facilitate transmission. Ex-1017, 280-81. The ’967 Patent confirms this, noting that RFID transponders “include an integrated circuit chip having radio frequency circuits, control logic, memory and an antenna structure mounted on a supporting

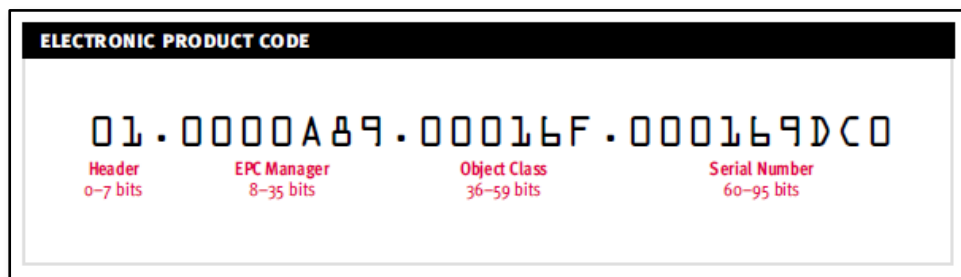
substrate....” Ex-1001, 1:40-43.

159. As such, IBM200 discloses an RFID integrated circuit chip which is electrically coupled to the antenna structure.

[1D] wherein the RFID integrated circuit chip is encoded with a unique object number, the unique object number comprising an object class information space and a unique serial number space,

160. IBM200 teaches wherein the RFID integrated circuit chip is encoded with a unique object number, the unique object number comprising an object class information space and a unique serial number space.

161. IBM200 discloses that the RFID IC chip is encoded with a unique object number: the “RFID tag includes a digitally encoded EPC, which is uniquely assigned to the object.” Ex-1007, 3:1-2. This encoded EPC is the “*unique object number*.” *See supra* §V.A.3, 4. The RFID tag contains an integrated circuit chip with memory space where this unique object identifier, i.e., the EPC, is stored. Ex-1007, 3:32-41 (“The EPC 110 is a simple, compact ‘license plate’ or identifier that uniquely identifies an associated object 102”; “The EPC is divided into numbers that identify manufacturer, *object type*, and *serial number*.”).



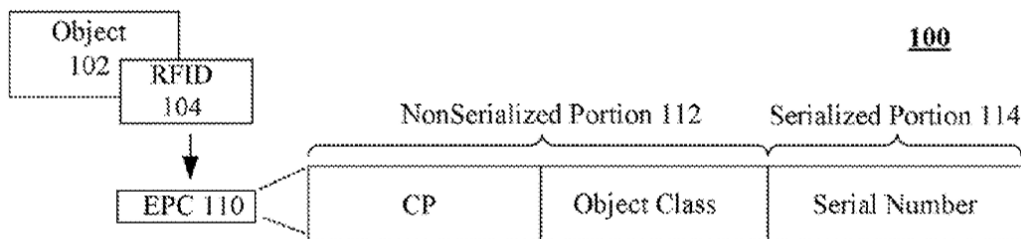
162. IBM200 also discloses that the unique EPC object number is divided into a “nonserialized portion” and a “serialized portion.” The nonserialized portion of the unique object number includes “an object classification” space (among others) and the serialized portion is a unique serial number space that is “specifically associated with object 102.” *Id.*, 3:56-4:9. IBM200 discloses that each unique object number, or “EPC[,] has a pure identity that includes a *nonserialized portion* and a *serialized portion*...the pure identity is an abstract name or number used to identity object that consists of the information required to uniquely identify a specific object and nothing more.” Ex-1007, 3:56-64.

163. IBM200 teaches “the nonserialized portion of the EPC can specify a manufacturer and an object classification of object 102. The serialized portion of EPC can be specifically associated with object. Combining the serialized portion with the nonserialized portion can result in a unique identifier for object.” *Id.*, 3:65-4:11 (italics added).

164. Moreover, a POSITA would have understood that both the ’967 patent and IBM200 reflect the unique object number as *taught by the EPC standard*—indeed, the ’967 patent teaches “a particular EPCglobal key type, for example an SGTIN-96....” Ex-1001, 8:28-29. IBM200 teaches the same. Ex-1007, 3:39-43 (“The five EAN-UCC types include a Serialized Global Trade Item Number (SGTIN) type,...”). IBM200 teaches that the “nonserialized portion” of its unique

object identifier is the portion of an SGTIN number consisting of the “Company Prefix” and “Item Reference”—i.e., the GTIN. Ex-1007, 5:27-52 (“UPC codes use Global Trade Item Numbers (GTINs)...Object A has an associated GTIN of 20045670000103...The TDS specification details a manner to convert GTINs into identity URIs. Object A [is] urn:epc:id:sgtin:00456700.20010.nnnnn...where nnnnn represents a serial number to be generated by any entity possessing an EPCglobal, Inc. approved unique identifier and producing taggable objects.”).

165. IBM200 also discloses a unique serial number space. IBM200 teaches that the “EPC is divided into numbers that identify manufacturer, object type, and serial number.” Ex-1007, 3:39-41. This “serialized portion of EPC can be specifically associated with [an] object.” *Id.*, 4:8-9. IBM200 also teaches that “each EPC identifier includes a serial number that is unique for each tagged object/location/entity or collection thereof.” *Id.*, 1:34-36.



Ex-1007, Fig. 1.

166. IBM200 thus teaches that the unique object class information space (nonserialized portion) and the serial number space (serialized portion) comprise the

unique object identifier (EPC number).

[1E] wherein the unique serial number space is encoded with one serial number instance from an allocated block of serial numbers, the allocated block being assigned a limited number of most significant bits, and

167. IBM200 in view of Kuhno teaches Element [1E]. A POSITA would understand the term “an allocated block of serial numbers” to mean “a pre-authorized range of binary serial numbers.” *See supra* § V.A.1.

168. IBM200 discloses a “unique serial number space” that is encoded with “a serial number instance.” Ex-1007, 1:34-36 (teaching that “each EPC identifier includes a serial number that is unique”). IBM200 teaches that “different manufacturer divisions that produce objects having the same object classification can be assigned a subset of numbers from a number pool of unassigned numbers to ensure that unique serial numbers are always generated.” *Id.*, at 5:7-11. IBM200 further teaches that there are multiple ways to allocate serial numbers, such as “allow[ing] blocks of serial numbers to be generated and handed out at once, which can be significant advantage for many supply chain processes.” *Id.*, at 6:33-36. Moreover, IBM200 clarifies that “automatic assignment and generation of serial numbers is not limited to simple increasing one-up generated number, but can utilize any ... serial number generating function that results in unique numbers” *Id.*, 4:49-53.

169. Kuhno discloses one such serial number generating function, and

advantageously associates an item with a production line and date. A POSITA would have been motivated to combine Kuhno's serial number allocation scheme with IBM200 to include such additional information. *See supra* § VIII.A.1.

170. Kuhno teaches subdividing the serial number space (referred to as the “carton serial number”) before encoding, such that “the carton serial number may include several components desired, for example including the Julian calendar date, the production line number and the carton number.” Ex-1008, 14:25-28. In Kuhno's system, the “Carton Serial Length defines the number of digits in the carton serial number,” including the predefined production line number and date and the carton number, and the “Carton Sequence Length is the number of digits in the carton number.” *Id.*, 14:31-35; *see also* Figure 19 below:

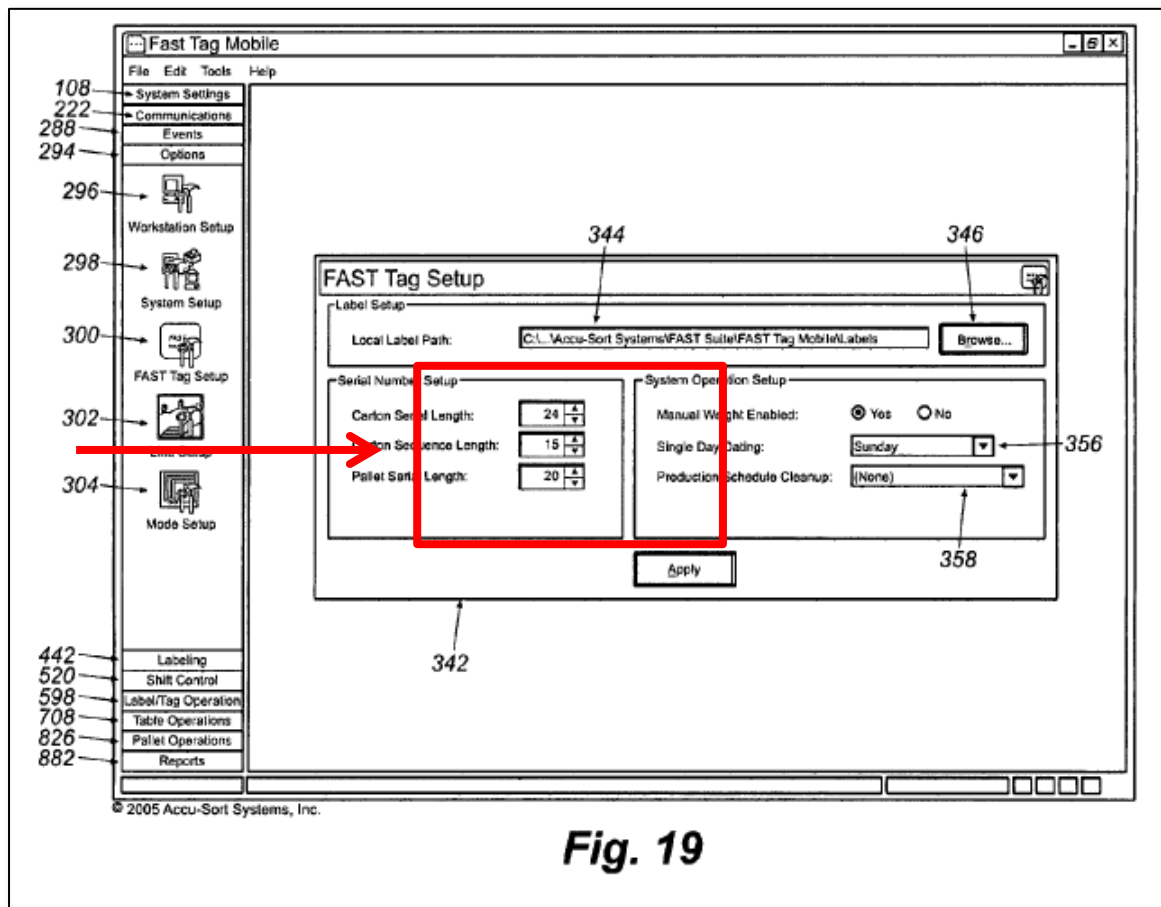
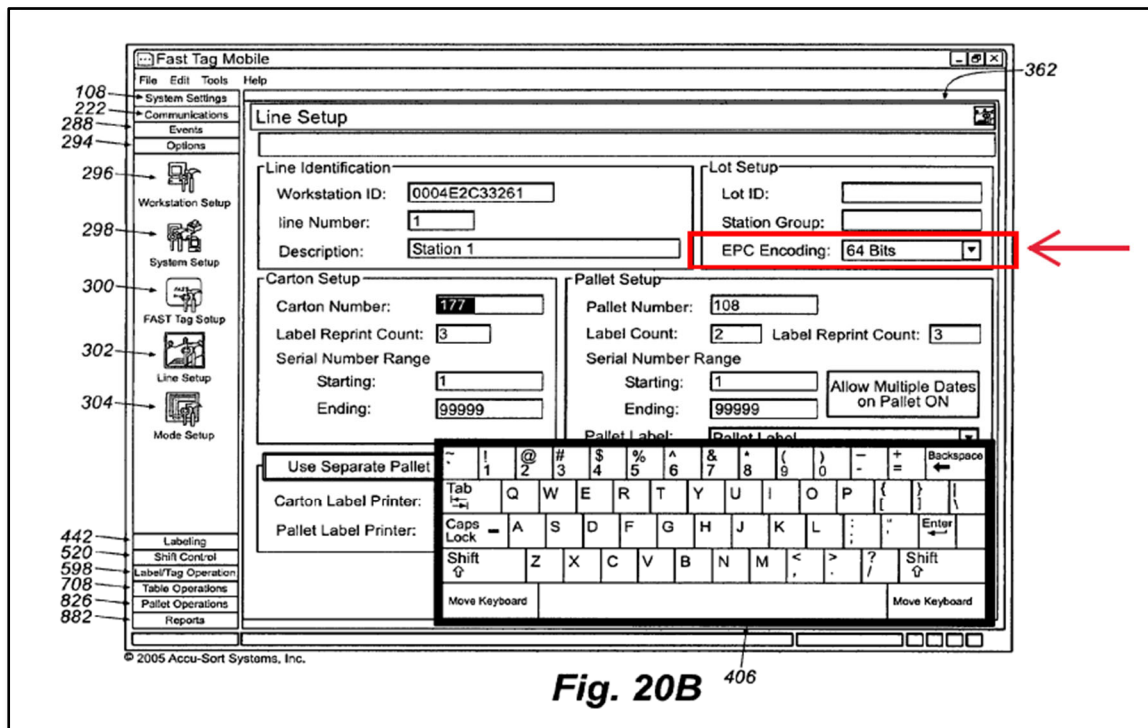


Fig. 19

171. Kuhno thus divides the serial number space into two partitions: (1) a fixed portion, which may include the Julian calendar date and/or production line number; and (2) the incrementing carton number. *Id.*, 14:21-35. By defining and then specifying that the Carton Sequence Length is smaller than the Carton Serial Length, Kuhno necessarily partitions the Carton's serial number space into two fields, one of them being the defined Carton Sequence Length.

