1-1-1981

Mapping a CAD data base file.

Iges Standards

MAPPING
A CAD DATA BASE FILE
TO
IGES STANDARDS

by

Diane Louise Kmetz

A Thesis

Presented to the Graduate Committee

of Lehigh University

in Candidacy for the Degree of

Master of Science

in

Computer Science

Lehigh University

1981

This thesis is accepted and approved in partial fulfillment of the requirements for the degree of Master of Science.

August 19, 1981
(date)

Professor in Charge

Chairman of Department

TABLE OF CONTENTS

LIST OF CHARTS

LIST OF FIGURES

ABSTRACT

As more and more industries get involved with Computer Aided Design / Computer Aided Manufacturing (CAD/CAM) systems, the need for the ability to transfer graphical models between systems increases. This need was actively realized by the Dept. of Defense and together with the National Bureau of Standards, CAD/CAM vendors, and corporate users an initial set of specifications for the transfer of data between CAD/CAM systems was developed as the Initial Graphics Exchange Specification (IGES).

This thesis describes the data base and file generation capabilities of the Tektronix Mechanical Engineering Graphics CAD/CAM system as input to the preprocessor designed for the mapping of the 2 dimensional geometric entities to IGES. Many assumptions had to be made due to flaws discovered in the input such as the absence of the group entity. The preprocessor was tested with a simple model utilizing the 2-D entities and appears to be in proper IGES form. The true test would have been to pass the IGES file through a postprocessor (from IGES) to another CAD/CAM system, but that is beyond the scope of this thesis.

INTRODUCTION

Computer graphics is a field where the combination of the advances in hardware and software have turned it into a powerful tool. An area where industry can very profitably utilize this tool is in Computer Aided Design / Computer Aided Manufacturing (CAD/CAM). This tool can be used by designers, draftsman, and manufacturing, all drawing upon the information stored in a single data base.

The fact that this area has grown so rapidly has motivated the vendors to design their data bases independently trying to get an edge on the market. Unfortunately, these independent data bases are incapable of communicating with each other. This is a serious problem in installations that have several different CAD/CAM systems with differently designed data bases.

In an attempt to alleviate this incompatibility between data bases, the National Bureau of Standards together with interested parties formed the Initial Graphics Exchange Specifications (IGES). This file format acts as a stepping stone from one data base to another, with pre- and post- processors required to perform the mapping.

WHAT IS A CAD/CAM SYSTEM?

A total CAD/CAM system is the marriage of the two individual partners, CAD and CAM, with the data base bonding them together.

CAD/CAM technology is in a state of change which makes it difficult to tag it with an exact definition. It is generally considered to be an interactive graphic system that allows the user to rapidly accomplish drafting, design (geometric modeling) analysis, and numerical control machining functions in less time than would be required by traditional methods with increased accuracy. Each system has its differences, some subtle and some blatant. A most obvious difference is the category of the system configuration. A CAD/ CAM system may be turnkey (dedicated system complete with application-specific software) mini-computer based or large mainframe-computer based. More subtle differences can be detected, upon closer investigation, in the areas of entity generation or manipulation methods. Some systems are limited to just 3, 4, or 6 sided polygons, while others are unrestricted as to the number of sides allowed. Some systems don't have 3-D fillet surface capabilities, or may be very limited as to what types of entities can be grouped or mirrored.

The designer creates models by generating 2 and 3 dimensional graphic entities and manipulating them as required with the CAD portion of the system. Among the 2-D entities are points, lines, curves, and splines. 3-D entities can be created by taking 2-D entities, which may have been manipulated, and rotating them about an axis. Certain other 3-D entities may be part of the system's extended geometry as torus, cylinder, sphere, and plane. These models may be subjected to engineering analysis for stress determination or for clearance-interference testing.

The CAM portion of the system is used by the manufacturer to view the model, designed in CAD, to simulate complex cutter motions. The animated tool paths are displayed on the screen for immediate feedback.

A complete system must have a data base capability with storage and retrieval functions. The key ingredient to a successful CAD/CAM implementation is a centralized, communicating design data base, rapidly accessible (interactively) by all users. The data base may contain a collection of standard parts from the work of many individuals in a working environment. Conversely, many individuals can be using the system at the same time to design a single model. An integrated design data base is essential to the effective management of an evolving design involving large amounts of data. The nature of an CAD/CAM data base is different from the classical manufacturing data base (for inventory, accounting etc.). The difference is based on the types of data that are stored, the algorithms that generate the data, and the requirements that are imposed on CAD/CAM systems for efficient primary memory structures.[1] The number of data entities or types of data is greater in a CAD/CAM data base, and the relationship between data entities may change depending on the type of part or analysis desired.

----

1. P.L. Ciampi and J.D. Nash, "Concepts in CAD Data Base Structures", 13th ADC, (1976), p. 293.

A substantial part of total system productivity is in the editing or revision function since change is the rule rather than the exception in industry. Once the drawing is stored in the data base, all the user need do is to call that drawing back on the screen to add, delete, connect, rotate or move the components interactively. The CAD/CAM system also takes advantage of the repetive nature of engineering drawing by allowing one to create a library of standardized symbols that are used over and over again and stored in the data base. Having access to this common data base, personnel from both design and manufacturing can rapidly recall a drawing at any time during the production process to make necessary modifications. The revised drawing may be stored as a revised copy, or may replace the original drawing in the data base. It frees the user (usually an engineer or designer) from tedious, time consuming chores that have little to do with technical expertise, to go from an initial concept to the finished part with one system.

The basic CAD system developed in the early 1960's in the areospace industry aided with large amounts of government funding. These expensive systems were run on large mainframes, but with the advent of the mini-computer, compact and less expensive systems were developed. Since 1970, a number of commercial mini-based graphic systems have been available. These systems found widespread application in the preparation and generation of printed circut board artwork. These turnkey systems are used for digitizing manually produced layouts and for generating plot tapes. In addition to artwork creation, CAD systems are frequently applied to the creation of support documentation. Interactive graphics allows the user to communicate easily with the computer in pictures instead of studying reams of paper loaded with data. CAD/CAM is still in its infancy and must be refined before its full potential is realized. Projects were developed to help organize the refinement activity such as IPAD (NASA's Integrated Program for Areospace Vehicle Design).

Most systems consist of a mini-computer or central processing unit (28k to 100k of core), disk units for storage and retrieval, hard copy units, plotter, and an interactive graphic terminal. Some configurations allow the minis to communicate (directly or via modems and phone lines) to large mainframe computers that can take properly formatted graphic information and run large scale engineering analysis programs (e.g. ANSYS, SAP, NASTRAN). The commands are entered into the system via electronic pen, alphanumeric keyboard, function keyboard, or menu selection.

CAD functions may be grouped into broad categories: geometric modeling, drafting, analysis, and kinematics. Geometric modeling is the area where the user describes the shape of a structure with a model constructed graphically on a CRT screen with a light pen, or by input of coordinates and parameters from a keyboard. The system converts the pictorial data into a mathematical model stored in a data base for latter use. This may be the most important feature of a CAD system because so many other design functions depend on the model. Most modeling is done with wire frames that represents the part shape with interconnected line elements, and can be 2 or 3 dimensional. The 3-D function of hidden line removal may be done by various semi-automatic or interactive techniques in which the portions of the drawing to be hidden are removed or changed to dashed line representations. The more advanced systems have 3D solid modeling using building blocks of elementary solid shapes called primitives.

A point to mention with respect to data base manipulation is the availability of both 2-D and 3-D data bases. The 2-D data bases have construction information for projection into 3-D. Some CAD systems have the capability of working in a true 3-D data base where each point in a 3-D drawing has (x,y,z) coordinates.

The drafting feature of a CAD system automatically produces detailed engineering drawings from the data base and may include automatic scaling and dimensioning functions. The drawing capabilities may include size and location of lines, arcs, text, cross-hatching, and superimposing of patterns. Software to check design violations such as near-shots and open runs in printed circut board designs may be included.

The data base can now be accessed to aid in different areas of analysis. Once the model is created and stored in the data base, a good CAD system will allow the user, thru simple keyboard commands, to calculate values such as weight, volume, surface area, or moment of inertia of a part. A powerful method of analysis of a model is the finite element method in which the structure is broken down into a network of simple elements. The CAD system may have automatic node and element generation capabilities to break the model into workable elements. The processing of these elements require tremendous computational power and so must be done on large mainframe computers. The same function done by hand is tedious and time-consuming. The beauty of a CAD system is that if the finite analysis indicates too much deflection or stress the model may be modified and re-analysed before it's actually built.

Kinematics automates the motion of simple hinged parts such as doors or cranks, and may be included in the CAD system package.[2]

The CAM functions included in a system allow the user to produce numerical control instructions for machine tools, produce process plans for fabricating the complete assembly, program robots to handle tools and coordinate plant operations with a factory management system.[3]

----

2. J.K. Krouse, "CAD/CAM - Bridging the Gap from Design to Production", Machine Design, June 12, (1980), p. 121.

3. Ibid., p. 117.

CAD DATA BASES

The intent of this paper is to deal with the 2-D entities created in the CAD portion of the system and the resulting data base.

The fact that the data base is the integrating element between CAD and CAM, brings us to the question, 'What is a CAD system data base?'. The data base lies at the center of a multi-user multi-application system used for long-term storage of data and for exchange of information between the application programs and between different users. Most mini-based graphic systems include the collection, reduction, and verification of design data and the management of the design data base in addition to artwork generation. The central data base can actually consist of a variable number of data bases, storing information such as offsets of a contour, description of a part, or the results of a finite element method computation.[4] Let's refer to this data base as the general data base that is distinguished from the strictly graphic data base by the type of non-graphical data maintained such as weight, volume, or finite element results.

For short-term storage the information for the current part rests in the working area (core) where it's accessible for further use. The integration of all data needed for the analysis routines and for drafting will result in data bases far too large to reside in core memory. Most CAD systems do not interact directly with the disk data base

----

4. K.P. Beier and W. Jonas, "DINAS - A Transportable Executive for Interactive Computer Aided Design", IEEE, (1978), p. 396.

structure. Instead they extract the required data and then re-format it into primary core storage data structures which they use for processing and on command can add to or change the permanent data base. Until the command is given to store the data in the data base, this data resides only in core, where it's manipulated and will not affect the data base until commanded to do so. At this time the data is transferred to the permanent storage medium. Each CAD system has actually two data structures; the disk structure which contains all of the data associated with a design, and a core storage working area structure.

Organization of very large data on peripheral storage devices has become an important CAD issue. Investigations of CODASYL type data base management systems and development of new structures and concepts have been performed. The integration of data into a single data base can be accomplished in a variety of ways.

The development of a single, integrated (allow for multiple types of analysis or application) data base capable of storing all design information in a special data structure, with language facilities which can reformat data for analysis and also compute new data which is a function of existing data, is essential in a top-notch CAD system. Data for analysis is computed or mapped into drawing formats when it's needed.

One possible approach is to accumulate the data requirements for all anticipated analysis and drafting into a large data base, implying that the information on each entity would be stored in multiple ways. An example of the kinds of information stored would be 'two joint centers with an edge between them, a set of loads and section modulus (for statistical analysis); as a mass with known conductance, with surfaces of given area (for thermal conductance analysis) and as surfaces of given reflectance and orientation (for lighting design) plus various sections and plans for drafting.'.[5] The requirements of each analysis requires unique information about various relations between elements. All of the data and relationshtps are amendable to storage in a standard data base definition and access system, following CODASYL standards (a subschema could be defined for each analysis).[6] The problem with this storage technique lies in the fact that much of the data stored is functionally related implying that a change made to one piece requires that all data related to the changed values or relations be updated also. Keeping a data base consistent is expensive and difficult. A solution is to store only non-redundant data, recomputing the redundant information from the stored data; in other words, to reduce the data to its independent components and store in non-redundant syntax. When a particular analysis requires the data to be organized in a specific way, the information is computed from the stored data, manipulated, then re-organized into the non-redundant description. The resulting data base is compact and manageable. The CAD system must provide facilities for specifying functional relationships between data and for computing needed information from the stored entities.

