Published February 1, 2003. Distribution restricted to Sponsors until May 1, 2003.

TECHNICAL REPORT

EPC-256: The 256-bit Electronic Product Code™ Representation

Daniel Engels

AUTO-ID CENTER MASSACHUSETTS INSTITUTE OF TECHNOLOGY, 400 TECHNOLOGY SQUARE, SIXTH FLOOR, CAMBRIDGE, MA 02139-4307, USA

ABSTRACT

The Electronic Product Code™ is a naming and identification scheme that was designed to uniquely identify all objects, whether physical or virtual. However, the 96-bit and 64-bit representations of the Electronic Product Code™, the EPC-96 and EPC-64 respectively, have practical limitations on the number of objects, services, assemblies, and groupings that may be uniquely identified. These limitations arise due to the limited number of bits that are allocated to represent the various components of the Electronic Product Code™. We present a 256-bit representation of the Electronic Product Code™, the EPC-256, that does not suffer from these practical limitations.

Published February 1, 2003. Distribution restricted to Sponsors until May 1, 2003.

TECHNICAL REPORT

EPC-256: The 256-bit Electronic Product Code™ Representation

Biography

Daniel W. Engels
Associate Director

Daniel W. Engels received his B.S. from the University of Buffalo, his M.S. from the University of California, Berkeley, and his Ph.D. from the Massachusetts Institute of Technology all in Electrical Engineering and Computer Science. His master's thesis is in the area of computer-aided design for electronic systems, and his doctoral thesis is in the field of theoretical computer science. Dr. Engels joined the Auto-ID Center after obtaining his doctoral degree where he leads the day-to-day research activities of the Center.
Dr. Engels' research interests include scheduling theory and applications, real-time system design, distributed and mobile computing, and computer-aided design for embedded systems.

Published February 1, 2003. Distribution restricted to Sponsors until May 1, 2003.

TECHNICAL REPORT

EPC-256: The 256-bit Electronic Product Code™ Representation

Contents

1. INTRODUCTION

The Electronic Product Code™ is a naming and identification scheme designed to enable the unique identification of all objects, whether physical or virtual. An Electronic Product Code™ uniquely identifies an object with the combination of three unique numbers: Domain Manager number, Object Class number, and Serial Number. A representation of an Electronic Product Code™ is referred to as an EPC™, and 96-bit and 64-bit representations have been defined previously [2, 1]. While there is no limitation on the number of objects that may be identified with an Electronic Product Code™, specific EPC™ representations do enforce limitations on how many Electronic Product Codes™ may be represented by that specific EPC™ version.

The EPC-96 and EPC-64 were designed for the short term use of the Electronic Product Code™ as a physical object identifier. The EPC-96 and EPC-64 versions were developed with the knowledge that the limitations imposed by these representations may make them insufficient for the long term use of the Electronic Product Code™ as a universal identification scheme. Larger EPC™ representations have always been expected and planned for with the use of a Version Number in the EPC™ representation.

The EPC-96 and EPC-64 have limitations that arise due to the limited number of bits that are allocated to represent the three component numbers of the Electronic Product Code™. Of particular practical importance is the number of bits allocated to represent the Domain Manager numbers of an EPC™. The EPC-64 provides up to 26 bits to represent Domain Manager numbers (allowing up to 67,108,864 Domain Managers) with each Domain Manager able to uniquely identify more than 6.87×10^{10} objects. The EPC-96 provides 28 bits to represent Domain Manager numbers (allowing up to 268,435,456 Domain Managers) with each Domain Manager able to uniquely identify more than 1.15×10^{18} objects.

While the number of possible Domain Managers and the number of items uniquely identifiable per Domain Manager with the EPC-64 and EPC-96 are sufficient over at least the medium term use of the Electronic Product Code™, particularly as a physical object identifier, the number of unique Domain Manager numbers may not be sufficient for the long term use of the Electronic Product Code™ as a universal identification scheme. The use of the Electronic Product Code™ to uniquely identify non-objects, such as services and aggregations, will accelerate the usage of the Electronic Product Code™ namespace and further exacerbate problems arising from the limitations of the EPC-96 and EPC-64 representations. Therefore, we present a 256-bit representation of the Electronic Product Code™, the EPC-256, that provides up to 128 bits to represent Domain Manager numbers (able to represent more than 3.40×10^{38} Domain Manager numbers) with each Domain Manager able to uniquely identify more than 1.32×10^{36} objects. Consequently, the EPC-256 is designed to be usable over the long term use of the Electronic Product Code™ as a universal identification scheme, not just a physical object identification scheme.