172. Furthermore, a POSITA would have understood that the Kuhno's partitions are being encoded onto the RFID tag memory at the bit level—the carton

serial numbers, including the subdivided fields, are encoded in an RFID tag in binary. Kuhno teaches that “[a] drop-down box 374 is used to select the encoding type of the RFID tags used in the system: 64 Bits or 96 Bits.”; *see also id.*, Fig. 20B (illustrating “EPC Encoding: 64 bits”). And in Fig. 19, the “Carton Serial Length” of 24 bits conforms to the SGTIN-64 scheme, which allocates 25 bits to the serial number field. *See* Ex-1008, Figure 20A (setting “64 Bits” in dropdown menu 374 for “EPC Encoding”); *id.*, 23:41-44 (disclosing fields for EPC encode type SGTIN). A POSITA would have understood Kuhno’s reference to a 24-digit Carton Serial Number does not refer to decimal digits, as 24 decimal digits require 80 bits. That is far greater than the 25 total bits assigned to the serial number space in SGTIN-64. As such, a POSITA understood that Kuhno encodes the partitioned serial number space in binary at the bit level.



173. Kuhno also teaches the allocated block being assigned a limited number of most significant bits. First, Kuhno teaches that the predetermined data and line number partition of the serial number is encoded in the serial number's most significant bits. Second, Kuhno teaches that the incrementing carton number partition of the serial number is encoded in the least significant portion of the serial number.

174. Kuhno discloses that its “system application defines a serial number for each carton that is the combination of a predetermined number (discussed above), production line number 366 and an end portion”—i.e., the carton number, which Kuhno describes as “the least significant portion of the serial number.” Ex-1008,

16:8-15. Kuhno also discloses that “[a]s each carton in the production line is processed by system 10, the system application increments the serial number end portion....” *Id.*, 16:16-22.

175. Therefore, because the Carton Serial Length—i.e., the total number of bits—and Carton Sequence Length—the number of bits allocated to the “end portion” of the carton serial number—are defined in advance, Kuhno necessarily teaches that the remaining bits are allocated to the leading portion of the carton serial number.

176. Moreover, Kuhno establishes a *pre-authorized range of binary serial numbers* that *includes a limited, predefined sequence of higher order bits at the leading end*. Kuhno’s fixed-bit-length leading portion—which comprises higher-order information such as the Julian calendar date or production line—necessarily distinguishes the resulting carton serial numbers from others with a different binary prefix. Kuhno teaches that the “production line” number and “production date” encoded in a serial number’s most significant bits are pre-defined for a “predefined group of cartons” to be labeled by one station. *See* Ex-1008, 7:13-16 (“[T]he production date may be predefined, e.g. as the day a predefined group of cartons begins processing through station 10. Thus, even if more than one day is required to process the entire group, the initial date is the production date for all cartons in the group.”), 7:50-52 (“The entire group of cartons and pallets to be scanned and labeled

by one station, such as mobile unit 10, is referred to as a ‘production line.’”).

177. Moreover, a POSITA independently understood that commissioning RFID tags with unique serial numbers partitioned into MSBs and bits of lesser significance is similar to many established methods. For example, in IP addressing, which a POSITA would have been familiar with, the MSBs represent the network portion of the address, and the network portion is hierarchically related to its LSBs, which represent the subnets or hosts within the network. *See supra* § V.A.1. In addition to IP addressing, a POSITA would have recognized that an RFID tag’s memory is conceptually the same as a digital computer’s memory and that storing serial numbers in tag’s memory is analogous to storing data in a specific memory address in the digital computer; and that both are done in binary. *See supra* § V.A.2. The MSBs of the digital computer memory address create blocks of addresses, with the LSBs providing the offsets within those blocks to represent a memory space instance. A POSITA would have understood that each RFID tag’s memory can be viewed as a “memory device” or a partition within a larger “memory block” encompassing all tags in the system. Allocating serial numbers to these tags is essentially assigning unique memory addresses within the RFID system’s overall address space. So, just as memory addresses use MSBs to define larger memory blocks, a POSITA would have known that serial numbers can also use MSBs to define allocated blocks assigned to different production lines, facilities, or other

categories.

178. As such, IBM200 in view of Kuhno discloses *wherein the unique serial number space is encoded with one serial number instance from an allocated block of serial numbers, the allocated block being assigned a limited number of most significant bits.*

[1F] wherein the unique serial number space comprises the limited number of most significant bits uniquely corresponding to the limited number of most significant bits of the allocated block and of remaining bits of lesser significance that together comprise the one serial number instance.

179. IBM200 in view of Kuhno renders obvious [1F]. IBM200 discloses a unique serial number space. *See* [1D]. IBM200 in view of Kuhno discloses the unique serial number space being encoded with one serial number instance from an allocated block of serial numbers, the allocated block being assigned a limited number of most significant bits. *See* [1E]. Kuhno in particular teaches partitioning the unique serial number space into a limited number of MSBs (comprising the higher-order information at the leading end of the carton serial number) and remaining bits of lesser significance, or LSBs (the lower-order “carton number” at the end of the carton serial number), which together comprise the one serial number instance. *See* [1E].

180. Further, Kuhno teaches partitioning the unique serial numbers space at the bit level as recited by the “uniquely corresponding to” limitation. Kuhno assigns

serial numbers with most significant bits representing (1) a production date and (2) a production line for a “pre-defined group of cartons” that are “labeled by one station.” *See* [1E], Ex-1008, 7:13-16, 7:50-52. Kuhno further teaches use of the same production date information for the entire pre-defined group, such that “even if more than one day is required to process the entire group, the initial date is the production date for all cartons in the group.” Ex-1008, 7:13-16.

181. Because the pre-defined group of items all have the same production date and all are part of the same production line from the same station, all items in the pre-defined group have the same most significant bits in their serial numbers. Moreover, because that shared date/production line combination are unique, Kuhno teaches the claimed unique correspondence between the MSBs and the range of serial numbers in the allocated block. As demonstrated in the MIT White paper, a POSITA would have been knowledgeable about established hierarchical systems and methods such as IP addressing and computer memory allocation. *See generally* Ex-1014; *see also supra* §§ V.A.1, 2. As such, a POSITA knew that in both systems, the MSBs are typically used to define larger, more general classifications, while LSBs identify specific instances within a classification. A POSITA would have understood that the same principle applies in IBM200, which teaches holding the nonserialized portion of the EPC constant in order to establish an object classification hierarchy to guarantee uniqueness. In other words, IBM200’s EPC

“nonserialized” and “serialized” portions are related by the pattern of *parent.child*, where “parent” is the higher-level category and the “child” is an instance in that category. Kuhno’s division of the serial number into a predetermined portion and an incrementing end portion parallels this approach: Kuhno subdivides the carton serial number into, for example, a *predetermined number.carton number*, where the predetermined number is a fixed, block-defining categorical prefix, and the incrementing portion identifies a serial number instance within that block (much like hosts within a subnet or memory locations within a block). As such, a POSITA understood that the same hierarchical numbering principles which apply to IP addressing and memory allocation also applies to and is implemented in Kuhno.

182. Therefore, IBM200 in view of Kuhno discloses *wherein the unique serial number space comprises the limited number of most significant bits uniquely corresponding to the limited number of most significant bits of the allocated block and of remaining bits of lesser significance that together comprise the one serial number instance.*

183. Therefore, IBM200 in view of Kuhno discloses Claim 1 of the ’967 Patent.

3. Claim 6: The RFID transponder of claim 1, wherein the allocated block is defined by at least 3 most significant bits.

184. IBM200 and Kuhno disclose Claim 1, and Kuhno discloses the

additional limitations in Claim 6. Kuhno discloses a Carton Serial Length of 24 bits and a Carton Sequence Length of 15 bits, thereby defining an allocated block of 9 MSBs, which is “at least 3 most significant bits.” Ex-1008, Fig. 19.

185. Therefore, claim 6 is obvious in view of IBM200 and Kuhno.

4. Claim 7: The RFID transponder of claim 1, wherein the allocated block is one of a plurality of allocations intermittently received by an encoder.

186. IBM200 and Kuhno disclose Claim 1, and IBM200 discloses the additional limitations in Claim 7. IBM200 teaches that the allocated block is one of a plurality of allocations intermittently received by an encoder, by teaching that encoders in different manufacturing divisions are “linked to a centralized system that assigns serial numbers,” Ex-1007, 5:18-22, and each encoder identifies its corresponding pool or block of allocated numbers, then encodes serial numbers from that pool, *Id.*, 6:37-58. A POSITA would have understood that the allocation process inherently involves intermittent allocations. For example, when one division exhausts its pool of serial numbers, IBM200 teaches that a new block is to be allocated from the central pool. Moreover, IBM200 also teaches “lifecycle management techniques” wherein “previously assigned serial numbers can be deleted, deallocated, or unassigned once they are no longer needed, and “[d]eallocated serial numbers can be re-assigned to new objects as needed. Ex-1007, 6:28-32. Because these assignments are made at different times with undefined

intervals, the encoders, by definition, receive the allocations intermittently

187. Therefore, claim 7 is obvious in view of IBM200 and Kuhno.

5. Claim 8: The RFID transponder of claim 1, wherein the RFID integrated circuit chip is comprised of silicon.

188. IBM200 and Kuhno disclose Claim 1, and IBM200 discloses the additional limitations in Claim 8. IBM200 teaches an RFID tag that “includes a transponder, *silicon chips*, and/or an antenna,” and notes that RFID tags “can be active devices having internal power sources used to power integrated circuits (ICs) that generate outgoing signals.” Ex-1007, 3:20-29. A POSITA understood the “chip” disclosed by IBM200 is an integrated circuit chip. *See* [1C].

189. Therefore, IBM200 teaches an RFID transponder having a RFID IC circuit that is a “silicon chip.” *See* [1C]; *see also* Ex-1007, 3:19-21.

190. Therefore, claim 8 is obvious in view of IBM200 and Kuhno.

6. Claim 10: The RFID transponder of claim 1, wherein the substrate, antenna structure, and RFID integrated circuit chip are covered by a face stock material.