----

5. C.M. Eastman, " Databases for Physical System Design: A Survey of US Efforts", CAD '76, (1976), p. 3.

6. Ibid., p. 3.

CAD systems can be designed to handle non-spatial (e.g. part number, value, character, tolerances) or spatial data. Spatial data may be defined in many ways: 'an edge between two joints, rectangular solids, a set of line vectors in 2 or 3 d, or a polyhedral geometry'.[7] If storing all representations is the chosen strategy, then the shape consists of all these representations and must be maintained properly. Else, computing shape data as needed requires the most complete and general representation be used to store the non-redundant shape description - a polyhedron.[8] The choice has major implications on the physical size of the data base required to depict the design data, and the speed of interaction with the software. Only non-redundant data will allow implementation on minis. Ultimately, CAD systems will use a mixture of both storing and computing redundant information based on an optimization criterion.

Most CAD systems involve some properties and data which will be part of any application, while other data will be variable, depending on the use of the system within an organization. Data base systems are often categorized on the basis of their structuring methods (hierarchial, network, or relational), and are restricted to specific mechanisms. Many CAD systems are hierarchial. They incorporate a data base from which are extracted subschemas, from which may be extracted sub-sub-schemas.[9]

----

7. Ibid., p. 4.

8. Ibid., p. 4.

9. Ibid., p. 5.

There is a dire need for communication between integrated CAD system data bases. This sparked the formation of IGES (Initial Graphics Exchange Specifications) to provide a format for the exchange of data between CAD systems. The standards were derived from existing practices and collective experiences to facilitate communication of data between CAD/CAM systems of different hardware, operating systems, and application software.[10] It specifies file structure and language formats for communication of product definition data created and used by CAD/CAM systems. Product definition (PD) data is that information, regardless of form, required to describe and communicate engineering characteristics of physical objects as manufactured products.[11] It includes geometric (points, curves, surfaces, and solids), topological (vectors, edge, face, and objects), and non-geometric (dimensions and notes) information required for engineering drawings and product models.[12] It has been widely recognized that a roadblock to increased productivity and to realizing the full potential of CAD/CAM systems is the incompatibility of data, and this hopes to provide the solution to the problem.

----

10. National Bureau of Standards, Initial Graphic Exchange Specifications, 1980.

11. Ibid.

12. Ibid.

MEG

The CAD/CAM system available to me is Tektronix's Mechanical Engineering Graphics (MEG) Design, Drafting, and Numerical Control. To demonstrate the capabilities of this system, I set out to design a roll (used in rolling steel) and perform stress analysis on the finished model. A data base was created to hold the finished model along with the sequence of parts leading up to the completed design. The sequence of events was as follows :

STEP 1 - using the Point and Line functions, outline the shape of a quarter roll.

STEP 2 - using the Arc/Circle function, create the rounded curves (see Figure 1).

STEP 3 - using the Entity Manipulation function, mirror the quarter roll to create a half roll.

STEP 4 - using the surface of revolution capability of the Extended Geometry function to create a 3-D figure.

STEP 5 - in the Models & Fonts function, increase the number of view paths from the default values for a more defined shape.

The created model may then be viewed from different angles such as Top, Side, and Isometric by simply telling MEG how many and which views are desired (see Figure 2). My CAD system also has an analysis function that allows for calculations of surface area, weight, center of mass, and moment of inertia as part of the stress analysis.

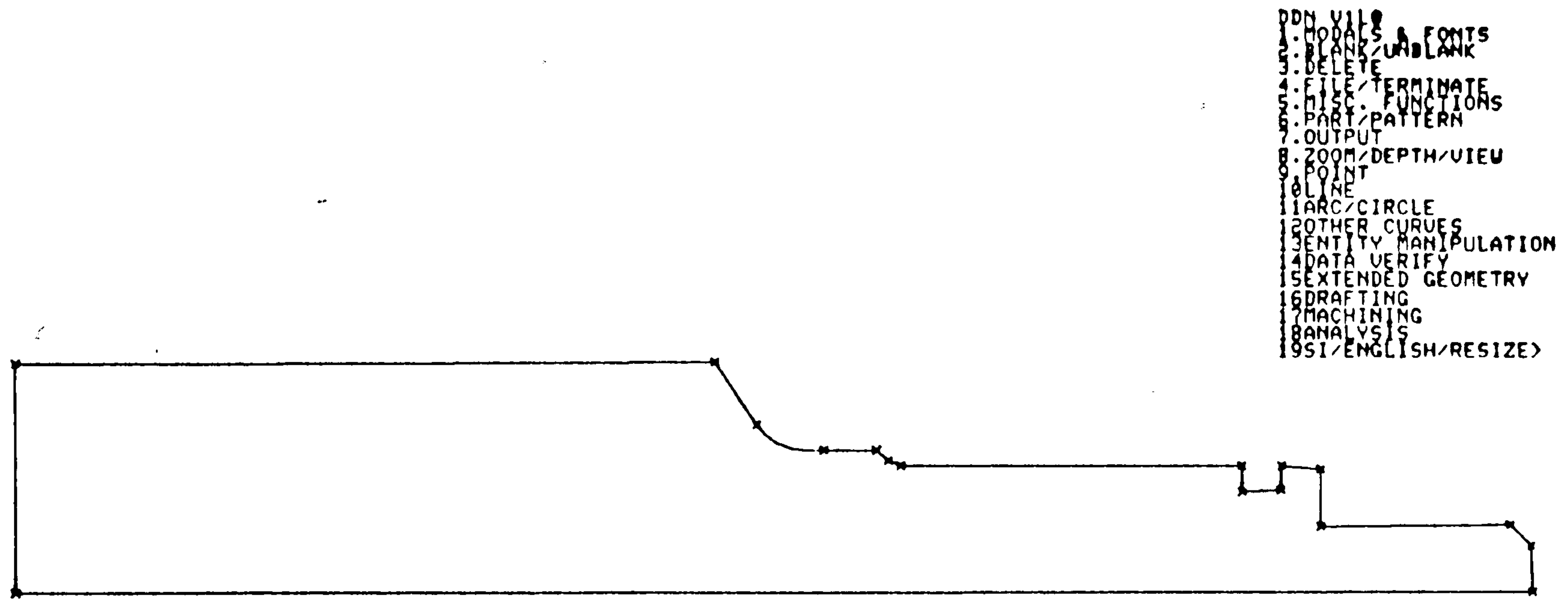

FIGURE 1. OUTLINED SHAPE OF THE QUARTER ROLL

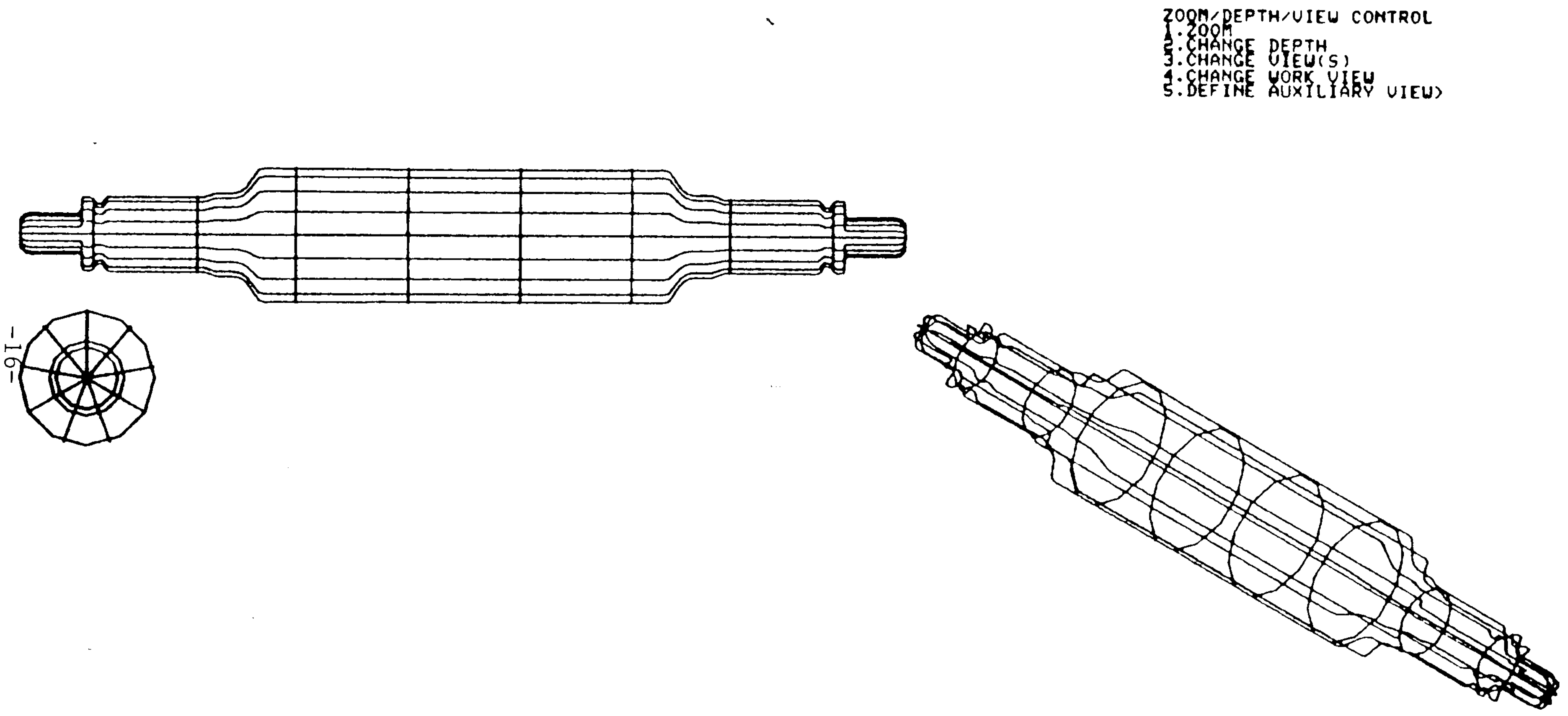

FIGURE 2. THE ROLL VIEWED FROM TOP, SIDE, AND ISOMETRIC

Using the mathematical representation of the 3-D roll stored in the data base, the CAD system can be requested to reformat the data for use by the Finite Element Method software package. The model will be broken-up into a network of simple elements showing deflection or stress when certain loads are applied. Based on the FEM results, the user can modify the shape of the roll, re-store the figure in the data base, and continue with the stress analysis.

A created design may be stored in the system (on the permanent storage medium) in two ways depending on the future intent of the design: pattern or part. It should be stored as a pattern if it's a specially created symbol that will be used frequently, because it will be stored in the pattern library which may be accessed by all parts. When the design is stored as a part it can't readily be accessed by other parts. An easy way to think of the difference between a part and a pattern is that the patterns are the building blocks that parts can be constructed from. For example, if I create a design of a special spline that I know is an element in a lot of my models, I should store it as a pattern. That way when I construct each model, I can access the pattern library and call for my special spline, instead of creating it from scratch each time for each model.