2. THE ELECTRONIC PRODUCT CODE™ AND THE EPC™

The Electronic Product Code™ is a unique identification scheme consisting of three partitions: Domain Manager, Object Class, and Serial Number. The Domain Manager number identifies the entity responsible for allocating and maintaining the Object Class numbers and Serial Numbers for that domain. The owner of a specific Domain Manager number is also responsible for ensuring the reliable operation of the Object Name Service (ONS) for that domain and for maintaining and publishing associated documents [3]. The Object Class number identifies a logical grouping of objects. In the identification of manufactured goods, the Object Class number may be considered to be the product identifier or Stock Keeping Unit (SKU) [4], that is, it identifies a group of products that have the same characteristics (as defined by that Domain Manager). The Object Class number may also be used for any other object grouping scheme developed by the Domain Manager. The Serial Number encodes a unique identification number for an object or logical entity identified by a Domain Manager as part of a specific Object Class. The combination of Domain Manager number, Object Class number, and Serial Number uniquely identifies an object.

An EPC™, a representation of an Electronic Product Code™, encodes the three numbers in the final three of its four internal partitions. The first partition is a Version Number that indicates the number of bits contained in each of the remaining three partitions (in order following the Version Number): Domain Manager, Object Class, and Serial Number.

3. DESIGN STRATEGY

We consider the design strategy for a large bit length EPC™ identification number. We include underlying assumptions, theoretical constraints, and practical implementation constraints that lead to a particular design approach.

3.1. Object Identification

The larger EPC™ identification versions defined here serve the same function as the 96-bit and 64-bit EPCs™ previously defined. Therefore, the same reasoning and assumptions that went into the design of these EPCs™ applies to a larger EPC™ representation [2, 1]. Thus, the larger EPC™ versions must contain the same four partitions as the EPC-64 and EPC-96 versions: Version Number, Domain Manager, Object Class, and Serial Number.

The Electronic Product Code™ was designed to uniquely identify objects, both physical objects and virtual objects. The EPC-96 and EPC-64 were designed to enable the inexpensive identification of physical objects. A larger EPC™ version may be more expensive to use for identifying physical objects but will have little or no cost impact on the identification of virtual objects.

Since a larger identification version enables more flexibility in the partition sizes, a variable partitioning scheme should at least be considered. The disadvantages of variable partition schemes – more complex descriptions, parsers, routers, and software – should, however, limit the extent of partition variations.

3.2. Superset of Smaller EPCs

An EPC™ of a specific length should be able to represent a proper superset of the Electronic Product Codes™ representable by all smaller length EPC™ versions. In this way, all identifiers representable by the smaller EPC™ versions can be represented by the larger EPC™ version.

3.3. Version Number Partition

The Version Number must be at least 8 bits in length to accommodate the 8-bit Version Numbers used in the EPC-96 and the 2-bit Version Numbers used in the EPC-64. The EPC-96 does not utilize all possible Version Numbers; therefore, an 8 bit Version Number for the EPC-256 is possible. The maximum size of the Version Number partition is limited by the practical number of versions expected. A size of 8 bits provides for up to 256 different versions to be shared with all smaller versions. The limited number of EPC™ versions being defined thus far combined with the practical need to minimize the number of versions, allows that 8 bits should be sufficient for the Version Number partition.

3.4. Domain Manager Partition

The Domain Manager partition should be at least 28 bits in length, the maximum length available in the EPC-96 and EPC-64 versions. The maximum length of the Domain Manager partition should be large enough to enable the representation of enough Domain Managers over the long term use of the Electronic Product Code™ while still enabling each Domain Manager to uniquely identify a large number of objects. We consider that 128 bits enables the representation of more than 3.40 x 10^{38} items, which is nearly equal to the number of molecules on the surface of the earth. Therefore, a 128-bit Domain Manager partition would provide for the long term use of the Electronic Product Code™,

3.5. Object Class Partition

As with the Domain Manager partition, the Object Class partition must provide for at least the capacity of the largest EPC-96 and EPC-64 version. Therefore, the Object Class partition must be at least 24 bits, the number of bits defined in the EPC-96 Type I version. This Object Class length is sufficient to identify more types of objects than are currently identified in widely used identification schemes. Therefore, a larger Object Class partition is not required to meet current uses; however, a larger Object Class partition should enable future applications and identification methodologies applied by a particular Domain Manager.