191. IBM200 and Kuhno disclose Claim 1, and Kuhno discloses the additional limitations recited in Claim 10. Kuhno teaches a label that “may store ... EPC information on a radio frequency identification (“RFID”) tag attached to the label stock.” Ex-1008, 1:23-27. Kuhno also teaches an embodiment wherein “RFID printer 20 uses printable label stock that contains an embedded RFID tag.” *Id.*, 5:23-

25. Kuhno explains that “Label” “refers generally to a substrate capable of attachment to a carton or pallet and upon which information regarding the carton or pallet is disposed through a barcode, an RFID tag, or both.” Ex-1008, 5:29-22. Moreover, Kuhno teaches that the term “RFID label” “refer[s] to a programmable/encodable radio frequency device and antenna attached to a substrate so that the device and antenna can be secured to a desired object such as a carton.” *Id.*, 5:16-19. A POSITA would have understood *face stock material* refers to printable label stock material.

192. Therefore, claim 10 is obvious in view of IBM200 and Kuhno.

7. Claim 11: The RFID transponder of claim 1, wherein the RFID integrated circuit chip is disposed on the substrate.

193. IBM200 and Kuhno disclose Claim 1, and Kuhno discloses the additional limitations recited in Claim 11. *See* Claim 1. Kuhno teaches RFID labels having “a programmable/encodable radio frequency device and antenna attached to a substrate.” Ex-1008, 5:16-19. Additionally, a POSITA understood that the “programmable/encodable radio frequency device” includes an RFID IC chip, and that the disposition of the RFID IC chip on a substrate is conventional aspect of RFID tags. *See supra* § V.A.

194. Accordingly, claim 11 is obvious in view of IBM200 and Kuhno.

8. Claim 12: The RFID transponder of claim 1, wherein the limited number of most significant bits that correspond to the allocated block further correspond to an encoder that encodes the RFID transponder.

195. IBM200 and Kuhno disclose Claim 1, and IBM200 and Kuhno disclose the additional limitations in Claim 12. Encoders are devices used to “commission[]” RFID tags. Ex-1001, Abstract. “Commissioning [is] the process of encoding specific information [in order to] associate[] a specific object with a unique RFID transponder.” *Id.*, 1:48-50.

196. Kuhno teaches that a component of the leading end partition in the serial number space is the “production line,” which refers to “[t]he entire group of cartons and pallets to be scanned and labeled by one station, such as mobile unit 10.” Ex-1008, 7:51-52. Thus, the “production line number” that Kuhno teaches may be included in a serial number’s most significant bits, corresponds to a specific encoding station/lot combination for a group of items to be encoded.

197. Kuhno further teaches that a “station” or “mobile unit 10” includes “an RFID printer,” which is the “device that encodes RFID tags.” *Id.*, 4:53-55, 5:4-15. Therefore, Kuhno teaches including in a serial number’s MSB’s a “production line number” that corresponds to the encoder/station that encodes the RFID transponder.

198. Accordingly, Claim 12 is obvious in view of IBM200 and Kuhno.

199.

- 9. Claim 20: The integrated circuit of claim 16, wherein the limited number of most significant bits that correspond to the allocated block further correspond to an encoder that encodes the memory of the integrated circuit.**

200. Claim 20 is substantively identical to claim 12, and is thus also obvious in view of IBM200 and Kuhno. *See supra* § VIII.A.8.

B. Ground 1B: Claims 2-5, 13-16, and 17-19 are Unpatentable Over IBM200 in view of Kuhno and TDS

- 1. A POSITA Would Have Been Motivated To Combine TDS with IBM200 and Kuhno**

201. As explained above, a POSITA would have been motivated to combine IBM200 and Kuhno. *See supra* § VIII.A.1. Like IBM200 and Kuhno, TDS is analogous art directed to related methods and systems for encoding RFID tags, including RFID tags having associated EPCs. *See supra* § VII.E. In particular, a POSITA would have motivated to apply the TDS encoding schemes to the teachings of IBM200 and Kuhno, such as when encoding Gen2 RFID tags with EPCs. For example, the RFID tag industry, which included the U.S. Department of Defense (“DoD”), was developing UHF Gen2 RFID tags. *See, e.g.*, Ex-1020, 2 (“Radio Frequency Identification will be a mandatory DoD requirement on solicitations issued on or after October 1, 2004....”); *id.*, 1 (“DoD will migrate to the next generation tag (UHF Gen 2) and supporting technology. When the specification for UHF Gen 2 is finalized, the Department will announce a transition plan to this technology....”). Gen2 was driven, in part, by a number of companies that intended

to use Gen2 tags encoded with EPC numbers to track their inventory. Naturally then, in order to conform to these standards and engage customers, a POSITA would have been motivated to implement the industry specification for encoding those Gen2 tags—i.e., TDS. *See* Ex-1011, 4 (“This Tag Data Standards Version 1.3 is aimed for use in Gen 2 Tags....”). The DoD RFID Policy memo specified several EPC encoding schemes, including SGTIN-96. *See generally* Ex-1020.

202. IBM200 and Kuhno teach encoding EPCs using specific EPC encoding types such as SGTIN-96. Ex-1007, 3:44-51; *see also* Ex-1008, 23:41-44 (“RFID tab 742 provides fields for EPC encode type (SGTIN, SSCC, SGLN and GRIA), EPC product code, EPC manufacturer number, and RFID tag serial number range.”). As such, POSITA encoding RFID tags with EPCs as taught in IBM200 and Kuhno would be motivated to implement the TDS to ensure compliance with that standard. Moreover, because IBM200 and Kuhno both teach encoding RFID tags with EPCs according to specific EPC encoding types addressed in the TDS, a POSITA would know how to combine the teachings of these three references, with predictable results and a reasonable expectation of success.

2. Claim 2: The RFID transponder of claim 1, wherein the object class information space is encoded with information comprising a company prefix, an item reference, a partition value, and a filter value.

203. IBM200 and Kuhno disclose Claim 1, and TDS discloses the additional

limitations in Claim 2. TDS teaches that the object class information space is encoded with information comprising a company prefix, an item reference, a partition value, and a filter value. Ex-1011, § 2.1.2.1. The figure below illustrates the data fields of an SGTIN-96 number, including a company prefix, an item reference, a partition value, and a filter value. Note that the serial number space is defined as a 38-bit sequence of binary values (0s or 1s); moreover, the “Max. decimal value” shown in the figure is the number of items the 38-bit serial number can identify. *See generally Id.*

	Header	Filter Value	Partition	Company Prefix	Item Reference	Serial Number
SGTIN-96	8	3	3	20-40	24-4	38
	0011 0000 (Binary value)	(Refer to Table 5 for values)	(Refer to Table 6 for values)	999,999 – 999,999,999,999 (Max. decimal range*)	9,999,999 – 9 (Max. decimal range*)	274,877,906,943 (Max. decimal value)

204. Accordingly, and as depicted in the figure above, a POSITA would have understood that “GS1 defines EPCglobal SGTIN number fields as having a company prefix, an item reference, a partition value, and a filter value that comprise the object class information space.” Ex-1001, 9:7-13; *see also supra* §V.A.

205. Therefore, claim 2 is obvious in view of IBM200, Kuhno, and TDS.

3. Claim 3: The RFID transponder of claim 1, wherein the unique object number is a portion of a serialized global trade item number.

206. IBM200 and Kuhno disclose Claim 1, and TDS discloses the additional limitations in Claim 3. TDS teaches that the unique object number is a portion of a serialized global trade item number. Ex-1011, § 2.5 (“The EPC Tag Encoding scheme for SGTIN permits the direct embedding of EAN.UCC System standard GTIN and Serial Number codes on EPC tags; see also *id.*, § 2.5 (“2.5 Serialized Global Trade Item Number (SGTIN)”)).

207. Therefore, IBM200, Kuhno, and TDS disclose claim 3.

4. Claim 4: The RFID transponder of claim 3, wherein the serialized global trade item number is an EPC SGTIN-96 encoding.

208. IBM200, Kuhno, and TDS disclose Claim 3, and TDS discloses the additional limitations in Claim 4. TDS teaches the serialized global trade item number being an EPC SGTIN-96 encoding, as illustrated in the figure below. Ex-1011, § 2.5. The figure below illustrates the data fields of an SGTIN-96 number, including a company prefix, an item reference, a partition value, and a filter value. Note that the serial number space is defined as a 38-bit sequence of binary values (0s or 1s); moreover, the “Max. decimal value” shown in the figure is the number of items the 38-bit serial number can identify. *See generally Id.*, § 2.1.2.1.

	Header	Filter Value	Partition	Company Prefix	Item Reference	Serial Number
SGTIN-96	8	3	3	20-40	24-4	38
	0011 0000 (Binary value)	(Refer to Table 5 for values)	(Refer to Table 6 for values)	999,999 – 999,999,999,999 (Max. decimal range*)	9,999,999 – 9 (Max. decimal range*)	274,877,906,943 (Max. decimal value)

209. Therefore, claim 4 is obvious in view of IBM200, Kuhno, and TDS.

5. Claim 5: The RFID transponder of claim 1, wherein the unique serial number space is at least 38 bits.

210. IBM200 and Kuhno disclose Claim 1, and TDS discloses the additional limitations in Claim 5. TDS teaches that the unique serial number space of SGTIN-96 is 38 bits. Ex-1011, § 2.5. Importantly, that the serial number space is defined as a 38-bit sequence of binary values (0s or 1s); moreover, the “Max. decimal value” shown in the figure is the number of items the 38-bit serial number can identify. *See generally Id.*, § 2.1.2.1.

	Header	Filter Value	Partition	Company Prefix	Item Reference	Serial Number
SGTIN-96	8	3	3	20-40	24-4	38
	0011 0000 (Binary value)	(Refer to Table 5 for values)	(Refer to Table 6 for values)	999,999 – 999,999,999,999 (Max. decimal range*)	9,999,999 – 9 (Max. decimal range*)	274,877,906,943 (Max. decimal value)

211. Therefore, claim 5 is obvious in view of IBM200, Kuhno, and TDS.

6. Independent claim 13:

[13pre] A RFID transponder comprising:

212. IBM200 discloses an RFID transponder, as does Kuhno, for the same reasons as element [1pre]. *See* [1pre].

[13A] a substrate;

213. IBM200 discloses substrate, as does Kuhno, for the same reasons as element [1A]. *See* [1A].

[13B] an antenna structure formed on the substrate; and

214. IBM200 discloses an antenna structure formed on the substrate, as does Kuhno, for the same reasons as element [1B]. *See* [1B].

[13C] an RFID integrated circuit chip which is electrically coupled to the antenna structure,

215. IBM200 discloses an RFID integrated circuit chip which is electrically coupled to the antenna structure, as does Kuhno, for the same reasons as element [1C]. *See* [1C].

[13D] wherein the RFID integrated circuit chip is encoded with a global trade item number including a unique object number, the unique object number comprising an object class information space including a block of at least 50 bits and a unique serial number space including a block of at least 38 bits,

216. IBM200 discloses the limitations of [13D] except for “an object class information space including a block of at least 50 bits and a unique serial number space including a block of at least 38 bits,” for the same reasons as element [1D]. *See* [1D]. TDS teaches an object class information space having a block of at least

50 bits for encoding filter value (3 bits), partition value (3 bits), company prefix (at least 20 bits), and item reference (at least 24 bits), and a unique serial number space with 38 bits:

	Header	Filter Value	Partition	Company Prefix	Item Reference	Serial Number
SGTIN-96	8	3	3	20-40	24-4	38
	0011 0000 (Binary value)	(Refer to Table 5 for values)	(Refer to Table 6 for values)	999,999 – 999,999,999,999 (Max. decimal range*)	9,999,999 – 9 (Max. decimal range*)	274,877,906,943 (Max. decimal value)

Ex-1011, § 2.5. The figure above illustrates the data fields of an SGTIN-96 number, including a company prefix, an item reference, a partition value, and a filter value. Note that the serial number space is defined as a 38-bit sequence of binary values (0s or 1s); moreover, the “Max. decimal value” shown in the figure is the number of items the 38-bit serial number can identify. *See generally Id.*, § 2.1.2.1.