When I refer to entities stored in the data base, I am only concerned with the entities stored as parts, not patterns, because if a pattern is to be an element of the model, it will also be stored as an element of a part in the data base. Not very surprising then, any file that is created as output from a MEG function will contain only the data of the designs stored as parts. (The two output files will be covered in thorough detail later.)

THE MEG DATA BASE

The data base is logically and physically divided into 5 different areas as follows:

AREA 1 - This is the largest area of the data base, and is used for storage of PARTS. The PARTS themselves consist of a complete copy of COMMON, as it existed when the part was filed, and a copy of all the data pages for the PART. (A page is the way information is kept in the COMMON arrays.) When the file command is issued, the information from AREA 4 is copied to this area. The PARTS are filed sequentially from the start of this section, while the index information starts at the end and grows toward the beginning.

AREA 2 - This area contains PATTERNS, which are portions of drawings that have been saved for use in other drawings.

AREA 3 - This is the Special Definition area which is not of interest in the paper.

AREA 4 - This is the PAGE STORAGE area where all the data is maintained for the PART or PATTERN currently being created. As the arrays are filled in core, they are written to this area while the CAD operations are being performed.

AREA 5 - This is a scratch area that is used when certain CAD operations are performed.

To summarize the structure of the MEG data base in basic terms, let's use the following bottom to top explanation. First, at the bottom, is the entity (a single geometric item created using one of the MEG operations, ex: point, line, or circle). Many entities make up a PART. Many PARTS are stored in AREA 1. There is only one AREA 1 per MEG data base.

Now, let us continue with a detailed discussion of the geometric building block of the MEG data base, the entity. Each entity is assigned a pointer number (a number which locates the exact position within the data base) and a sequence number (the number which reflects the history of the PARTS construction). Either number can be used to identify an entity within a PART to MEG. The pointer number is an actual physical location on the disk within the data base where the data resides for each entity. The initial value of the pointer is based on the first available location for each part, which is 130, and is incremented depending on the amount of space needed to store each entity, usually 1. The sequence number is a logical number in the respect that it's used to identify the creation order of each entity of a part. It always starts at 2 and is incremented by 1 for each additional entity. (The sequence numbers 0 and 1 are reserved by the system.)

As an example to clarify the above, I created a part called PTSANDLINE through the following steps. First I created a point at x,y

= (5,5), then another point at x,y = (7,5) and drew a line connecting the 2 points.

ENTITY in order of creation)	LOCATION	SEQUENCE #	POINTER #
POINT	(5,5)	2	130
POINT	(7,5)	3	131
LINE	(5,5)-(7,5)	4	132

For each PART in AREA 1 there are four arrays - Tabl, Tab2, Tab3, Tab4 - that contain data base information. Tabl is the Master Entity list. All the data for a specific entity can be accessed through the Tabl array. The entity header (160 bits) for each entity in Tabl contains pointers to the entity information in Tab2 (all the integer data), Tab3 (all the real data), and Tab4 (information for each view defined by the part). (Tab4 will not be discussed in more detail because it's not of concern in this paper.) The pointer identifies the page number and the item number, in that page, for the particular entity. Other data contained in the header are the entity type and sub-type, number of Tab2 words, number of Tab3 words, and level number. (Entities may be created at different levels indicating, perhaps, a building foundation at one level, and the floor plan at another. Also, entities may be manipulated by level. You can delete all entities at a specified level.) Tab2 and Tab3 vary in the number of values and the meaning of the values according to the entity type and sub-type.

MEG was intended to be truly associative. What this means is that if a line was created thru a point and tangent to a curve, and the curve was translated to a different location, the line would also be changed appropriately. But in reality, associativity was not programmed into the MEG system, even though that capability resides in the data base. Case in point : When a line is of sub-type 5 (through a point and tangent to a curve) the TAB2 data has a pointer to the sequence number of the point, and to the sequence number of the curve. The TAB3 data has the two end point coordinates (x,y,z), of the point and the point on the curve. Now, when the curve is translated, the line remains in its current position.

There exists a function in MEG that permits piecemeal inspection of the data base, by indicating the entity in the current part to be examined. The values displayed through this function are taken from the data base and unpacked for user convenience. (Please see Chart 1 for examples of Tabl, Tab2, and Tab3 values for PTSANDLINE.) In reality, the values are stored in the data base using bit manipulation methods making actual viewing of the data base (in EBCDIC and HEX) very difficult.

MEG has a function that allows the user to save parts (but not patterns) in separate files on disk for later transfer to other data bases if desired. Two types of files can be created: Fast Save and Computer Independent Save. Fast Save files allow the user to transfer parts previously stored in one data base to a separate file. If desired the entire data base, with all its parts, can be created as one FS file. Later, when MEG is working with a different data base, a certain MEG function can access this file and store the saved parts in the second data base. This relieves the user from having to recreate the parts in a different data base.

CHART 1   DATA BASE INFORMATION FOR THE ENTITIES IN PTSANDLINE

TAB1 INFORMATION	POINT	POINT	LINE
1. pointer	130	131	132
2. type	1	1	2
3. sub-type	1	1	2
4. attention status	on	on	on
5. blanked status	no	no	no
6. deletable	yes	yes	yes
7. dormant	no	no	no
8. EMPTY			
9. font	solid	solid	solid
10. display	all	all	all
11. group counter	0	0	0
12. sequence number	2	3	4
13. def. counter (not used)	0	0	0
14. attention point x,y	2572,1416	3132,1416	2852,1416
15. number of Tab2 words	0	0	2
16. number of Tab3 words	3	3	8
17. level	0	0	0
18. view number	1	1	1
19. pen number	0	0	0
TAB2 INFORMATION			130
			131
TAB3 INFORMATION	5.0	7.0	0.0
	5.0	5.0	1.0
	0.0	0.0	5.0
			5.0
			0.0
			7.0
			5.0
			0.0

The Fast Save can only transfer parts to data bases on the same computer system. The CIS files allow transfer of parts to any computer system that supports the MEG CAD/CAM system. The entire data base could be transferred, but only as a series of individual CIS files, each containing one part.

The purpose of the CIS file is transportability of parts to other MEG systems. Tektronix may have realized the growing need for communication between data bases on different systems. Their first shot at it would naturally be communicating graphical data between different systems running their MEG software. Or, it may have been their first attempt at meeting the IGES standards. But as it stands, the CIS file is definitely not in IGES form. (Tektronix has changed its marketing strategy and has dropped support of MEG, so no revision of the CIS format can be expected from that company.)

The CIS file has several shortcomings and flaws. One of them is the fact that it doesn't contain the level information that exists in the header of the data base for each entity. In constructing the IGES file from CIS, I assume the level to be 0 for all entities.

Blanked entities are not part of the CIS file. A blanked entity is one which resides in the data base, but the display of this entity has been suppressed. The CIS file saves only the current part and only the portion which is displayed in the workspace. Therefore, blanked entities and entities out of view as a result of zooming, for example, are not transferred.

IGES

IGES is a set of specifications for the exchange of data between CAD/CAM systems. The idea to create a file in IGES form as an immediate step in the solution of the CAD/CAM data exchange problem was developed at a Dept. of Defense meeting, Sept. 1979. As a result of their recommendation, a meeting was convened by the military, NASA, and the National Bureau of Standards, Oct, 1979.[13] Following presentations by vendors, corporate system designers, and standards groups, it was agreed that IGES was needed immediately.

Members from the NBS and large corporations, involved in CAD/CAM, were asked to join the IGES committee and produce the standards early in 1980. The Initial Graphics Exchange Specification report is the result of an intensive three month effort by the technical committee and many others who have worked with the committee and provided input and feedback.

It must be noted that IGES was formed to meet an immediate need. It's expected that IGES will be an important step in the complex procedure required to produce a full fledged national standard.[14] At that time, hopefully, IGES will be accepted by all vendors.

Being that this is the first attempt at standardization, IGES may be missing some of the special entities belonging to a specific system,

----

13. Ibid.

14. Ibid.

or may contain some errors in entity representation. Only time and feedback from people using IGES will improve the standards.

The IGES file provides formats that support the communication of geometric and non-geometric categories. This file structure treats the part definition as a file of entities, each represented in an application independent format, to and from which the native representation of a part in a given CAD/CAM system can be mapped. To use IGES, pre- and post- processors must be written for each CAD/CAM system. The CAD/CAM data base must be reformatted (via software) to the IGES specifications.

The basic unit of information is the entity. There are three kinds of entities: Geometric (i.e. points and lines), Drafting (i.e. dimensions and notes), and Structure & Definition (to define views of the model and form relations among other entities). (We will concern ourselves only with the geometric entities.) There may be any number of entity occurences in the file, each one consisting of the identification of its type, parameter values, and possibly pointers to other entities as required by the part. Each entity occurences is assigned a unique ID in the file. This ID is used to relate various entity occurences, thereby avoiding redundant data. Many also are assigned a form number as an attribute to further define the entity within its type. The form number was created to allow several entities to share the same parameter entry with a slight change in interpretation. For example let's use the conic. The form number serves the same purpose in IGES as the type parameter in the TAB2 array does in CIS, which is to indicate whether the conic is a hyperbola, ellipse, or parabola.

The information for the entities is distributed into two subfiles. The first subfile contains the Directory Entry (DE) data for each entity. This is a fixed length entry, formatted the same for each entity type. The second subfile contains Parameter Data (PD) for each entity occurence, which varies both in length and format. The DE data for each entity occurence contains a pointer to the PD for that entity.

The created IGES file is a file of 80 column card images, containing these 5 sub-sections as sequenced:

1. Start section (code = S).
2. Global section (code = G).
3. Directory Entry section (code = D).
4. Parameter Data section (code = P).
5. Terminate section (code = T).

The code must be in col. # 73 followed by the sequence number of the cards which starts at 0000001 for each section.

In brief, the Start section provides a human-readable prolog to the file, sort of an introduction. The Global section describes the preprocessor (the 'from' CAD/CAM system) and information needed by the postprocessor (the 'to' system) to handle the file. The DE section has exactly two card images for each entity, providing an index for the file and to contain some basic information about the entity. The PD section contains the parameter data associated with each entity. And of course, the Terminate section is the last card image in the file, and contains the last sequence number used in each of the previous sections.

The current version of the standards seems to imply that there will be one IGES file per CIS file (containing one part). The reason I state this is because they refer to 'Drawing Identification' in the Global Section in both the 'to' and 'from' system information. (Remember, there is just one Global Section per IGES file.) Maybe a way around this situation would be to utilize the user-defined associativity definition capabilities (explained in full detail later) provided by IGES. In that case, the postprocessor (from IGES) would have to provide that the proper links are maintained for the target data base as specified in the associativity definition. This is strictly speculation. IGES does not mention when or how to distinguish a part within a data base, unless they simply assumed that for the time being there would be only one part per file. Being that this point is so vague, I concerned myself only with one part contained in both the CIS and IGES file.

THE STRUCTURE OF THE CIS FILE

Let me explain the reason I decided to use the CIS file and not the FS file for mapping to IGES. This decision also influenced whether I used the MINI or IBM version of MEG.

Most of my previous investigation of the MEG System was performed on the IBM (32 bit) version, but I wasn't able to make much progress decoding the data base information. I was able to detect the header information for an entity on the MINI (16 bit) version, but could not find the information as indicated by the pointers in the Fast Save file. This file is voluminous, and lacking any helpful documentation. Back a few pages ago, I stated that MEG could create both FS and CIS files. In reality, only the MINI version can create the CIS file. There exists documentation for the CIS file, but this special file generation capability is not available on the IBM version for some reason (probably Tektronix dropped support of MEG before that portion of the code was implemented). Some of the information in the CIS file is in hexidecimal, some in ASCII. No single way of printing the file was very helpful. After much searching, I uncovered a 'dump' feature to get the hex and ASCII representation of a file and it works beautifully. I could read the data base information needed to convert a CIS file to IGES.