3.6. Serial Number Partition

As with the Domain Manager partition and the Object Class partition, the Serial Number partition must provide for at least the capacity of the largest EPC-96 and EPC-64 version. Therefore, the Serial Number partition must be at least 36 bits, the number of bits defined in the EPC-96 Type I version.

3.7. Multiple Versions

The 256-bit EPC™ is designed to meet the future needs of the applications using the Electronic Product Code™. The exact needs of the future applications are not concretely known at this point in time; therefore, the 256-bit EPC™ versions must be flexible enough so as not to limit future applications. Multiple versions provide this flexibility. By defining multiple versions now, certainty is provided on the structure of at least some of the EPC-256 versions.

Table 1: EPC-256 Type I, Type II, and Type III bit allocations.

		VERSION NUMBER	DOMAIN MANAGER	OBJECT CLASS	SERIAL NUMBER
EPC-256	**TYPE I**	8	32	56	192
EPC-256	**TYPE II**	8	64	56	128
EPC-256	**TYPE III**	8	128	56	64

4. EPC-256 DESIGN

Given the discussions of the previous section, a practical large size object identification number that is consistent with the previously defined EPC-96 and EPC-64 versions can be defined. The large number of bits available for future use in a 256-bit representation provides a large amount of flexibility in the design of an EPC-256. We utilize this flexibility and define three versions of the EPC-256 as shown in Table 1.

All three of these versions provide 56 bits for the Object Class number enabling the unique identification of more than 7.20 x 10^{16} object classes by each Domain Manager. The number of bits in the Domain Manager partition is the primary differentiator between each of these 256-bit versions. The EPC-256 Type I is capable of representing more than 4.2 billion unique Domain Managers, each capable of uniquely identifying more than 1.05 x 10^{65} items. The EPC-256 Type II is capable of representing more than 1.84 x 10^{19} unique Domain Managers, each capable of uniquely identifying more than 2.45 x 10^{55} items. The EPC-256 Type III is capable of representing more than 3.40 x 10^{38} unique Domain Managers, each capable of uniquely identifying more than 1.32 x 10^{36} items.

5. CONCLUSIONS

The Electronic Product Code™ enables the unique identification of all objects, whether physical or virtual, and may be used to identify non-objects, such as services and groupings. An EPC™ is a standard representation of the Electronic Product Code™. The previously defined EPC-96 and EPC-64 versions were designed specifically for the low-cost identification of physical objects. The low cost limitations of these versions resulted in a potential limitation on the practical use of the Electronic Product Code™ over its long term use to identify objects and non-objects alike. The EPC-256 is a larger Electronic Product Code™ representation designed to enable the long term use of the identification scheme in all contexts requiring unique identification.

The desire to remain flexible in the face of unknown future applications has led us to define three EPC-256 versions that remain consistent with the previously defined 96-bit and 64-bit Electronic Product Code™ representations while providing well defined, yet flexible 256-bit representations. The definition of a 256-bit EPC™, a length that is not expected to be exceeded even over the long term use of Electronic Product Code™ identifiers, further clarifies the set of potential EPC™ representations, removing a considerable amount of uncertainty over possible future versions.

6. REFERENCES

1. **D. Brock. "The Compact Electronic Product Code – A 64-bit Representation of the Electronic Product Code™".**
   Technical Report MIT-AUTOID-WH-008, Auto-ID Center, November 2001.

2. **D. Brock. The Electronic Product Code™ – A Naming Scheme for Physical Objects.**
   Technical Report MIT-AUTOID-WH-002, Auto-ID Center, January 2001.

3. **Auto-ID Center and OatSystems. "The object name service: Version 0.5 (beta)".**
   Technical Report MIT-AUTOID-TM-004, Auto-ID Center, February 2002.

4. **Uniform Code Council. Uniform Product Code (U.P.C.) Symbol Specification Manual.**
   http://www.uc-council.org/re ib/01302/d36-t.htm, January 1986.

Designed and produced by Foxner. www.foxner.com