217. Accordingly, limitation [13D] is obvious in view of IBM200, Kuhno and TDS.

[13E] wherein the unique serial number space has one serial number instance from an allocated block of serial numbers, the allocated block being assigned a limited number of most significant bits, and

218. IBM200 and Kuhno renders obvious limitation [13E] for the same reasons as [1E]. *See* [1E].

[13F] wherein the unique serial number space comprises at least 3 most significant bits uniquely corresponding to the limited number of most significant bits of the allocated block.

219. IBM200 and Kuhno disclose [13F] for the same reasons as element [1F] and claim 6. *See* [1F] and claim 6.

220. Accordingly, claim 13 is obvious in view of IBM200, Kuhno, and TDS.

7. Claim 14: The RFID transponder of claim 13, wherein the unique serial number space further comprises remaining bits of lesser significance that together with the at least 3 most significant bits comprise the one serial number instance.

221. IBM200, Kuhno, and TDS disclose Claim 13, and IBM200 and Kuhno disclose the additional limitations in Claim 14, for the same reasons as element [1F] and Claim 6. As discussed in [1F], IBM200 and Kuhno disclose a unique serial number space comprised of an MSB “segment” and an LSB “segment.” *See supra* [1F]. Moreover, in claim 6, IBM200 and Kuhno renders obvious the most significant bits comprise at least bits. *See supra* § VIII.A.3.

222. Accordingly, claim 14 is obvious in view of IBM200 and Kuhno.

8. Claim 15: The RFID transponder of claim 13, wherein the at least 3 most significant bits that correspond to the allocated block further correspond to an encoder that encodes the RFID transponder.

223. IBM200, Kuhno, and TDS disclose Claim 13, and IBM200 and Kuhno disclose the additional limitations in Claim 15, for the same reasons as Claim 12 and Kuhno render claim 15 obvious. *See supra* § VIII.A.2.

9. Independent Claim 16

[16pre] An integrated circuit, comprising:

224. IBM200 disclose an RFID tag with an integrated circuit chip for the same reasons as element [1C]. *See* [1C].

[16A] a radio frequency circuit configured to backscatter a return link signal for processing by an RFID interrogator;

225. TDS discloses [16A]. TDS teaches that the standard applies to Generation 2 UHF RFID tags (“Gen2 tags”). Ex-1011, 3:35-38. A POSITA would have understood that Gen2 tags communicate information to an interrogator using a technique known as backscatter modulation to reflect an incoming RF signal back to the interrogator, and this is confirmed by the ’967 Patent, where it discusses a “common type of passive inlay transponder,” “a battery-assist tag,” which “is a hybrid RFID transponder that uses a battery to power the RFID chip and a backscatter return link to the interrogator.” Ex-1001, 6:8-14.

226. Therefore, TDS discloses this [16A].

[16B] control logic; and

227. IBM200 teaches [16B]. Element [16B] generically recites a common component in RFID IC chips, as the ’967 Patent concedes. A POSITA understood that the RFID IC chips disclosed in IBM200 comprised control logic, which is necessary to control its operation to “store data that is selectively transmitted to authorized RFID transceivers” and “respond to radio frequency queries.” Ex-1007,

3:19-24; Ex-1017, 12 (“In programmable systems, write and read access to the memory and any requests for write and read authorization must be controlled by the data carrier’s internal logic.”). The ’967 Patent confirms that RFID transponders typically “include an integrated circuit chip having radio frequency circuits, control logic, memory and an antenna structure mounted on a supporting substrate....” Ex-1001, Background, 1:40-44.

[16C] a memory,

228. IBM200 teaches [16C]. Element [16C] generically recites a common component in RFID IC chips, as the ’967 Patent concedes. IBM200 discloses RFID tags with “physical memory,” noting they are the “size of a serial number is limited because of physical memory constraints of an RFID tag.” Ex-1007, 5:67-6:, including RFID tags with a standard “64-bit” memory and resulting encoding schemes. *Id.*, 6:4-6. Moreover, a POSITA understood IBM200’s RFID tags comprised a memory in order to “store data that is selectively transmitted” as disclosed. Ex-1017, 12 (“Three main procedures are used to store the data: in inductively coupled RFID systems EEPROMs (electrically erasable programmable read-only memory) are dominant.”). The ’967 Patent further confirms that RFID transponders typically “include an integrated circuit chip having radio frequency circuits, control logic, memory and an antenna structure mounted on a supporting substrate....” Ex-1001, Background, 1:40-44.

[16D] wherein the memory is configured with a unique object number space, the unique object number space comprising an object class information space and a unique serial number space,

229. IBM200 teaches [16D] for the same reasons as element [1D]. *See* [1D].

[16E] wherein the unique serial number space is encoded with one serial number instance from an allocated block of serial numbers, the allocated block being assigned a limited number of most significant bits, and,

230. IBM200 and Kuhno disclose [16E] for the same reasons as element [1E]. *See* [1E].

[16F] wherein the unique serial number space comprises the limited number of most significant bits uniquely corresponding to the limited number of most significant bits of the allocated block and remaining bits of lesser significance that together comprise the one serial number instance.

231. IBM200 and Kuhno disclose [16F] for the same reasons as element [1F]. *See* [1F].

232. Therefore, IBM200, Kuhno, and TDS disclose claim 16.

10. Claim 17: The integrated circuit of claim 16, wherein the object class information space is encoded with information comprising a company prefix, an item reference, a partition value, and a filter value.

233. IBM200, Kuhno, and TDS disclose Claim 16, and IBM200, Kuhno and TDS disclose the additional limitations in Claim 17, for the same reasons as Claim 2. *See* Claim 2.

- 11. Claim 18: The integrated circuit of claim 16, wherein the unique object number is a portion of a serialized global trade item number.**

234. IBM200, Kuhno, and TDS disclose Claim 16, and IBM200, Kuhno and TDS disclose the additional limitations in Claim 18, for the same reasons as Claim 3. *See supra* § VIII.B.3.

- 12. Claim 19: The integrated circuit of claim 18, wherein the serialized global trade item number is an EPC SGTIN-96 encoding.**

235. IBM200, Kuhno, and TDS disclose Claim 18, and IBM200, Kuhno and TDS disclose the additional limitations in Claim 19, for the same reasons as Claim 4. *See supra* § VIII.B.4.

C. Ground 1C: Claim 9 is Unpatentable over IBM200 in view of Kuhno and Nitzan

- 1. Claim 9: The RFID transponder of claim 1, wherein the RFID integrated circuit chip is comprised of polymer.**

236. IBM200 and Kuhno disclose Claim 1, and Nitzan discloses the additional limitations in Claim 9. Nitzan is analogous art, as it discloses “an integrated circuit (“IC”), which is arranged to store a code including information....” Ex-1022, 7:58-60. “Generally speaking, the code comprises any information that is to be transmitted from transponder 28 to reader 32. For example, the information may comprise an ID number that identifies the package 24.” *Id.*, 8:67-9:3. “

237. Nitzan discloses that “IC 56 may comprise an organic polymer

electronic chip, as known in the art.” Ex-1022, 25:64-65. The ’967 Patent confirms that it was common knowledge that RFID IC chips can be comprised of polymer. The ’967 Patent recognizes that “combining the benefits of the latest technology in RFID transponders with sensing devices, a broader class of devices called wireless sensors is emerging.” Ex-1001, 1:65-67. Those “wireless sensors” are “made using silicon circuits, polymer circuits....” *Id.*, 2:11. Therefore, a POSITA would have understood that the RFID integrated circuit chip of IBM200 could be comprised of polymer.

238. Therefore, Nitzan discloses the additional limitations of claim 9.

D. Ground 2A: Challenged Claims 1-8 and 11-20

1. A POSITA would have been motivated to combine Bowers, Tilson, and TDS with a Reasonable Chance of Success

239. Claim elements [1A]-[1C] generically recite the typical components of an RFID tag, as taught by Bowers and Tilson. TDS discloses [1D], which teaches a unique object identifier comprising an object class information space and a unique serial number space. Tilson in view of Bowers discloses [1E]: Tilson teaches assigning serial numbers with a limited number of MSBs, while Bowers teaches allocating a range of those serial numbers. Tilson in view of Bowers renders obvious [1F].

240. A POSITA would have been motivated to combine the teachings of

Bowers, Tilson and TDS to solve the problem of allocating unique object identifiers across multiple manufacturers and product categories. Bowers, Tilson, and TDS are analogous art directed to methods for commissioning RFID tags. Furthermore, both Bowers and Tilson use RFID tags encoded with data structures that partition the serial number into fields, including an “MSB field” signifying specific item classifications or attributes. The TDS standard complements these disclosures by teaching categorical prefixes to create a hierarchical number structure that forces uniqueness. Ex-1029, §7.4, Lines 952-54.

241. A POSITA would have started with Bowers, filed in 1997, which teaches allocating a serial numbers to a predefined set of tags, which are then applied to a predefined set of articles. Ex-1009, 10:47-55. These serial numbers are also recorded in a database, which includes entries for various information about the article, such as an “Article identification information 414” or “article status 412.” *Id.*, Fig. 13, 10:61-64. Bowers also teaches that the tag serial number 404 is comprised of a “changeable portion 406,” and a “fixed portion 408,” and that “[o]ptionally, some or all of the other information 408 is stored in the tag 10.” *Id.*, 10:65-66.

242. A POSITA would have been motivated to use TDS with the structural aspects of Bowers’ tag, as TDS included an “object class” field for Bowers’ disclosure of information identifying an item’s type. Ex-1009, 6:36-38 (“...some or

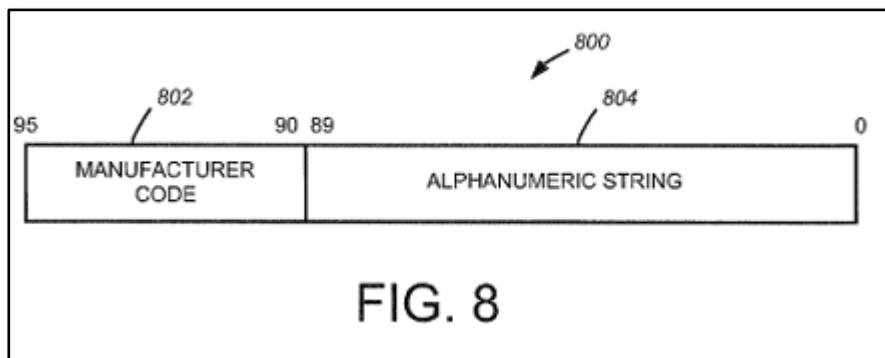
all of the other information 208 is also stored in the tag”); 6:61-64 (“Article identification information 212 is a description of the article ... (e.g., SONY 19" TV, Model No. KV1926....)”).

243. Moreover, Bowers teaches the use of RFID tags in a retail environment while emphasizing the importance of avoiding duplication of serial numbers across entities, such as different stores or chains of stores. *Id.*, 6:39-47 (“Preferably none of the serial numbers in the database 200 are used in databases or EAS systems of other entities In this manner, there is no chance of a tag used by another entity (e.g., another chain of stores) can be confused with any of the tags 10 in the database 200.”), 9:28-37 (“...in a retail environment, the likely explanation when no match occurs is that the tag ... belongs to an article associated with a different store, or different chain of stores.”).

244. As such, a POSITA would have been motivated to turn to Tilson, which teaches one way of avoiding duplication of serial numbers because it discloses an RFID tag with “a serial number that uniquely identifies” an item that is partitioned into two parts: (1) a predetermined set of MSBs identifying a manufacturer code (or other information), followed by (2) a set of LSBs with an “alphabetic, numeric, or alphanumeric string” identifying the individual item. Ex-1010, 7:66-8:3 (“...serial number has a manufacturer code portion 802 and an alphabetic, numeric, or alphanumeric portion 804, which together form the unique serial number....”); 8:3-

5 (“...bits 90-95 of a 96-bit data structure may store the manufacturer code and bits 0-89 may store the alphanumeric portion”).