The CIS file format is divided into 5 parts.

First is the header record that contains : COMPUTER INDEPENDENT, and last is the trailer record that contains : THIS IS THE END, both in ASCII representation.

The second, third, and fourth records are repeated for each entity in the file. The second record (and hardest to recognize) is the header information from Tabl (Please refer back to chapter 'THE MEG DATA BASE') in binary representation consisting of the 160 bits of Tabl information reformatted into 22 bytes of information. (Please see Chart 2 for a description of each of the 22 bytes along with the actual data for the PTSANDLINE example.) A note to make here is that the sequence number now starts at 0. In reformatting, this value is 2 less than the value actually in the data base. The third record is a Tab2 record containing any integer values associated with the entity, each as a 10 character integer. The fourth record is a Tab3 record containing the associated real values, each as a 20 character real with the decimal point in the tenth position.

My next task was to determine the actual data needed by IGES to make sure the CIS file contains all the required information. Investigation indicated the CIS file contained most of the required information. Later, I'll explain what's missing from the CIS file. Each of the 5 sub-sections (briefly covered in chapter 'IGES') will be discussed in more detail. The information which will be required for PTSANDLINE will be shown.

The Start section is just an English prolog to the file, conveying any information the author feels is important. The Global section describes the pre/post processor, and information needed by the postprocessor to properly map the IGES file. Most of the values are fixed

CHART 2 HEADER INFORMATION IN THE CIS FILE FOR PTSANDLINE

BYTE	DESCRIPTION	POINT	POINT	LINE
1	entity type	01	01	02
2	entity sub-type	01	01	02
3	view number	01	01	01
4	number of integer words	00	00	00
5		00	00	02
6	number of real words	00	00	00
7		03	03	08
8	blanked switch	00	00	00
9	deletable	00	00	00
10	sequence number	00	00	00
11		00	01	02
12	view of definition only	00	00	00
13	curve font	00	00	00
14	group status	00	00	00
15	curve density	00	00	00
16	number of attributes	00	00	00
17	dormant switch	00	00	00
18	infinite line	00	00	00
19	length of integer entry	00	00	00
20		00	00	14=20
21	length of real entry	00	00	00
22		3C=60	3C=60	A0=160

according to the system from which the IGES file is created. System vendor and the number of bits for integer representation are such examples. A few values are variable and must somehow be input depending on the part being mapped. Drawing identification and file name are variable examples. The Global section has 22 values using free format input. (Please see Chart 3 for the description of the 22 values along with a mapping using the PTSANDLINE example.)

The Directory Entry section contains 20 right justified fields of 8 columns spread across 2 cards for each entity. Each field contains either the actual attribute, or a pointer to a location in the file where the information is stored. A negative number implies it's a pointer and not an attribute. (Please see Chart 4 for the description of the 20 fields along with the PTSANDLINE mapping.) In this example, none of the values are pointers.

The Parameter Data section contains various free format fields of data associated with each particular entity. The standards currently have specific forms of PD for these geometric entity types: circle, composite entity, conic, copious data, face, line, parametric spline, parametric bicubic, spline surface, point, ruled surface, and surface of revolution. The first field always contains the entity type number as coded in IGES standards. The other fields depend on the entity type. The free format part of these cards ends in col. 64, with 65 - 72 containing a pointer to the first card in the DE section for the specific entity. Except for text strings, parameter values are restricted from crossing card boundaries. A semicolon indicates that the parameter

CHART 3 DESCRIPTION OF THE GLOBAL SECTION OF IGES FILE

PARAMETER	DESCRIPTION	PTSANDLINE
1	delimeter character	,
2	end of line character	;
3	drawing id (sending)	ptsandline
4	file name	cis
5	system id	Tektronix,V1.L02
6	translator version	1
7	# of bits for integer	16
8	# of bits for real exp.	8
9	# of bits for real mantissa	32
10	# of bits for double precision exp.	8
11	# of bits for double precision man.	64
12	drawing id (receiving)	ptsandline
13	drawing scale	1.
14	unit flag	1
15	units	inch
16	max. line weight	1
17	max. line width	0.014
18	date and time of file generation	
19	min. resolution of model space	0
20	max size of model space	1023
21	name of author	d.l.kmetz
22	organization	b.s.corp

CHART 4 DESCRIPTION OF THE DIRECTORY ENTRY SECTION OF IGES FOR PTSANDLINE

NUMBER	FIELD	POINT	POINT	LINE
(card 1)				
1.	entity type	116	116	110
2.	parameter data (pointer)	1	2	3
3.	version	1	1	1
4.	line font pattern	1	1	1
5.	level	0	0	0
6.	view	1	1	1
7.	defining matrix	0	0	0
8.	label display asso.	0	0	0
9.	status	0	0	0
10.	sequence number	1	3	5
(card 2)				
11.	entity type	116	116	110
12.	line weight	0	0	0
13.	pen number	1	1	1
14.	parameter card count	1	1	1
15.	form number	0	0	0
16	RESERVED FOR FUTURE USE			
17.	RESERVED FOR FUTURE USE			
18.	entity label	point	point	line
19.	entity subscript	5	5	4
20.	sequence number	2	4	6

list is complete and any remaining values should receive default values. (Please see Chart 5 for a description and the PTSANDLINE mapping.)

There is only one card for the Terminate section, divided into 10 fields of 8 columns, containing the last sequence number for each of the previous 4 sections.

All the information I need for IGES at least for the PTSANDLINE example is indeed contained in the CIS file on the MINI version.

CHART 5 DESCRIPTION OF THE PARAMETER DATA SECTION OF IGES FOR PTSANDLINE

PARAMETER	DESCRIPTION	POINT	POINT
1	entity number	116	116
2	x coordinate	5.	7.
3	y coordinate	5.	5.
4	z coordinate	0.	0.
5	pointer to display symbol	0	0
6	number of asso. entities	0	0
7	end of line	;	;
8	DE pointer	1	3

			LINE
1	entity number		110
2	starting point	x	5.
3		y	5.
4		z	0.
5	ending point	x	7.
6		y	5.
7		z	0.
8	number of asso. entities		0
9	end of line		;
10	DE pointer		5

MAPPING

The preprocessor is programmed in Fortran, and handles all the tasks including the necessary bit manipulation required to extract the vital information residing in the binary coded header record of the CIS file.[15]

Using Chart 6 as a reference, I will expand on each of the MEG entities I map to IGES. My primary goal is to concern myself with only the 2-D MEG entities: point, line, circle, conic, rotated cubic spline, composite curve, group, rectangular array, and circular array.

POINT

A point is stored in MEG as x,y,z coordinates in the TAB3 array, and pointers in the TAB2 array if necessary as indicated by the sub-type. The sub-type is based upon the creation method of the point.

IGES does not appear to be concerned with the creation method, just the coordinate values along with any pointers to associated entities. (For example, if it's a member of a group, the pointer will indicate the group entity of which it's a member.)

LINE

A line is stored in MEG as the x,y,z coordinates of the starting and ending points in the TAB3 array, along with a pointer in the TAB2 array if necessary as indicated by the sub-type. The sub-type

----

15. The preprocessor can be found in Christmas-Saucon, Room 102.

CHART 6 TABLE OF MEG AND IGES 2-D GEOMETRIC ENTITIES

MEG		IGES	
TYPE	NAME	TYPE	NAME
1	point	116	point
2	line	110	line
3	circle	100	circle
4	conic	104	conic
5	rotated cubic spline	112	parametric spline
6	composite curve	102	composite entity
12	rectangular array	-	DOES NOT EXIST
13	circular array	-	DOES NOT EXIST
15	group	302	asso. instance with form no. 1

is based upon the creation method of the line.

IGES stores this entity essentially the same way, by defining the two end points, along with any pointers to associated entities.

CIRCLE

A circle is stored in MEG as starting and ending angles in radians, x,y,z coordinates of the center point, and radius in the TAB3 array, and pointers in the TAB2 array if necessary as indicated by the sub-type.

The IGES circle is defined in terms of the center and end points (x,y). The points are defined so that the desired arc is traced from PT1 to PT2 in a counterclockwise direction. If the circle were to be in 3-D space (model space) a pointer to a transformation matrix would be included in the DE section for this entity. There may also be pointers to associated entities.

CONIC

A conic has no sub-type association. It's defined in MEG as having a type parameter (2=ellipse, 3=parabola, 4=hyperbola) in the TAB2 array. The information stored in the TAB3 array is a mixture of parametric and x,y forms. Given are the initial and final parameters, Z value of 0, end points coordinates (x,y), origin point in conic plane (xt,yt), coefficients of conic equation in a standard parametric form, and angle of rotation. MEG says nothing about the direction in which to construct each particular conic.

IGES constructs the conic in an xt-yt plane. The conic is defined by its 6 coefficients and starting and ending points. If the conic is an ellipse, it will be constructed in a counterclockwise direction. For a hyperbola, the starting and ending points determine the branch of this conic.[16] The DE section for this entity must have a pointer to the transformation matrix if it exists in 3-D space, and the form number indicated the curve type (0=to be determined from the conic equation, 1=ellipse, 2=hyperbola, 3=parabola). It may also have associated pointers.

SPLINE

The 2-D MEG spline is a rotated cubic spline with no sub-type association. The TAB2 array contains the number of points defining the spline, number of points to approximate each segment to .0001 crown height, and pointers to the points if not existing. The TAB3 array contains all the information needed to generate a spline through the indicated points using the Wilson-Fowler method.

In IGES the parametric spline is represented as a sequence of parametric cubic polynomials, but it can also represent other methods, one being the Wilson-Fowler method. To represent a 2-D spline, the z polynomial will be zero. The first parameter in the PD entry specifies the spline type (1=linear, 2=quadric, 3=cubic, 4=Wilson-Fowler). Also the dimension must be specified (2 or 3), the number of segments, along with all the polynomial (x,y) information.

----

16. National Bureau of Standards, Initial Graphic Exchange Specifications, 1980.

COMPOSITE CURVE

This entity is represented in MEG as having no sub-type association. Contained in the TAB2 array are the number of sub-curves, and pointers to each sub-curve (they must be contiguous). A negative pointer implies the sub-curves will be followed in reverse direction. (This condition was not handled in my program.) Contained in the TAB3 array is the starting parameter (0.0), the ending parameter (sum of the chord lengths of each sub-curve), chord length of first curve, sum of chord lengths of first 2 curves, etc.

IGES views a composite entity as a concatenation of entities, where the end point of each entity is the starting point of the next. The information IGES needs is the number of entities, and pointers to the DE of each entity. The sub-entities must have back pointers to the composite entity.

There seems to be a logical problem with IGES when an entity must be evaluated in the reverse direction. If an entity is defined in one direction and it's to be included in a composite curve where the reverse direction is necessary, there appears to be nothing in IGES analogous to the negative pointer in MEG. A duplicate entity must be created going in the reverse direction. But, if the entity that must be duplicated is a curve or ellipse, which must be evaluated in a counterclockwise direction, there seems to be no way to represent it properly in IGES.

GROUP

MEG has no sub-type association or TAB3 array data with this entity. The TAB2 array contains the number of entities in the Group along with a pointer to each of the entities. Something special to note about all the entities contained in the Group is that their Group Status bytes in their TAB1 arrays are activated. I have discovered an error in the MEG software that creates the CIS file. It appears that, even though this entity exists when the CIS file is created, it fails to include the Group entity itself. But all the entities comprising the Group do have the Group Status byte activated. I will discuss later how I attempt to program around this error.