245. Illustrated in the figure below are the two fields, including the MSBs field occupying “bits 90-95 of a 96-bit data structure.” *Id.*, 8:3-4. Tilson teaches that the alphanumeric string is composed of human-readable symbols displayed on a screen by a software and represented and stored in RFID tag as binary values. A POSITA understood that these two forms—the human-readable string and the binary representation—are equivalent. Put differently, *the alphanumeric string and its binary representation are not two different values or information; rather, they are equivalent and represent the same information.* Moreover, a POSITA would have understood that the information saved in the RFID tag is actually in binary form, and that it is so in all cases. *See supra* § VII.D.



246. Tilson also teaches that a POSITA understood that the data structure of Fig. 8 may be modified to identify the component’s batch instead of its manufacturer. *Id.* A POSITA would have understood that this is merely a different

category to subdivide the serial number space.

247. Accordingly, a POSITA would have been motivated to combine Tilson's partitioning of the serial number space with Bowers in order to achieve Bowers' goal of avoiding duplication of serial numbers across entities. Moreover, Tilson was readily combinable with Bowers since Bowers also discloses a partitioned serial number consisting of an initial set of bits at the leading end of the number, followed the remaining bitts, which identified an individual item. Ex-1009, 10:47-55.

248. Indeed, a POSITA was aware of partitioning an RFID tag's serial number space to carry different information. For example, *Dummies* teaches subdividing a serial number space to identify facilities to avoid duplication: "A range of serial numbers for each product is allocated to each manufacturing facility. ... In this way, the serial number is effectively subdivided into a facility number, line number, and subserial number...." Ex-1013, 240.

249. Moreover, a POSITA also understood that such a serial number could be readily implemented in a TDS compliant tag, as Tilson discloses the use of a "G2 DogBone model of RFID tag," which was known to be a TDS/Gen2 tag. Ex-1010, 3:52-55.

250. Therefore, a POSITA would have been motivated to combine Tilson's MSB-defined serial number data structure with Bowers' allocation of encoded RFID

tags, and adapt the combination to TDS's widely adopted standard with predictable results and a reasonable expectation of success.

2. Independent Claim 1

[1pre] An RFID transponder comprising:

251. To the extent the preamble is limiting, Bowers, or Bowers in view of Tilson, discloses [1pre]. Bowers discloses an RFID tag. Ex-1009, 4:59. A POSITA would have understood that term was interchangeable with RFID transponder. Ex-1001, 5:65-67 (“[T]ransponders is a term for wireless sensors that is often used interchangeably with the term tags”). Furthermore, Tilson discloses that “[t]he RFID assembly 100 includes a back plate 102, an RFID tag or transponder 104, and a cover 106.” Ex-1010, 2:64-66.

[1A] a substrate:

252. Bowers or Bowers in view of Tilson discloses [1A]. A POSITA understood that Bowers' RFID tags includes substrates. Bowers disclose an RFID tag that includes “a coil antenna” and “an integrated circuit (IC).” Ex-1009, 4:59-5:12; Ex-1017, 57 (“Figure 3.30 illustrates the basic layout of a surface wave transponder. A finger-shaped electrode structure—the interdigital transducer—is positioned at the end of a long piezoelectrical substrate....”); *see also id.*, Fig. 3.30, 3.31 below depicting RFID tags on substrates. Furthermore, Tilson discloses a transponder's construction with components affixed to a backing or support

substrate: “The RFID tag 104 may be formed on a film, such as paper, having an adhesive backing.” Ex-1010, 3:22-23. A POSITA knew that the film disclosed in Tilson is a substrate because a POSITA would have recognized that substrates are conventional components in RFID tags that provide a stable and supportive structure for mounting critical components such as the integrated circuit and antenna. Ex-1017, 57. The ’967 patent also admits that RFID transponders, typically “include an integrated circuit chip having radio frequency circuits, control logic, memory and an antenna structure mounted on a supporting substrate.” Ex-1001, 1:40-45.

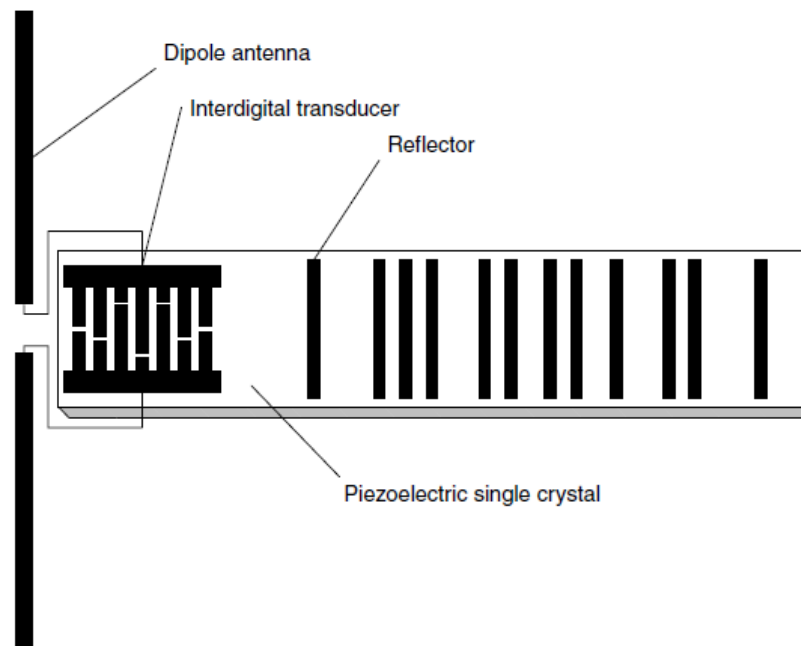


Figure 3.30 Basic layout of an SAW transponder. Interdigital transducers and reflectors are positioned on the piezoelectric crystal

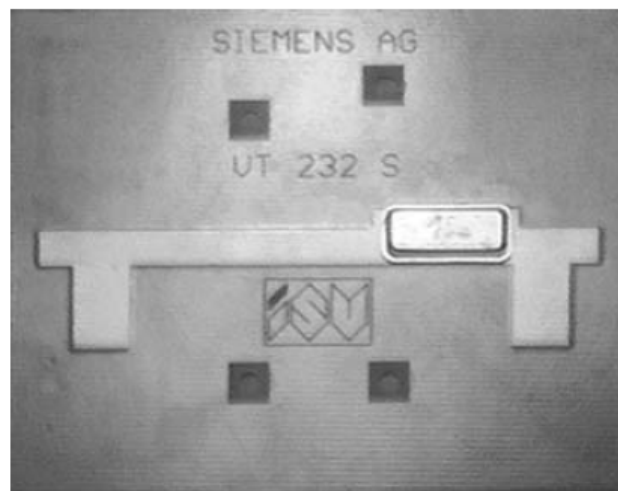


Figure 3.31 Surface acoustic wave transponder for the frequency range 2.45 GHz with antenna in the form of microstrip line. The piezocrystal itself is located in an additional metal housing to protect it against environmental influences (reproduced by permission of Siemens AG, ZT KM, Munich)

[1B] an antenna formed on the substrate:

253. Bowers discloses element [1B]. Bowers discloses an RFID tag that

includes an IC with “programmable memory ... for storing bits of identification data” is “electrically connected” to “a coil antenna.” Ex-1009, 4:59-5:12. Tilson also discloses “a passive type RFID tag” that includes “an antenna 116 coupled to a non-volatile memory 118....” Ex-1010, 3:47-50.

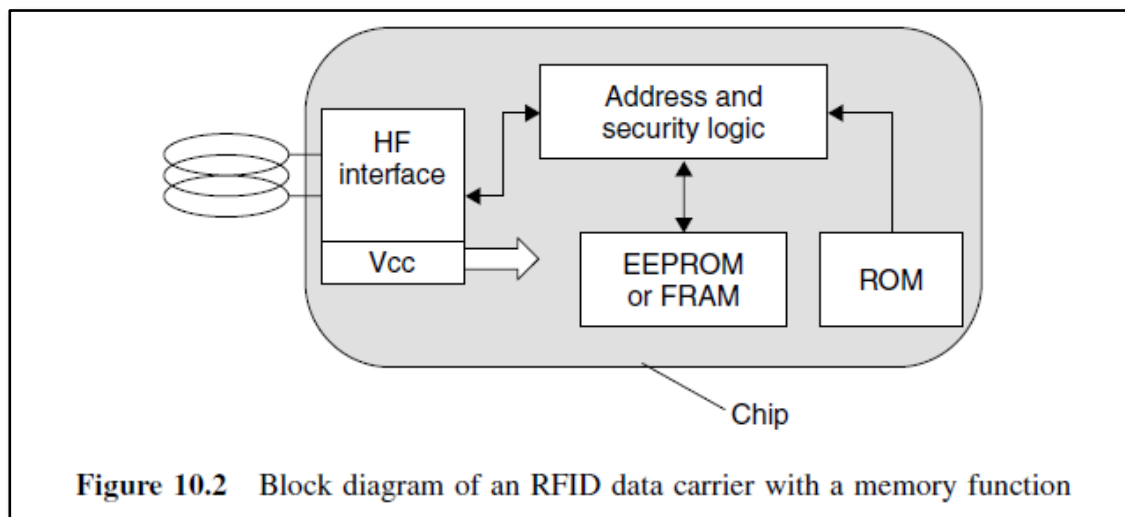
254. Moreover, a POSITA would have known that a substrate provides the physical support for the components of an RFID tag, such as the IC and antenna, and that an antenna of an RFID tag is formed on the substrate was commonly found in TDS-compliant RFID tags. *See supra* [1A].

[1C] and an RFID integrated circuit chip which is electrically coupled to the antenna structure:

255. Bowers discloses [1C]. Bowers states that “IC 18 is electrically connected to the resonant circuit 12,” which includes a coil antenna. Ex-1009, 4:59-5:12. Moreover, Tilson discloses that “the RFID tag includes an antenna 116 coupled to a non-volatile memory.” Ex-1010, 3:47-50. Tilson discloses that “the RFID tag includes an antenna 116 coupled to a non-volatile memory.” Ex-1010, 3:47-50. A POSITA would have understood that the “non-volatile memory” is an RFID integrated circuit chip and that the antenna of the RFID is connected to it. Ex-1017, 73 (“The voltage u_2 induced in the transponder coil is used to provide the power supply to the data memory (microchip) of a passive transponder.”) (italics original).

256. A POSITA would have known that it was well-established in the art that IC chips are fundamental components of RFID tags, serving as the memory unit that stores information such as the EPC. *See supra* § V.

257. Furthermore, a POSITA would have understood that the antenna must be electrically coupled to the RFID IC chip to facilitate communication between the tag and an RFID interrogator. The antenna transmits information stored in the IC and receives signals. For the antenna to communicate the stored information, the RFID IC chip must be electrically connected to the antenna so as to provide it the information to transmit, or to store or analyze the information sent to the RFID tag's antenna. Ex-1017, 274 (“Passive transponder, i.e. transponders that do not have their own power supply, are supplied with energy via the HF field of the reader. To achieve this, the HF interface draws current from the transponder antenna, which is rectified and supplied to the chip as a regulated supply voltage.”); *see also id.*, Figure 10.2.



[1D] wherein the RFID integrated circuit chip is encoded with a unique object number, the unique object number comprising an object class information space and a unique serial number space,

258. TDS discloses element [1D]. Bowers and Tilson each disclose a serial number space, but do not disclose an object class information space. Ex-1009, 6:39-41; Ex-1010, 7:62-63.

259. First, TDS teaches encoding a unique object number comprising an object class information space and unique serial number space: “In the Gen 2 Specification, the tag memory is separated into four distinct banks, each of which may comprise one or more memory words, where each word is 16 bits long. These memory banks are described as ‘Reserved’, ‘EPC’, ‘TID’ and ‘User’ . . . the ‘EPC’ memory bank contains data used for identifying the object to which the tag is or will be attached.” Ex-1011, § 3.2. TDS discloses that EPC encoding scheme SGTIN-96 is comprised of a GTIN and a serial number uniquely identifying the tagged item

within the GTIN. *Id.*, § 3.5. TDS teaches that “the combination of GTIN and a unique serial number” is a “unique object identifier.” *Id.*

260. TDS also teaches that “a GTIN identifies a particular class of object, such as a particular kind of product or SKU.” Ex-1011, § 2.1.2.1. The GTIN—i.e., the object class information space—consists of the following information elements: “Company Prefix, assigned by GS1 to a managing entity,” and “[t]he Item Reference, assigned by the managing entity to a particular object class.” *Id.* The serial number—i.e., the unique serial number space—is “assigned by the managing entity to an individual object.” *Id.* Note that the figure below illustrates the data fields of an SGTIN-96 number, including a company prefix, an item reference, a partition value, and a filter value. Importantly, the serial number space is defined as a 38-bit sequence of binary values (0s or 1s); moreover, the “Max. decimal value” shown in the figure is the number of items the 38-bit serial number can identify. *See generally Id.*, § 2.1.2.1.

	Header	Filter Value	Partition	Company Prefix	Item Reference	Serial Number
SGTIN-96	8	3	3	20-40	24-4	38
	0011 0000 (Binary value)	(Refer to Table 5 for values)	(Refer to Table 6 for values)	999,999 – 999,999,999,999 (Max. decimal range*)	9,999,999 – 9 (Max. decimal range*)	274,877,906,943 (Max. decimal value)

*Max. decimal value range of Company Prefix and Item Reference fields vary according to the contents of the Partition field.

Table 4. The EPC SGTIN-96 bit allocation, header, and maximum decimal values.

261. Moreover, Bowers discloses tag information like the TDS object class identification space. As such, the TDS format was a readily understood means for Bowers' RFID tags. Ex-1009, 6:36-38 (“...other information 208 is also stored in the tag”); 6:61-64 (“Article identification information 212 is a description of the article ... (e.g., e.g., SONY 19" TV, Model No. KV1926....)”). Likewise, Tilson discloses the use of a “G2 DogBone model of RFID tag,” which was known to be a Gen2/TDS compliant tag. Ex-1010, 3:52-55.

262. Thus, TDS discloses element [1D].

[1E] wherein the unique serial number space is encoded with one serial number instance from an allocated block of serial numbers, the allocated block being assigned a limited number of most significant bits, and

263. Bowers in view of Tilson and TDS renders obvious [1E]. A POSITA would have understood the term “an allocated block of serial numbers” to mean “a pre-authorized range of binary serial numbers.” *See supra* § V.A.1. Bowers discloses allocating a block of serial numbers. Tilson teaches a way of avoiding such duplication by partitioning a binary serial number space based on a limited number of predetermined MSBs. TDS discloses the overall data structure, including the object class information space and unique serial number space..

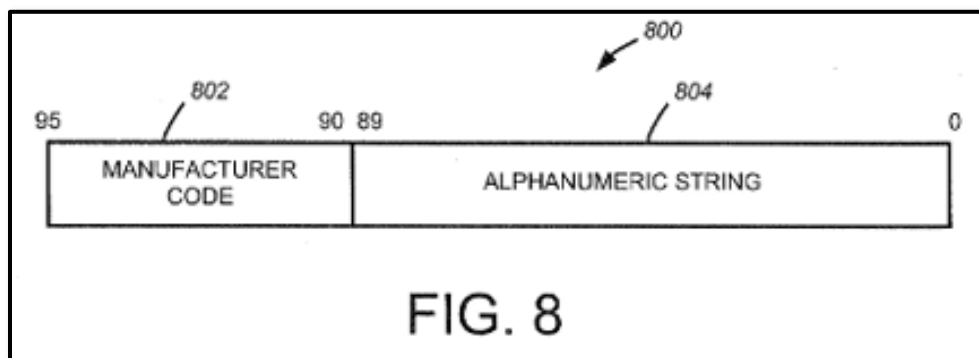
264. First, TDS discloses a unique serial number space. *See* [1D]. Moreover, a POSITA would have been familiar with implementing that standard. *Id.*

265. Second, Bowers discloses allocating blocks of serial numbers. Bowers

discloses encoding the unique serial number space with one serial number instance from a set (i.e., block or range) of serial numbers, wherein “a predefined set of tags 10 are prepared to be applied to a predefined set of articles (not shown), unique tag information is assigned to each tag 10 and the tags are programmed accordingly,” and “one tag 10 is associated with each article.” Ex-1009, 10:47-55. This disclosed “predefined set of tags” with “unique tag information” for each tag is a pre-authorized range of serial numbers.

266. Third, Tilson discloses a unique serial number space being assigned a limited number of MSBs. Tilson teaches that “the data structure 800 is organized to store a serial number. In the preferred embodiment, the serial number has a manufacturer code portion 802 and an alphabetic, numeric or alphanumeric portion 804, which together form the unique serial number for the respective concrete component.” Ex-1010, 7:65-8:5; *see also* figure below. *Id.* Importantly, Tilson teaches that the alphanumeric string is composed of human-readable symbols displayed on a screen by a software and represented and stored in RFID tag as binary values. A POSITA understood that these two forms—the human-readable string and the binary representation—are equivalent. Put differently, *the alphanumeric string and its binary representation are not two different values or information; rather, they are equivalent and represent the same information.* Moreover, a POSITA would have understood that the information saved in the RFID tag is actually in binary

form, and that it is so in all cases. *See supra* § VII.D.



267. Tilson also teaches that one “skilled in the art will understand that the data structure 800 may be modified to carry additional information. For example, it may be configured to carry a name or identifier of manufacturer of the concrete component, a name or identifier of the entity that supplied the concrete used to make the component, [or] an identifier of the concrete batch used to make the concrete component” *Id.*, 8:32-41.

268. Further still, a POSITA understood that partitioning the serial number space to create a prefix to a set of subserial numbers was an existing practice in the RFID field. For example, it was known that an “RFID tag offers enough space to hide the month and the year of expiry of the drugs.” Ex-1021, 2865.

269. A POSITA would have understood that Tilson explicitly discloses a bit level partition of the serial number space. “For example, bits 90-95 of a 96-bit structure may store the manufacturer code, and bits 0-89 may store the alphanumeric portion.” Ex-1010, 8:3-5. In other words, Tilson partitions the 96-bit serial number

at the bit level, and that each bit-field is in a discrete memory unit that stores binary values.

270. Tilson thus teaches the *concept* of partitioning the serial number space itself into two portions. And this partitioning allocates the leading end (i.e., MSBs) to encoding specific information about the item. For example, the prefix could denote the batch ID of the tagged article. A POSITA knew that existing partitioning methods (such as subnetting and computer memory allocation) assign prefixes to a specific allocated block. *See supra* §§ V.A.1, 2. In particular, a POSITA would have understood that an RFID tag's memory is conceptually the same as a computer's memory: storing serial numbers in RFID tags is analogous to storing data in a computer; and both are done in binary. *Id.* The memory address space of a digital computer is typically partitioned into MSBs, which create or define the blocks of memory addresses, and LSBs, which provide offsets within those blocks to represent a memory address instance. *Id.*

271. A POSITA also understood that computer memory allocation and IP addressing methods use “most significant bits” within analogous “serial number spaces” to force uniqueness. For example, a POSITA would have recognized that each RFID tag can be viewed as an “offset” within a larger “memory block” encompassing all tags in the system, and that allocating serial numbers to these tags is essentially assigning unique memory addresses within the RFID system's overall

address space. *See supra* § V.A.2. In the same way that memory addresses use MSBs to define larger memory blocks, a POSITA would have known that serial numbers can also use MSBs to define allocated blocks assigned to different production lines, facilities, or other categories. *See supra* § V.A.4. A POSITA thus understood that the manufacturer's code (or batch ID, etc.) is a "limited, predefined sequence of higher order bits at the leading end" of the serial number—i.e., the MSBs of the tag's 90-95 bit field serve to categorize items based on certain attributes. Indeed, a POSITA understood subdividing serial numbers, as evidenced by *Dummies*, which teaches that:

A range of serial numbers for each product is allocated to each manufacturing facility. Within a facility, a range of numbers from those allocated to the facility is allocated to each line producing a particular product. In this way, ***the serial number is effectively subdivided*** into a facility number, line number, and subserial number, in which the allocation hierarchy is maintained between facility number and line number.

Ex-1013, 240.

272. Finally, POSITA would have been motivated to use Tilson's partitioning of serial numbers in Bowers to provide Bowers' goal of avoiding duplication of serial of serial numbers across entities. *See supra* § VIII.D.1.

273. Accordingly, a POSITA understood that Bowers and Tilson in view of TDS teaches *wherein the unique serial number space is encoded with one serial*

number instance from an allocated block of serial numbers [a preauthorized range of binary serial numbers], the allocated block being assigned a limited number of most significant bits.

[1F] wherein the unique serial number space comprises the limited number of most significant bits uniquely corresponding to the limited number of most significant bits of the allocated block and of remaining bits of lesser significance that together comprise the one serial number instance.

274. Bowers in view of Tilson and TDS renders obvious [1F]. Bowers, Tilson, and TDS disclose a “unique serial number space.” *See* [1D]. Bowers, Tilson, and TDS also disclose the unique serial number space being encoded with one serial number instance from an allocated block of serial numbers. *See* [1E].

275. First, a POSITA would have understood that the MSBs can function as a categorical prefix identifying a higher-level category or class. Ex-1029, §7.4, Lines 952-54. As such, a POSITA understood that Tilson discloses a partition in that serial number’s data structure, where the MSBs of the serial number represent a sub-classification that “function[s] as an additional data field within the serial number space that uniquely identifies the allocated block from which it came.” Ex-1032, 11. The use of MSBs as such was not unfamiliar to a POSITA; recall that the MIT Auto-ID Center studied IP addressing in its effort to design an EPC standard. To a POSITA then, IP addressing provided a model for using MSBs of a “serial number space” to identify a sub-class, such as batch ID, within a broader category, such as Item

Reference “Nike shirts.” The figure below demonstrates the use of the MSBs in an IP address’s last octet, which is analogous to the serial number space in that it specifies an individual host, to define the subnet. The highlighted numbers are the MSBs of the last octet that represent the new subnet—i.e., a sector of the network ID identified by the first three octets. These MSBs define the subnet to which the host belongs, and the host instance is identified by the remaining bits—i.e., the LSBs—in that last octet. In this way, an organization assigning IP addresses can ensure uniqueness among the hosts. Furthermore, this allocation method makes it easier to commission IP addresses because the organization can do it locally, without repeatedly resorting to another authority whenever it needs to assign an IP address.

See supra § V.A.1.

```
Subnet #2: 11000001.00000001.00000001.010 00000 = 193.1.1.64/27
Host #1: 11000001.00000001.00000001.010 00001 = 193.1.1.65/27
Host #2: 11000001.00000001.00000001.010 00010 = 193.1.1.66/27
Host #3: 11000001.00000001.00000001.010 00011 = 193.1.1.67/27
Host #4: 11000001.00000001.00000001.010 00100 = 193.1.1.68/27
Host #5: 11000001.00000001.00000001.010 00101 = 193.1.1.69/27
.
.
Host #15: 11000001.00000001.00000001.010 01111 = 193.1.1.79/27
Host #16: 11000001.00000001.00000001.010 10000 = 193.1.1.80/27
.
.
Host #27: 11000001.00000001.00000001.010 11011 = 193.1.1.91/27
Host #28: 11000001.00000001.00000001.010 11100 = 193.1.1.92/27
Host #29: 11000001.00000001.00000001.010 11101 = 193.1.1.93/27
Host #30: 11000001.00000001.00000001.010 11110 = 193.1.1.94/27
```

Ex-1019, 13.