IGES needs to know the number of entities, and a pointer to each of the members of the Group. Back Pointers are specified so each member entity must have a pointer to the DE of the Group. If an entity is a member of several groups, the entity will have a back pointer to each group of which it is a member.

RECTANGULAR and CIRCULAR ARRAYS

MEG stores these two entity types very efficiently by reducing redundancy. The elements of the array are not all stored in the data base! The TAB2 array contains a pointer to the Base entity, the number of elements in the array, do-don't list count, do-don't flag, and the positions of the elements. The do-don't information pertains to the individual elements in the array to indicate whether or not it's displayed. A rectangular array's TAB3 information includes coordinates of the origin, delta x distance, delta y distance, and the angle from the horizontal (tilt). A circular array's TAB3 information includes the origin, radius, starting radian angle, and delta radian angle.

There is no corresponding IGES entity! One possible, but inefficient way of mapping will be discussed under associativity.

IGES has included some advanced features in the first release. One of them is Associativity, which is used to relate several entities together logically. Needed for each type of relation is an Associativity Definition (specifies each type of relation) and an Associativity Instance (for each member of the type). The Associativity Definition permits the preprocessor to define an associativity schema by specifying the syntax of the relationship but not the semantics. Each different type of associativity is a class.

The syntax is indicated by the IGES schema which provides the specification as to whether or not back pointers are required. (Back pointer: an entity which is a member of a class, as in a Group, has a pointer to the DE of the Associativity Instance in its PD section.) It provides whether the class is ordered. (Ordered: the appearance of the entity in the class is significant.) Each entry (member of the class) could be composed of several items. The schema provides the number of items and for each whether it's a pointer (to an entity DE) or a data value.

IGES permits two kinds of Associativity: pre-defined (by IGES) and preprocessor defined. There are only 5 pre-defined in this version, but room for 5000. The pre-defined are identified by the form numbers 1 - 5000 and the preprocessor defined are identified by 5001 - 9999.

The pre-defined associativities are universal in their syntax and semantics because they are defined by IGES and their meaning is explained. Therefore no Associativity Definition is needed. Groups happen to be included in this kind of associativity and are identified by the form number 1. The preprocessor defined associativities require an Associativity Definition for their syntax plus documentation explaining the particular preprocessor's meaning of the relationship for the semantics. The postprocessor would have to refer to the Global Section to determine what preprocessor was used.

The preprocessor may have an entity that consists of a pattern of a complex shape. There is no corresponding entity in IGES. A way to map the entity may be to use a preprocessor defined Associativity Definition. First duplicate a simple entity (perhaps a point) at each location of the pattern. Then map the complex shape. A preprocessor defined Asso. Def. (with a form number greater than 5000) will relate the properly located simple entities and the complex shape. Preprocessor documentation will indicate how the postprocessor is to map the complex shape at each indicated location.

Each time an associativity relation is required, an Instance entity is used. If it's a pre-defined association, the version number will be

1, whereas if it's a preprocessor defined association the version number now will be a pointer to the DE of the Associativity Definition.

My first idea for mapping the MEG rectangular and circular arrays to IGES was to define my own Associativity Definition for both Rectangular and Circular arrays with an Associativity Instance based on my Definition. This would have helped me reduce the redundancy of duplicating the Base entity n times. Upon further investigation of the Associativity Def. schema, provided in the standards, there is no means for providing the delta distances, tilt, radius, and angles necessary to re-create the entitiy properly. The only option left is to take the information provided by the MEG Rectangular and Circular array entities, calculate the positions of each separate member element entity, and map each of those entities separately to IGES, making the IGES file very inefficient. For example, if I had a Circular array with a Base entity and 6 elements in the array, I must calculate the coordinates of each of the 6 elements and map 7 separate entity to IGES (versus 2 entities in MEG).

As stated in an earlier chapter, the CIS file has several flaws. The most detrimental for proper conversion from CIS to IGES is that the GROUP entity is never written to the CIS file, even though the member entities have the Group Status Byte (GSB) activated. There are many situations where the absence of the Group entity is harmful. If 2 GROUPS are adjacent to each other in the CIS file. I have no way of determining by examination of the CIS file where the members of one GROUP end and the members of the next GROUP begin. As far as I could tell, they all belong to one GROUP.

Another assumption I must make is that the entities comprising a Group are defined in sequential order (one right after another) which is not necessarily the way the Group was formed. The reason for this is the method used for determining members in a Group. In processing the CIS file, where the entities are listed in sequence, I assume the first entity without its GSB activated following one that is activated is where the Group entity itself should have been. Had the Group been there, it would have, with pointers, indicated which entities were really grouped. Let's use this as an example. If entities with seq. number 1, 3, 4, and 6 were grouped, the Group entity would have appeared in the CIS file following the entity with seq. number 6 with pointers to its members. But that is not how my preprocessor will interpret the CIS file. It will assume there to be three Groups: one after entity with seq. number 1 containing one member (because entity with seq. number 2 will not have its GSB activated), one after entity with seq. number 4 containing two members (because entity with seq. number 5 wil not have its GSB activated), and one at the end containing one member.

A question arose concerning the redundancy of member entities in the CIS file when an entity belongs to more than one GROUP. Following investigation of the CIS file for this particular case, the member entities are only written once in the CIS file, regardless of the number of GROUPS it belongs to. There would be (in theory anyway) a GROUP entity for each GROUP, containing pointers to the sequence numbers of its member entities. The header information for the entities in the CIS file doesn't reflect the number of GROUPS an entity belongs to as does the Group Counter byte that exists in the MEG data base. I must make another assumption før the preprocessor that an entity can only belong to one group.

IGES chose to implement several functional capabilities as associativities, as we discussed, one of which is the GROUP. "This associativity asserts that the member entities are related to each other in some manner which need not be made clear by the system."[17] As to why a COMPOSITE CURVE is not also indicated as an associativity function, I feel, has to do with the phrase 'related to each other in some manner'. A COMPOSITE CURVE must be a concatenation of other entities that are physically joined together to be manipulated as one. A COMPOSITE CURVE cannot be formed if the entities are not contiguous. This fact in itself disqualifies the COMPOSITE CURVE as being associated into a GROUP because the entities don't have the freedom to be just related in some manner, they must be contiguous.

----

17. Ibid.

BIBLIOGRAPHY

1. Beier, K.P. and Jonas, W. "DINAS - A Transportable Executive System for Interactive Computer Aided Design", Proc. Int. Conf. Interact. Tech. in Computer Aided Des. (IEEE), (1978), pp. 393-403.

2. Ciampi, P.L., and Nash, J.D. "Concepts in CAD Data Base Structures", Proc. 13th Annual Design Conf. (ADC), (1976), pp. 290-294.

3. Control Data Corporation, St. Paul, Minnesota. Telephone conversation with Steven Bodnar, April 14, 1981.

4. Eastman, C.M., "Databases for Physical System Design: A Survey of US Efforts", Proc. CAD '76, (1976), pp. 1-10.

5. Fowler, A.H. and Wilson, C.W., "Cubic Spline, A Curve Fitting Routine", Mathematics and Computers, Report Number V-1400, (1963), pp. 5-18.

6. Krouse, J.K., "CAD/CAM - Bridging the Gap from Design to Production", Machine Design, June 12 (1980), pp. 117-125.

7. National Bureau of Standards, Initial Graphics Exchange Specifications. Programmable Automation Section, 1 vol., (1980).

8. Tektronix, Inc. MEG Design Software Operators Manual. 1 vol. Oregon: Tektronix, Inc., 1979.

9. Tektronix, Inc. Plot80 Fortran IV Reference Manual. 1 vol. Oregon: Tektronix, Inc., 1979.

APPENDIX A
GROUP AND COMPOSITE ENTITIES

The differences between a group and composite curve are very subtle,. because they have a lot of similiarities. For example, you may delete the grouping of a group or a composite curve without deleting the entities which comprise the group or composite curve. You can not delete an entity within the group or composite curve without first removing the grouping. Groups can comprise points, arcs, circles, splines, conics, and composite curves, where as composite curves can only combine arcs, conics, lines, and splines. A group gathers separate entities together so they can be defined as a single entity, but is restricted as to what higher order operations can be performed. For example, one can't construct the surface of revolution of a group. This is due to the distinguishing fact that the members entities need not be contiguous! To create a group, the user must indicate each of the separate member entities. Each member entity of a group has the Group Status Byte activated indicating a relationship, and the group entity has only pointers to each member entity. Each entity comprising a composite also has the Group Status Byte activated. In addition the composite curve entity contains the sum of the individual chord lengths, and the incremental sums of the neighboring chord lengths. The member entities must be contiguous, else a composite curve can not be formed. To create a composite curve, the user must indicate the beginning member entity and then indicate the direction in which to chain the remaining entities together. I picture it as if a composite curve bolts together ajoining entities together to be manipulated as one, else there is no binding chain formed.

APPENDIX B
WILSON-FOWLER SPLINE

Here is the detailed analysis of the transformation of the Wilson-Fowler spline to the parametric IGES spline. Developed in the early sixties, the Wilson-Fowler spline method was developed as a mathematical model of fitting a curve through a given ordered set of points resulting in the rotated cubic spline entity. The fitted curve is comprised of a series of normalized cubic polynomials (one for each segment), hence the 'cubic spline' portion of the entity's name. [18] Each cubic passed through the points at the beginning and end of the segment. Each spline segment is defined in a separate coordinate system (p,q) whose X axis (p) is the straight line between the 2 points of the segment, and whose Y axis (q) is always drawn perpendicular to p. This is where the 'rotated' adjective of the name was derived. Each spline segment is defined by a cubic spline function; Ap + Bp + Cp + D (D is always 0 in that coordinate system), and the coordinates of the end-points. All this information is stored in the data base by segment (k) as;

tl - initial parameter.

XT , YT - coordinates of initial point.

θ - rotation angle in radians.

SL - slope at beginning relative to segment k as rotated.

SF - slope at end relative to segment k as rotated.

----

18. A.H. Fowler and C.W. Wilson, "Cubic Spline, A Curve Fitting Routine", Mathematics and Computers, Report Number V-1400, (1963), p. 5.

Within each segment (k) and for each tolerance parameter (t):

$$tl_k <= t <= t_{k+1}$$

$$p=t-tl_k$$

$$DT=tl_{k+1} - tl_k$$

The coefficients in p,q space;

$$A=(SL+SF) / (DT*DT)$$

$$B=(-2.0*SL-SF) / DT$$

The polynomial in p,q space;

$$q = Ap^3 + Bp^2 + SLp$$

The Wilson-Fowler spline can be converted to an IGES spline by rotating and translating the cubic spline back into the coordinate system of model space;

$$X' = XT + p \cos \Theta - q \sin \Theta$$

$$Y' = YT + p \sin \Theta + q \cos \Theta$$

Substituting for q gives;

$$XT' = XT + p \cos \Theta - (\sin \Theta (Ap^3 + Bp^2 + SLp))$$

$$YT' = YT + p \sin \Theta + (\cos \Theta (Ap^3 + Bp^2 + SLp))$$

Solve for the polynomial segment (k) in model space in powers of p;

$$X' = XT + (\cos \Theta - SL \sin \Theta)p - (B \sin \Theta)p^2 - (A \sin \Theta)p^3$$

$$Y' = YT + (\sin \Theta + SL \cos \Theta)p + (B \cos \Theta)p^2 - (A \cos \Theta)p^3$$

p becomes the indepedent variable for each segment.

The IGES coordinates of the points in the segment (k) of the spline are given by the cubic polynomials;

$$X(t) = AX + BXp + CXp^2 + DXp^3$$

$$Y(t) = AY + BYp + CYp^2 + DXp^3$$

Equate X' and Y' in MEG to X(t) and Y(t) in IGES;

$$XT + (\text{COS}\,\theta - \text{SL SIN}\,\theta)p - (\text{B SIN}\,\theta)p^2 - (\text{A SIN}\,\theta)p^3 = AX + BXp + CXp^2 + DXp^3$$

$$YT + (\text{SIN}\,\theta + \text{SL COS}\,\theta)p + (\text{B COS}\,\theta)p^2 - (\text{A COS}\,\theta)p^3 = AY + BYp + CYp^2 + DYp^3$$

Therefore the coefficients of the polynomials are the same. IGES needs the X and Y coordinate polynomials for each segment in the MEG spline.[19]

----

19. Control Data Corporation, Steven Bodnar, April 14, (1981).

APPENDIX C
THE PREPROCESSOR AS CODED

There were some logical problems that had to be addressed prior to developing the actual source code to perform the mapping. First, there exists a major difference in organizational structure between CIS and IGES. CIS lists all the essential information by entity. In other words, all the information is given for a particular entity before any information is given for subsequent entities. IGES is structured quite differently. Each entity has 2 separate parts: a DE entry, and a PD entry. IGES is organized this way to permit a uniform structure within the file with a minimium of wasted space.

What this implies is that the preprocessor needs a method of storing the entity information received from CIS before writing it to the IGES file in proper form and organization. The storage method may be an array or temporary file.

Second, in trying to plan the detailed flow of the GROUP entity, I've realized some special processing of this entity must be performed for proper IGES mapping. The reason being is that IGES needs both the pointers from the Associativity Instance to the member entities, and the pointer from the member entity back to the Associativity Instance. The Associativity Instance contains the needed information for any group in the CIS file. This presents no difficulty. I could set aside a storage area keeping record of the number of members in the group, and the pointers to the DE for each member. The problem arises in the fact that, according to the schema dictated by IGES for groups, back pointers to the Asso. Instance are required in each of the member entities, indicating membership. This implies I'd have to store all the PD for each member, and the Asso. Instance before writing them to the IGES file, to be able to fill in all the proper pointers.

See Figure 3 for the General Flowchart design. Listed is a more detailed description of each major process that's executed from the main routine.

SUBROUTINE INIT - Initialize: counters, pointers, and constants.

SUBROUTINE START - This will provide a man-readable prolog to the IGES file. I will decide on the appropiate message. The start sequence number will be incremented for each record written.

SUBROUTINE GLOBAL - This will contain the information describing the preprocessor, which is the Tektronix 4081 mini computer. Most of the values used here are taken from the 4081 Users Manuals. The values that are variable depending on the part being mapped are not previded by the user, but set values for my test case. The values used for the postprocessor information will be default values, since this is the preprocessor program. The global sequence number will be incremented for these 2 records.

Pass by the first CIS record.

SUBROUTINE READHR - This will read in the binary represented 22 bytes of header information and be converted into 17 integer variables, placed in common.

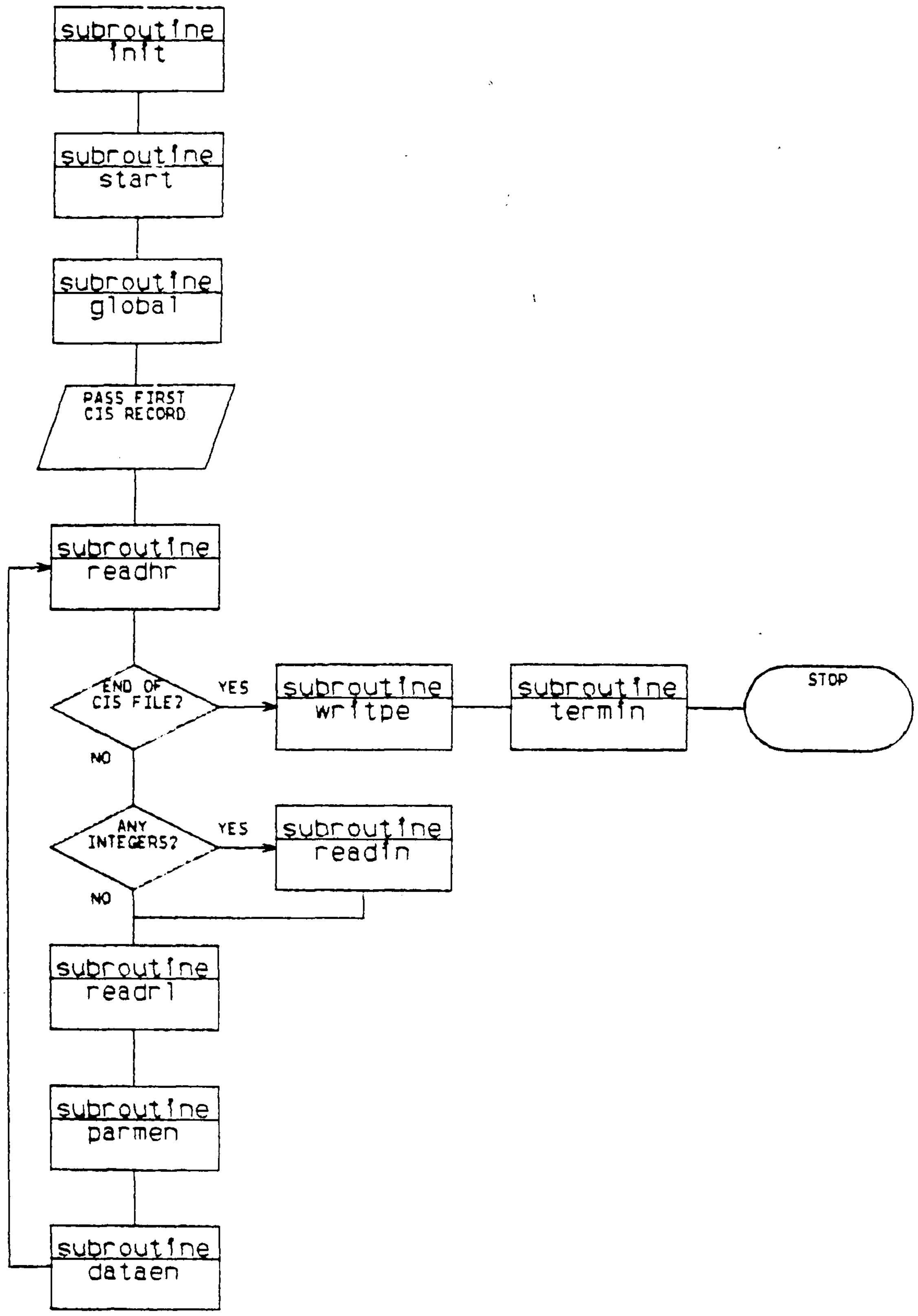

FIGURE 3. FLOWCHART OF THE MAJOR SUBROUTINES OF THE PREPROCESSOR

SUBROUTINE READIN - Based on the header information, the integer values, if any, will be read, converted from 10 byte ASCII character representation to integer, and stored in common.

SUBROUTINE READRL - Based on the header information, the proper number of real values will be read, converted from 20 byte ASCII representation to real and stored in common.

SUBROUTINE PARMEN - There will be a 'computed goto' to each of the entities to be mapped. Subroutines will be called from here to perform any special processing necessary for each entity (i.e. extensive processing for splines). The PE data array will be filled with values required for each entity. This information will then be stored to be written to the IGES file following all the DE records for each entity mapped. In other words, first come all the DE records, then all the PE records. If a form number is associated with the entity, it will be passed to SUB. DATAEN. The PD sequence number counter and pointer value for the DE entry will be incremented.

SUBROUTINE DATAEN - Using a data statement it will set the DE data values that remain the same regardless of the entity. The rest of the DE data array will be filled with entity related values, and the two DE records will be written to to the IGES file. The DE sequence number counter is incremented.

SUBROUTINE WRITPE - The PD data for the entire IGES file will be written, with back pointers that are required filled in.

SUBROUTINE TERMIN - The last sequence number used in all of the 5 sections is all that is written to the last record in the IGES file.

APPENDIX D
A TEST OF THE PREPROCESSOR

The model I designed to be mapped to IGES, comprises the following entities created as stated:

1. POINT - key-in at (3,5).
2. POINT - key-in at (3,7).
3. GROUP - indicated the 2 points.
4. POINT - key-in at (4,6).
5. LINE - connecting the 2 points at (3,5) and (3,7).
6. CIRCLE - thru the 3 points at (3,7), (4,6), and (3,5).
7. COMPOSITE- indicated the circle and line.
8. CONIC - key-in center at (4,4), starting angle = 0, ending angle = 360, and rotation angle = 0.
9. CIRCLE - screen position center and radius of .25.
10. SPLINE - thru 3 screen positioned points.
11. LINE - screen position.
12. LINE - screen position.
13. GROUP - indicated the 2 lines.

See Figure 4 for a view of the model.

Figures 5A-5G is a 'dump' of the CIS file created from my test part.

Figures 6A-6B is the IGES file my preprocessor created from the CIS file.