276. Accordingly, a POSITA understood that Tilson in view of Bowers necessarily includes a “one-to-one correspondence” between the MSBs of the unique serial number space and the range of serial numbers within that allocated block. This is because Tilson teaches the use of MSBs of a serial number to identify its batch, thereby creating a hierarchical structure in which blocks of subserial numbers are defined by their batch ID. A POSITA understood that in hierarchical structures, a defined field at the leading end of a sequence of numbers (i.e., the MSBs) functions as a categorical prefix. This prefix identifies a higher-level category or class (e.g., batch ID) for the subserial number associated with an instance of the tag. *See* [1E]; Ex-1029, § 7.4, Lines 952-54. As such, a POSITA would have been motivated to ensure that the MSBs-prefix of the unique serial number space (bits 90-95, *see supra* [1E]) uniquely correspond to the range of serial numbers within the allocated block of Tilson in view of Bowers.

277. As such, Bowers teaches attaching “a predefined set of tags 10” to “a predefined set of articles” where “unique tag information is assigned to each tag 10 within the tag set and each of the tags are programmed accordingly....” Ex-1009, 6:14-21. Bowers further teaches the importance of avoiding duplication of tag information across entities, such as different stores or chains of stores in a retail setting. Using Tilson’s entity identifier encoded in a serial number’s MSBs of serial

numbers used at a particular store in Bowers, as discussed above, achieves this goal. *See supra* § VIII.D.1. This results in partitioning the unique serial numbers space at the bit level as recited by the “uniquely corresponding to” limitation. *See supra* § VI.2.

278. Therefore, Bowers in view of Tilson and TDS discloses [1F].

279. Therefore, Tilson in view of Bowers and TDS render obvious claim 1.

3. Claim 2: The RFID transponder of claim 1, wherein the object class information space is encoded with information comprising a company prefix, an item reference, a partition value, and a filter value.

280. Bowers in view of Tilson and TDS discloses claim 1, and TDS discloses the additional limitations of claim 2. *See* Claim 1. TDS discloses the SGTIN-96 encoding scheme, which allocates fields within the object class information space for company prefix, item reference, partition, and filter values. Ex-1011, § 2.1.2.1.

281. The figure below illustrates the data fields of an SGTIN-96 number, including a company prefix, an item reference, a partition value, and a filter value. Importantly, the serial number space is defined as a 38-bit sequence of binary values (0s or 1s); moreover, the “Max. decimal value” shown in the figure is the number of items the 38-bit serial number can identify. *See generally id.*

	Header	Filter Value	Partition	Company Prefix	Item Reference	Serial Number
SGTIN-96	8	3	3	20-40	24-4	38
	0011 0000 (Binary value)	(Refer to Table 5 for values)	(Refer to Table 6 for values)	999,999 – 999,999,999,999 (Max. decimal range*)	9,999,999 – 9 (Max. decimal range*)	274,877,906,943 (Max. decimal value)

*Max. decimal value range of Company Prefix and Item Reference fields vary according to the contents of the Partition field.

Table 4. The EPC SGTIN-96 bit allocation, header, and maximum decimal values.

Id.

4. **Claim 3: The RFID transponder of claim 1, wherein the unique object number is a portion of a serialized global trade item number.**

282. Bowers in view of Tilson and TDS discloses claim 1, and TDS discloses the additional limitations of claim 3. *See* Claim 1. TDS discloses the SGTIN coding scheme, which incorporates unique serial numbers alongside GTINs. Ex-1011, § 2.1.2.1. A POSITA understood that the serialized GTIN includes the unique object number. *Id.*

5. **Claim 4: The RFID transponder of claim 3, wherein the serialized global trade item number is an EPC SGTIN-96 encoding.**

283. Bowers in view of Tilson and TDS discloses claim 1, and TDS discloses the additional limitations of claim 4. *See* Claim 1. TDS discloses encoding EPCs using SGTIN-96. Ex-1011, § 2.1.2.1.

6. Claim 5: The RFID transponder of claim 1, wherein the unique serial number space is at least 38 bits.

284. Bowers in view of Tilson and TDS discloses claim 1, and TDS discloses the additional limitations of claim 5. *See* Claim 1. For example, TDS discloses that the serial number in SGTIN-96 is 38 bits, as shown in the figure below. Ex-1011, § 2.1.2.1.

285. The figure below illustrates the data fields of an SGTIN-96 number, including a company prefix, an item reference, a partition value, and a filter value. Importantly, the serial number space is defined as a 38-bit sequence of binary values (0s or 1s); moreover, the “Max. decimal value” shown in the figure is the number of items the 38-bit serial number can identify.

	Header	Filter Value	Partition	Company Prefix	Item Reference	Serial Number
SGTIN-96	8	3	3	20-40	24-4	38
	0011 0000 (Binary value)	(Refer to Table 5 for values)	(Refer to Table 6 for values)	999,999 – 999,999,999,999 (Max. decimal range*)	9,999,999 – 9 (Max. decimal range*)	274,877,906,943 (Max. decimal value)

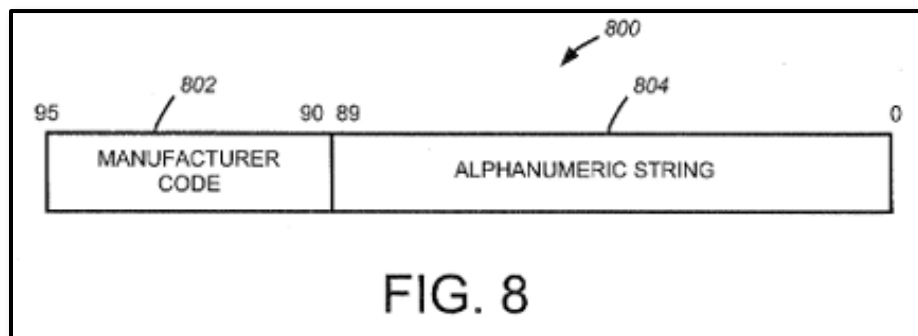
*Max. decimal value range of Company Prefix and Item Reference fields vary according to the contents of the Partition field.

Table 4. The EPC SGTIN-96 bit allocation, header, and maximum decimal values.

Ex-1011, § 2.1.2.1.

7. Claim 6: The RFID transponder of claim 1, wherein the allocated block is defined by at least 3 most significant bits.

286. Bowers in view of Tilson and TDS discloses claim 1, and Tilson discloses the additional limitations of claim 6. *See* Claim 1. Tilson discloses data structures for serial numbers that include high-order bits that indicate the item's manufacturer, project, or other attributes, as depicted in Fig. 8 below. Importantly, Tilson teaches that the alphanumeric string is composed of human-readable symbols displayed on a screen by a software and represented and stored in RFID tag as binary values. A POSITA understood that these two forms—the human-readable string and the binary representation—are equivalent. Put differently, *the alphanumeric string and its binary representation are not two different values or information; rather, they are equivalent and represent the same information*. Moreover, a POSITA would have understood that the information saved in the RFID tag is actually in binary form, and that it is so in all cases. *See supra* § VII.D.



287. The bits of the manufacturer code are the MSBs of the serial number, and Tilson teaches an example where “bits 90-95 of a 96-bit data structure may store

the manufacturer code, and bits 0-89 may store the alphanumeric portion.” Ex-1010, 8:3-5. As such, the manufacturer code in this example is 6 bits long, and since the manufacturer code defines the allocated block, the allocated block is defined by 6 most significant bits, which is *at least 3 most significant bits*.

288. Therefore, Tilson discloses the additional limitations of claim 6.

8. Claim 7: The RFID transponder of claim 1, wherein the allocated block is one of a plurality of allocations intermittently received by an encoder.

289. Bowers in view of Tilson and TDS discloses claim 1, and Bowers discloses the additional limitations of claim 7. *See* Claim 1. Bowers teaches that a predefined set of tags are assigned one serial number per tag, and that the set of tags are to be applied to a predefined set of articles. The predefined set of tags, when storing serial numbers in their memory, are effectively a predefined set of serial numbers, or a block of serial numbers. These serial numbers are then associated with an article from the predefined set of articles. Once all the articles have been tagged, and a new set of articles that require tagging has been defined, Bowers again, at a different time, allocates serial numbers to a predefined set of tags. *See supra* § VII.C. Because these assignments are made at different times with undefined intervals, the encoders, by definition, receive the allocations intermittently.

9. Claim 8: The RFID transponder of claim 1, wherein the RFID integrated circuit chip is comprised of silicon.

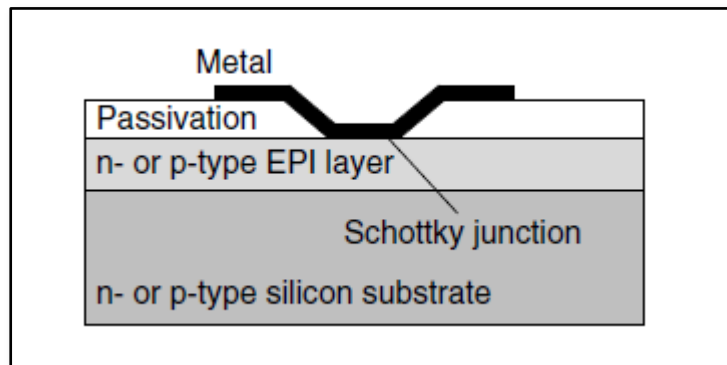
290. Bowers in view of Tilson and TDS discloses claim 1, and Tilson teaches the additional limitations of claim 8. *See* Claim 1. Tilson disclose RFID integrated circuits, *see* [1C], and a POSITA knew that IC chips were commonly made from silicon. Ex-1017, 341 (“A significant improvement was achieved when entire microprocessors were successfully integrated into a silicon chip, and these were chips incorporated into an identification card.”).

10. Claim 11: The RFID transponder of claim 1, wherein the RFID integrated circuit chip is disposed on the substrate.

291. Bowers in view of Tilson and TDS discloses claim 1, and Tilson teaches the additional limitations of claim 11. *See* Claim 1. Tilson discloses RFID tags with integrated circuit chips attached to a substrate. *See* Claim [1A]-[1C].

292. A POSITA would have understood that the disposition of the RFID IC chip on a substrate is a conventional aspect of RFID tag design. For example, “[i]n RFID transponders primarily p-doped Schottky diodes are used, since there are particularly suitable for detectors with no zero bias in small signal operation, i.e. for the conditions that occur in every transponder.” Ex-1017, 134. The figure below shows a layout of a Schottky diode that illustrates the substrate. A POSITA would have understood that the IC chip is disposed on the substrate for the chip to properly function, receive power, and communicate with the other transponder components.

293. Therefore, Tilson discloses the additional limitations of claim 11.



Ex-1017, 134.

- 11. Claim 12: The RFID transponder of claim 1, wherein the limited number of most significant bits that correspond to the allocated block further correspond to an encoder that encodes the RFID transponder.**

294. Bowers in view of Tilson and TDS discloses claim 1, and Tilson and TDS teach the additional limitations of Claim 12. *See* Claim 1. Tilson discloses encoding information into the most significant bits of the serial number, such as a manufacturer's code, i.e., "a name or identifier of manufacturer of the concrete component." Ex-1010, 8:32-41.

295. Tilson teaches that the encoded information in the data structure of the serial number "may be configured to carry a name or identifier of manufacturer of the concrete component, a name or identifier of the entity that supplied the concrete used to make the component, an identifier of the concrete batch used to make the concrete component, a name or identifier of the entity that supplied the

reinforcement element, if any, used in the precast concrete component, a name of the project or structure for which the concrete component is to be used, etc.” *Id.* This indicates that the MSBs are used to encode specific categorical information, such as the status of the tagged article that such encoded information in the data structure of the serial number may be configured to carry.

296. It would have been obvious to a POSITA to adapt the manufacturer information from Tilson, the company prefixes disclosed in TDS, or the combination of the two, to correspond the MSBs to the encoder that encodes the RFID tags.

297. These manufacturers are the encoders, and by specifying that the MSBs store a manufacturer’s code, Tilson discloses the MSBs correspond to the manufacturer-encoder, rendering claim 12 obvious.