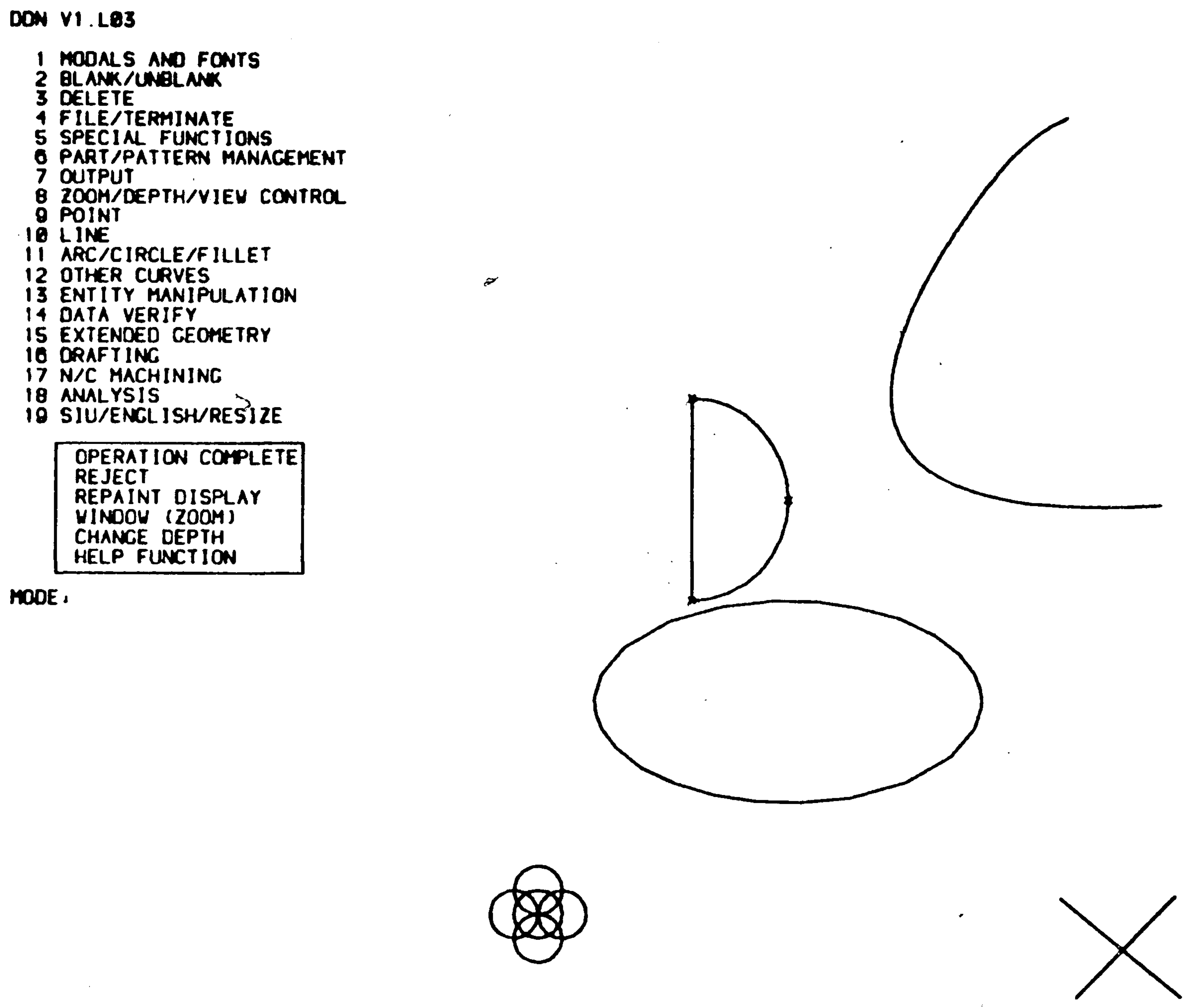

FIGURE 4. THE MODEL CREATED TO BE MAPPED TO IGES

```
****  COS File Dump  ****

     **** BLOCK 0000 , LENGTH 0100 ****

0000  434F 4D50 5554 4552 2049 4E44 4550 454E  4445 4E54 5454 5454 5454 5454 5454 5454   *COMPUTER INDEPENDENTTTTTTTTTTTTT*
0020  5454 5454 5454 5454 5454 5454 5454 5454  5454 5454 5454 5454 5454 5454 5454 5454   *TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT*
0040  5454 5454 5454 5454 5454 5454 5454 5454  5454 5454 5454 5454 5454 5454 5454 5454   *TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT*
0060  5454 5454 5454 5454 5454 5454 5454 5454  5454 5454 5454 5454 5454 5454 5454 5454   *TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT*
0080  5454 5454 5454 5454 5454 5454 5454 5454  5454 5454 5454 5454 5454 5454 5454 5454   *TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT*
00A0  5454 5454 5454 5454 5454 5454 5454 5454  5454 5454 5454 5454 5454 5454 5454 5454   *TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT*
00C0  5454 5454 5454 5454 5454 5454 5454 5454  5454 5454 5454 5454 5454 5454 5454 5454   *TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT*
00E0  5454 5454 5454 5454 5454 5454 5454 5454  5454 5454 5454 5454 5454 5454 5454 5454   *TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT*

     **** BLOCK 0001 , LENGTH 0100 ****

0000  0101 0100 0000 0300 0000 0000 0001 0000  0000 0000 003C 3C3C 3C3C 3C3C 3C3C 3C3C   *.....................<<<<<<<<<<<*
0020  3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C  3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C   *<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<*
0040  3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C  3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C   *<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<*
0060  3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C  3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C   *<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<*
0080  3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C  3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C   *<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<*
00A0  3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C  3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C   *<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<*
00C0  3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C  3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C   *<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<*
00E0  3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C  3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C   *<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<*

     **** BLOCK 0002 , LENGTH 0100 ****

0000  3030 3030 3030 3030 3033 2E30 3030 3030  3030 3030 3030 3030 3030 3030 3035 2E30   *0000000003.0000000000000000005.0*
0020  3030 3030 3030 3030 3030 3030 3030 3030  3030 2E30 3030 3030 3030 3030 3030 3030   *000000000000000000.0000000000000*
0040  3030 3030 3030 3030 3030 3030 3030 3030  3030 3030 3030 3030 3030 3030 3030 3030   *00000000000000000000000000000000*
0060  3030 3030 3030 3030 3030 3030 3030 3030  3030 3030 3030 3030 3030 3030 3030 3030   *00000000000000000000000000000000*
0080  3030 3030 3030 3030 3030 3030 3030 3030  3030 3030 3030 3030 3030 3030 3030 3030   *00000000000000000000000000000000*
00A0  3030 3030 3030 3030 3030 3030 3030 3030  3030 3030 3030 3030 3030 3030 3030 3030   *00000000000000000000000000000000*
00C0  3030 3030 3030 3030 3030 3030 3030 3030  3030 3030 3030 3030 3030 3030 3030 3030   *00000000000000000000000000000000*
00E0  3030 3030 3030 3030 3030 3030 3030 3030  3030 3030 3030 3030 3030 3030 3030 3030   *00000000000000000000000000000000*

     **** BLOCK 0003 , LENGTH 0100 ****

0000  0101 0100 0000 0300 0000 0100 0001 0000  0000 0000 003C 3C3C 3C3C 3C3C 3C3C 3C3C   *.....................<<<<<<<<<<<*
0020  3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C  3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C   *<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<*
0040  3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C  3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C   *<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<*
0060  3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C  3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C   *<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<*
0080  3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C  3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C   *<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<*
00A0  3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C  3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C   *<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<*
00C0  3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C  3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C   *<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<*
00E0  3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C  3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C   *<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<*
     ****  COS File Dump  ****

     **** BLOCK 0004 , LENGTH 0100 ****

0000  3030 3030 3030 3030 3033 2E30 3030 3030  3030 3030 3030 3030 3030 3030 3037 2E30   *0000000003.0000000000000000007.0*
0020  3030 3030 3030 3030 3030 3030 3030 3030  3030 2E30 3030 3030 3030 3030 3030 3030   *000000000000000000.0000000000000*
0040  3030 3030 3030 3030 3030 3030 3030 3030  3030 3030 3030 3030 3030 3030 3030 3030   *00000000000000000000000000000000*
0060  3030 3030 3030 3030 3030 3030 3030 3030  3030 3030 3030 3030 3030 3030 3030 3030   *00000000000000000000000000000000*
0080  3030 3030 3030 3030 3030 3030 3030 3030  3030 3030 3030 3030 3030 3030 3030 3030   *00000000000000000000000000000000*
00A0  3030 3030 3030 3030 3030 3030 3030 3030  3030 3030 3030 3030 3030 3030 3030 3030   *00000000000000000000000000000000*
00C0  3030 3030 3030 3030 3030 3030 3030 3030  3030 3030 3030 3030 3030 3030 3030 3030   *00000000000000000000000000000000*
00E0  3030 3030 3030 3030 3030 3030 3030 3030  3030 3030 3030 3030 3030 3030 3030 3030   *00000000000000000000000000000000*
```

FIGURE 5A. A 'DUMP' OF THE CIS FILE OF THE MODEL

```
        **** BLOCK 0005 , LENGTH 0100 ****

0000  0101 0100 0000 0300 0000 0200 0000 0000  0000 0000 003C 3C3C 3C3C 3C3C 3C3C 3C3C   *.....................<<<<<<<<<<<*
0020  3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C  3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C   *<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<*
0040  3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C  3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C   *<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<*
0060  3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C  3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C   *<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<*
0080  3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C  3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C   *<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<*
00A0  3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C  3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C   *<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<*
00C0  3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C  3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C   *<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<*
00E0  3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C  3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C   *<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<*

        **** BLOCK 0006 , LENGTH 0100 ****

0000  3030 3030 3030 3030 3034 2E30 3030 3030  3030 3030 3030 3030 3030 3030 3030 2E30   *0000000004.0000000000000000000.0*
0020  3030 3030 3030 3030 3030 3030 3030 3030  3030 2E30 3030 3030 3030 3030 3030 3030   *000000000000000000.0000000000000*
0040  3030 3030 3030 3030 3030 3030 3030 3030  3030 3030 3030 3030 3030 3030 3030 3030   *00000000000000000000000000000000*
0060  3030 3030 3030 3030 3030 3030 3030 3030  3030 3030 3030 3030 3030 3030 3030 3030   *00000000000000000000000000000000*
0080  3030 3030 3030 3030 3030 3030 3030 3030  3030 3030 3030 3030 3030 3030 3030 3030   *00000000000000000000000000000000*
00A0  3030 3030 3030 3030 3030 3030 3030 3030  3030 3030 3030 3030 3030 3030 3030 3030   *00000000000000000000000000000000*
00C0  3030 3030 3030 3030 3030 3030 3030 3030  3030 3030 3030 3030 3030 3030 3030 3030   *00000000000000000000000000000000*
00E0  3030 3030 3030 3030 3030 3030 3030 3030  3030 3030 3030 3030 3030 3030 3030 3030   *00000000000000000000000000000000*

        **** BLOCK 0007 , LENGTH 0100 ****

0000  0202 0100 0200 0000 0100 0300 0001 0000  0000 0014 00A0 A0A0 A0A0 A0A0 A0A0 A0A0   *................................*
0020  A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0  A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0   *................................*
0040  A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0  A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0   *................................*
0060  A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0  A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0   *................................*
0080  A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0  A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0   *................................*
00A0  A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0  A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0   *................................*
00C0  A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0  A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0   *................................*
00E0  A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0  A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0   *................................*
        ****  COS File Dump  ****


        **** BLOCK 0008 , LENGTH 0100 ****

0000  3030 3030 3030 3030 3030 3030 3030 3030  3030 3031 3131 3131 3131 3131 3131 3131   *00000000000000000001111111111111*
0020  3131 3131 3131 3131 3131 3131 3131 3131  3131 3131 3131 3131 3131 3131 3131 3131   *11111111111111111111111111111111*
0040  3131 3131 3131 3131 3131 3131 3131 3131  3131 3131 3131 3131 3131 3131 3131 3131   *11111111111111111111111111111111*
0060  3131 3131 3131 3131 3131 3131 3131 3131  3131 3131 3131 3131 3131 3131 3131 3131   *11111111111111111111111111111111*
0080  3131 3131 3131 3131 3131 3131 3131 3131  3131 3131 3131 3131 3131 3131 3131 3131   *11111111111111111111111111111111*
00A0  3131 3131 3131 3131 3131 3131 3131 3131  3131 3131 3131 3131 3131 3131 3131 3131   *11111111111111111111111111111111*
00C0  3131 3131 3131 3131 3131 3131 3131 3131  3131 3131 3131 3131 3131 3131 3131 3131   *11111111111111111111111111111111*
00E0  3131 3131 3131 3131 3131 3131 3131 3131  3131 3131 3131 3131 3131 3131 3131 3131   *11111111111111111111111111111111*

        **** BLOCK 0009 , LENGTH 0100 ****

0000  3030 3030 3030 3030 3030 2E30 3030 3030  3030 3030 3030 3030 3030 3030 3031 2E30   *0000000000.0000000000000000001.0*
0020  3030 3030 3030 3030 3030 3030 3030 3030  3033 2E30 3030 3030 3030 3030 3030 3030   *00000000000000000003.0000000000000*
0040  3030 3030 3035 2E30 3030 3030 3030 3030  3030 3030 3030 3030 3030 2E30 3030 3030   *000005.0000000000000000000.00000*
0060  3030 3030 3030 3030 3030 3030 3033 2E30  3030 3030 3030 3030 3030 3030 3030 3030   *00000000000003.00000000000000000*
0080  3037 2E30 3030 3030 3030 3030 3030 3030  3030 3030 3030 2E30 3030 3030 3030 3030   *07.0000000000000000000.000000000*
00A0  3030 3030 3030 3030 3030 3030 3030 3030  3030 3030 3030 3030 3030 3030 3030 3030   *00000000000000000000000000000000*
00C0  3030 3030 3030 3030 3030 3030 3030 3030  3030 3030 3030 3030 3030 3030 3030 3030   *00000000000000000000000000000000*
00E0  3030 3030 3030 3030 3030 3030 3030 3030  3030 3030 3030 3030 3030 3030 3030 3030   *00000000000000000000000000000000*
```