12. Independent Claim 13:

[13pre] An RFID transponder comprising:

298. Bowers discloses [13pre] for the same reasons as element [1pre]. *See* [1pre].

[13A] a substrate;

299. Bowers discloses [13A] for the same reasons as element [1A]. *See* [1A].

[13B] an antenna structure formed on the substrate; and

300. Bowers discloses [13B] for the same reasons as element [1B]. *See* [1B].

[13C] an RFID integrated circuit chip which is electrically coupled to the antenna structure,

301. Bowers discloses [13C] for the same reasons as element [1C]. *See* [1C].

[13D] wherein the RFID integrated circuit chip is encoded with a global trade item number including a unique object number, the unique object number comprising an object class information space including a block of at least 50 bits and a unique serial number space including a block of at least 38 bits,

302. TDS discloses [13D]. TDS teaches that SGTIN-96's object class information space is comprised of the filter value, partition, company prefix, and item reference, and that together, they comprise 50 bits. TDS also teaches that the unique serial number space includes a block of at least 38 bits, as depicted in the figure below. Importantly, the serial number space is defined as a 38-bit sequence of binary values (0s or 1s); the "Max. decimal value" shown in the figure is the number of items the 38-bit serial number can identify, not the value, or range of values, of serial numbers.

	Header	Filter Value	Partition	Company Prefix	Item Reference	Serial Number
SGTIN-96	8	3	3	20-40	24-4	38
	0011 0000 (Binary value)	(Refer to Table 5 for values)	(Refer to Table 6 for values)	999,999 – 999,999,999,999 (Max. decimal range*)	9,999,999 – 9 (Max. decimal range*)	274,877,906,943 (Max. decimal value)

*Max. decimal value range of Company Prefix and Item Reference fields vary according to the contents of the Partition field.

Table 4. The EPC SGTIN-96 bit allocation, header, and maximum decimal values.

Ex-1011, § 2.1.2.1.

[13E] wherein the unique serial number space has one serial number instance from an allocated block of serial numbers, the allocated block being assigned a limited number of most significant bits, and

303. Bowers discloses [13E] for the same reasons as element [1pre]. *See* [1E].

[13F] wherein the unique serial number space comprises at least 3 most significant bits uniquely corresponding to the limited number of most significant bits of the allocated block.

304. Bowers discloses [13F] for the same reasons as element [1pre]. *See* [1F].

13. Claim 14: The RFID transponder of claim 13, wherein the unique serial number space further comprises remaining bits of lesser significance that together with the at least 3 most significant bits comprise the one serial number instance.

305. Bowers in view of Tilson and TDS discloses claim 13. *See* Claim 13. Tilson discloses the additional limitations recited in Claim 14 for the same reasons as element [1F] and claim 6. *See* [1F]; claim 6.

14. Claim 15: The RFID transponder of claim 13, wherein the at least 3 most significant bits that correspond to the allocated block further correspond to an encoder that encodes the RFID transponder.

306. Tilson in view of Bowers and TDS disclose claim 13, and Tilson discloses the additional limitation of claim 15, for the same reasons as element [1F] and claim 12.

15. Independent Claim 16

[16pre] An integrated circuit, comprising:

307. Bowers or Bowers in view of Tilson discloses [16pre] for the same reasons as element [1C].

[16A] a radio frequency circuit configured to backscatter a return link signal for processing by an RFID interrogator;

308. Bowers in view of Tilson teaches [16A]. Tilson discloses that “[a] suitable RFID tag for use with present invention includes the Ultra High Frequency (UHF) Rafsec G2 DogBone model of RFID tag....” Ex-1010, 3:52-55. A POSITA understood that “[a]ll long-range systems operate using electromagnetic waves in the UHF and microwave range. The vast majority of such systems are also known as backscatter systems due to their physical operating principle.” Ex-1017, 23. A POSITA understood that the use of backscatter is a one of several “procedures for sending data from the transponder back to the reader....” *Id.*, 13.

309. Although Bowers does not teach a UHF system, a POSITA would still be motivated to combine Tilson and Bowers because the POSITA’s concern here is with the tag’s storage of identification information, independent of the RF technology being used.

310. Therefore, Bowers in view of Tilson discloses element [16A].

[16B] control logic; and

311. Bowers and Tilson teach [16B]. A POSITA would have understood

that, as the '967 Patent admits, RFID transponders typically “include an integrated circuit chip having radio frequency circuits, control logic, memory and an antenna structure mounted on a supporting substrate....” Ex-1001, 1:40-44. A POSITA also understood that the RFID IC chips disclosed in Tilson and Bowers comprised control logic, which is necessary to control its operation to store data and respond to queries from an RF interrogator. Ex-1010, 4:23-24 (“The RFID assembly 100 stores a unique serial number that may be read by an RFID reader.”); *see* Ex-1017, 12 (“In programmable systems, write and read access to the memory and any requests for write and read authorization must be controlled by the data carrier’s internal logic.”).

[16C] a memory,

312. Tilson and TDS disclose [16C]. Tilson teaches that its “RFID tag includes an antenna 116 coupled to a non-volatile memory 120, such as a read only memory (ROM), an erasable programmable ROM (EPROM), an electrically erasable programmable PROM (EEPROM), etc.” Ex-1010, 3:48-52. TDS discloses that Gen2 tags include memory for storing an EPC. Ex-1011, 20:522-528.

313. The '967 Patent confirms that RFID transponders typically “include an integrated circuit chip having radio frequency circuits, control logic, memory and an antenna structure mounted on a supporting substrate....” Ex-1001, 1:40-44.

[16D] wherein the memory is configured with a unique object number space, the unique object number space comprising an object class information space and a unique serial number space,

314. TDS discloses [16D] for the same reasons as element [1D].

[16E] wherein the unique serial number space is encoded with one serial number instance from an allocated block of serial numbers, the allocated block being assigned a limited number of most significant bits, and

315. Bowers in view of Tilson and TDS discloses [16E] for the same reasons as element [1E]. *See* [1E].

[16F] wherein the unique serial number space comprises the limited number of most significant bits uniquely corresponding to the limited number of most significant bits of the allocated block and remaining bits of lesser significance that together comprise the one serial number instance.

316. Bowers in view of Tilson and TDS discloses [16F] for the same reasons as element [1F]. *See* [1F].

317. Accordingly, Bowers in view of Tilson and TDS discloses claim 16.

16. Claim 17: The integrated circuit of claim 16, wherein the object class information space is encoded with information comprising a company prefix, an item reference, a partition value, and a filter value.

318. Bowers in view of Tilson and TDS discloses Claim 16. *See* Claim 16.

Tilson in view of TDS discloses the additional limitations recited in Claim 17 for the same reasons as Claim 2.

- 17. Claim 18: The integrated circuit of claim 16, wherein the unique object number is a portion of a serialized global trade item number.**

319. Bowers in view of Tilson and TDS discloses Claim 16. *See* Claim 16.

TDS discloses claim 19 for the same reasons as claim 3.

- 18. Claim 19: The integrated circuit of claim 18, wherein the serialized global trade item number is an EPC SGTIN-96 encoding.**

320. Bowers in view of Tilson and TDS discloses Claim 16. *See* Claim 16.

TDS discloses claim 20 for the same reasons as claim 4.

- 19. Claim 20: The integrated circuit of claim 16, wherein the limited number of most significant bits that correspond to the allocated block further correspond to an encoder that encodes the memory of the integrated circuit.**

321. Bowers in view of Tilson and TDS discloses Claim 16. *See* Claim 16.

The additional limitations recited in Claim 20 are also rendered obvious, for the same reasons as Claim 12.

E. Ground 2B: Claim 9

- 1. Claim 9: The RFID transponder of claim 1, wherein the RFID integrated circuit chip is comprised of polymer.**

322. Bowers in view of Tilson and TDS discloses claim 1, and Nitzan discloses the additional limitations in Claim 9. Nitzan is analogous art, as it discloses “an integrated circuit (“IC”), which is arranged to store a code including information....” Ex-1022, 7:58-60. “Generally speaking, the code comprises any

information that is to be transmitted from transponder 28 to reader 32. For example, the information may comprise an ID number that identifies the package 24.” *Id.*, 8:67-9:3. “

323. Nitzan discloses that “IC 56 may comprise an organic polymer electronic chip, as known in the art.” Ex-1022, 25:64-65. The ’967 Patent confirms that it was common knowledge that RFID IC chips can be comprised of polymer. The ’967 Patent recognizes that “combining the benefits of the latest technology in RFID transponders with sensing devices, a broader class of devices called wireless sensors is emerging.” Ex-1022, 1:65-67. Those “wireless sensors” are “made using silicon circuits, polymer circuits....” *Id.*, 2:11. Therefore, a POSITA would have understood that the RFID integrated circuit chip of IBM200 could be comprised of polymer.

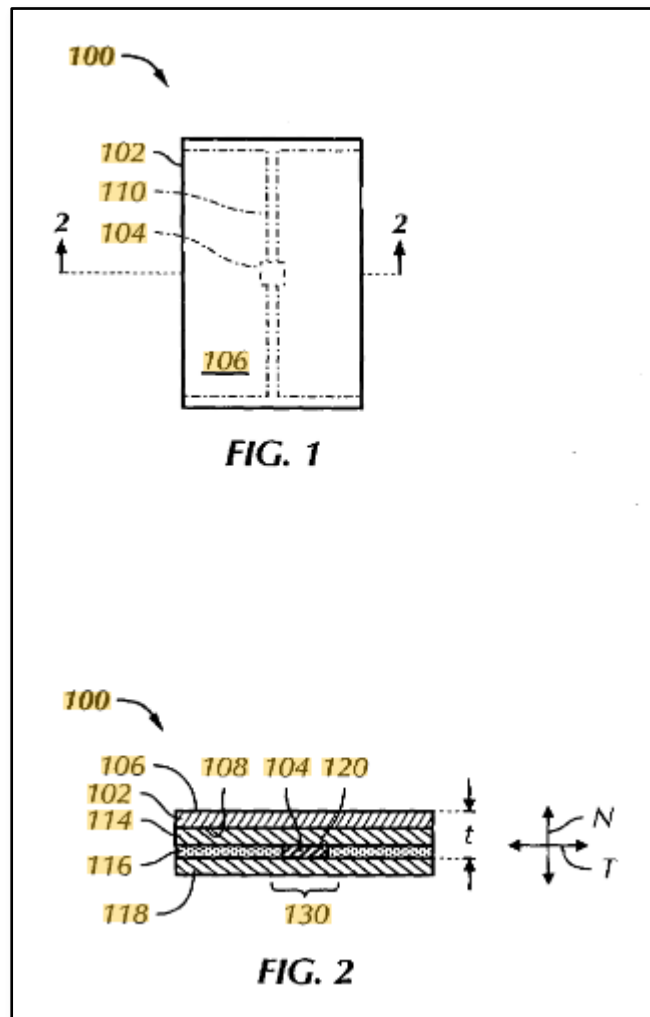
324. Therefore, Nitzan discloses the additional limitations of claim 9, and claim 9 is obvious over Bowers in view of Tilson, TDS, and Nitzan.

F. Ground 2C: Claim 10

1. Claim 10: The RFID transponder of claim 1, wherein the substrate, antenna structure, and RFID integrated circuit chip are covered by a face stock material.

325. Bowers in view of Tilson and TDS discloses claim 1, and Forster discloses the additional limitations in Claim 10. Bowers and Tilson disclose a substrate, antenna structure, and RFID integrated circuit chip. See [1A]; [1B]; [1C].

326. Forster discloses “a radio-frequency identification (RFID) tag 100 includes a face stock 102 and an RFID device 104. The face stock 102 may be described as having a printable side or surface 106 and an inlay side or surface 108. The RFID device 104 is mounted to or disposed on the inlay side 108 of the face stock 102 and may include any type of RFID structure, including chips, straps, power source (for active devices), and connecting structure for coupling with an antenna 110, as known in the art...The face stock 102 may include an inlay substrate 114....”
Ex-1023, 2:54-63; *see also* Figures 1 and 2 below.



327. Therefore, Forster discloses the additional limitations of claim 10, and claim 10 is obvious over Bowers in view of Tilson, TDS, and Forster.