5B.

```
     **** BLOCK 000A , LENGTH 0100 ****

0000  0300 0100 0300 0000 0100 0400 0001 0000  0000 001E 0078 7878 7878 7878 7878 7878   *.....................xxxxxxxxxxx*
0020  7878 7878 7878 7878 7878 7878 7878 7878  7878 7878 7878 7878 7878 7878 7878 7878   *xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx*
0040  7878 7878 7878 7878 7878 7878 7878 7878  7878 7878 7878 7878 7878 7878 7878 7878   *xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx*
0060  7878 7878 7878 7878 7878 7878 7878 7878  7878 7878 7878 7878 7878 7878 7878 7878   *xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx*
0080  7878 7878 7878 7878 7878 7878 7878 7878  7878 7878 7878 7878 7878 7878 7878 7878   *xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx*
00A0  7878 7878 7878 7878 7878 7878 7878 7878  7878 7878 7878 7878 7878 7878 7878 7878   *xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx*
00C0  7878 7878 7878 7878 7878 7878 7878 7878  7878 7878 7878 7878 7878 7878 7878 7878   *xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx*
00E0  7878 7878 7878 7878 7878 7878 7878 7878  7878 7878 7878 7878 7878 7878 7878 7878   *xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx*

     **** BLOCK 000B , LENGTH 0100 ****

0000  3030 3030 3030 3030 3030 3030 3030 3030  3030 3032 3030 3030 3030 3030 3031 3131   *00000000000000000002000000000111*
0020  3131 3131 3131 3131 3131 3131 3131 3131  3131 3131 3131 3131 3131 3131 3131 3131   *11111111111111111111111111111111*
0040  3131 3131 3131 3131 3131 3131 3131 3131  3131 3131 3131 3131 3131 3131 3131 3131   *11111111111111111111111111111111*
0060  3131 3131 3131 3131 3131 3131 3131 3131  3131 3131 3131 3131 3131 3131 3131 3131   *11111111111111111111111111111111*
0080  3131 3131 3131 3131 3131 3131 3131 3131  3131 3131 3131 3131 3131 3131 3131 3131   *11111111111111111111111111111111*
00A0  3131 3131 3131 3131 3131 3131 3131 3131  3131 3131 3131 3131 3131 3131 3131 3131   *11111111111111111111111111111111*
00C0  3131 3131 3131 3131 3131 3131 3131 3131  3131 3131 3131 3131 3131 3131 3131 3131   *11111111111111111111111111111111*
00E0  3131 3131 3131 3131 3131 3131 3131 3131  3131 3131 3131 3131 3131 3131 3131 3131   *11111111111111111111111111111111*
     ****  COS File Dump  ****

     **** BLOCK 000C , LENGTH 0100 ****

0000  3030 3030 3030 3030 3034 2E37 3132 3330  3030 3330 3030 3030 3030 3030 3037 2E30   *0000000004.71230003000000000007.0*
0020  3533 3030 3130 3130 3030 3030 3030 3030  3033 2E30 3030 3030 3030 3030 3030 3030   *530010100000000003.0000000000000*
0040  3030 3030 3030 2E30 3030 3030 3030 3030  3030 3030 3030 3030 3030 2E30 3030 3030   *000000.0000000000000000000.00000*
0060  3030 3030 3030 3030 3030 3030 3031 2E30  3030 3030 3030 3030 3030 3030 3030 3030   *00000000000001.00000000000000000*
0080  3030 3030 3030 3030 3030 3030 3030 3030  3030 3030 3030 3030 3030 3030 3030 3030   *00000000000000000000000000000000*
00A0  3030 3030 3030 3030 3030 3030 3030 3030  3030 3030 3030 3030 3030 3030 3030 3030   *00000000000000000000000000000000*
00C0  3030 3030 3030 3030 3030 3030 3030 3030  3030 3030 3030 3030 3030 3030 3030 3030   *00000000000000000000000000000000*
00E0  3030 3030 3030 3030 3030 3030 3030 3030  3030 3030 3030 3030 3030 3030 3030 3030   *00000000000000000000000000000000*

     **** BLOCK 000D , LENGTH 0100 ****

0000  0001 0100 0300 0400 0000 0500 0000 0000  0000 001E 0050 5050 5050 5050 5050 5050   *.....................PPPPPPPPPPP*
0020  5050 5050 5050 5050 5050 5050 5050 5050  5050 5050 5050 5050 5050 5050 5050 5050   *PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP*
0040  5050 5050 5050 5050 5050 5050 5050 5050  5050 5050 5050 5050 5050 5050 5050 5050   *PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP*
0060  5050 5050 5050 5050 5050 5050 5050 5050  5050 5050 5050 5050 5050 5050 5050 5050   *PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP*
0080  5050 5050 5050 5050 5050 5050 5050 5050  5050 5050 5050 5050 5050 5050 5050 5050   *PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP*
00A0  5050 5050 5050 5050 5050 5050 5050 5050  5050 5050 5050 5050 5050 5050 5050 5050   *PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP*
00C0  5050 5050 5050 5050 5050 5050 5050 5050  5050 5050 5050 5050 5050 5050 5050 5050   *PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP*
00E0  5050 5050 5050 5050 5050 5050 5050 5050  5050 5050 5050 5050 5050 5050 5050 5050   *PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP*

     **** BLOCK 000E , LENGTH 0100 ****

0000  3030 3030 3030 3030 3032 2D30 3030 3030  3030 3034 3030 3030 3030 3030 3033 3333   *0000000002-000000004000000000333*
0020  3333 3333 3333 3333 3333 3333 3333 3333  3333 3333 3333 3333 3333 3333 3333 3333   *33333333333333333333333333333333*
0040  3333 3333 3333 3333 3333 3333 3333 3333  3333 3333 3333 3333 3333 3333 3333 3333   *33333333333333333333333333333333*
0060  3333 3333 3333 3333 3333 3333 3333 3333  3333 3333 3333 3333 3333 3333 3333 3333   *33333333333333333333333333333333*
0080  3333 3333 3333 3333 3333 3333 3333 3333  3333 3333 3333 3333 3333 3333 3333 3333   *33333333333333333333333333333333*
00A0  3333 3333 3333 3333 3333 3333 3333 3333  3333 3333 3333 3333 3333 3333 3333 3333   *33333333333333333333333333333333*
00C0  3333 3333 3333 3333 3333 3333 3333 3333  3333 3333 3333 3333 3333 3333 3333 3333   *33333333333333333333333333333333*
00E0  3333 3333 3333 3333 3333 3333 3333 3333  3333 3333 3333 3333 3333 3333 3333 3333   *33333333333333333333333333333333*
```

5C.

```
        **** BLOCK 000F , LENGTH 0100 ****

0000  3030 3030 3030 3030 3030 2E30 3030 3030  3030 3030 3030 3030 3030 3030 3034 2E30   *0000000000.00000000000000000004.0*
0020  3030 3030 3030 3030 3030 3030 3030 3030  3032 2E30 3030 3030 3030 3030 3030 3030   *00000000000000002.0000000000000*
0040  3030 3030 3034 2E30 3030 3030 3030 3030  3030 3030 3030 3030 3030 3030 3030 3030   *00004.00000000000000000000000000*
0060  3030 3030 3030 3030 3030 3030 3030 3030  3030 3030 3030 3030 3030 3030 3030 3030   *00000000000000000000000000000000*
0080  3030 3030 3030 3030 3030 3030 3030 3030  3030 3030 3030 3030 3030 3030 3030 3030   *00000000000000000000000000000000*
00A0  3030 3030 3030 3030 3030 3030 3030 3030  3030 3030 3030 3030 3030 3030 3030 3030   *00000000000000000000000000000000*
00C0  3030 3030 3030 3030 3030 3030 3030 3030  3030 3030 3030 3030 3030 3030 3030 3030   *00000000000000000000000000000000*
00E0  3030 3030 3030 3030 3030 3030 3030 3030  3030 3030 3030 3030 3030 3030 3030 3030   *00000000000000000000000000000000*

        ****  COS File Dump  ****


        **** BLOCK 0010 , LENGTH 0100 ****

0000  0402 0100 0200 0000 0000 0700 0000 0000  0000 0014 0104 0404 0404 0404 0404 0404   *................................*
0020  0404 0404 0404 0404 0404 0404 0404 0404  0404 0404 0404 0404 0404 0404 0404 0404   *................................*
0040  0404 0404 0404 0404 0404 0404 0404 0404  0404 0404 0404 0404 0404 0404 0404 0404   *................................*
0060  0404 0404 0404 0404 0404 0404 0404 0404  0404 0404 0404 0404 0404 0404 0404 0404   *................................*
0080  0404 0404 0404 0404 0404 0404 0404 0404  0404 0404 0404 0404 0404 0404 0404 0404   *................................*
00A0  0404 0404 0404 0404 0404 0404 0404 0404  0404 0404 0404 0404 0404 0404 0404 0404   *................................*
00C0  0404 0404 0404 0404 0404 0404 0404 0404  0404 0404 0404 0404 0404 0404 0404 0404   *................................*
00E0  0404 0404 0404 0404 0404 0404 0404 0404  0404 0404 0404 0404 0404 0404 0404 0404   *................................*

        **** BLOCK 0011 , LENGTH 0100 ****

0000  3030 3030 3030 3030 3032 3030 3030 3030  3030 3230 3030 3030 3030 3030 3030 3030   *00000000020000000020000000000000*
0020  3030 3030 3030 3030 3030 3030 3030 3030  3030 3030 3030 3030 3030 3030 3030 3030   *00000000000000000000000000000000*
0040  3030 3030 3030 3030 3030 3030 3030 3030  3030 3030 3030 3030 3030 3030 3030 3030   *00000000000000000000000000000000*
0060  3030 3030 3030 3030 3030 3030 3030 3030  3030 3030 3030 3030 3030 3030 3030 3030   *00000000000000000000000000000000*
0080  3030 3030 3030 3030 3030 3030 3030 3030  3030 3030 3030 3030 3030 3030 3030 3030   *00000000000000000000000000000000*
00A0  3030 3030 3030 3030 3030 3030 3030 3030  3030 3030 3030 3030 3030 3030 3030 3030   *00000000000000000000000000000000*
00C0  3030 3030 3030 3030 3030 3030 3030 3030  3030 3030 3030 3030 3030 3030 3030 3030   *00000000000000000000000000000000*
00E0  3030 3030 3030 3030 3030 3030 3030 3030  3030 3030 3030 3030 3030 3030 3030 3030   *00000000000000000000000000000000*

        **** BLOCK 0012 , LENGTH 0100 ****

0000  3030 3030 3030 3030 3030 2E30 3030 3030  3030 3030 3030 3030 3030 3030 3030 2E32   *0000000000.0000000000000000000.2*
0020  3033 3138 3330 3038 3030 3030 3030 3030  3030 2E30 3030 3030 3030 3030 3030 3030   *031830080000000000.0000000000000*
0040  3030 3030 3035 2E30 3030 3030 3030 3430  3030 3030 3030 3030 3034 2E30 3030 3030   *00005.00000000400000000004.00000*
0060  3030 3030 3030 3030 3030 3030 3035 2E30  3030 3030 3030 3430 3030 3030 3030 3030   *0000000000005.00000004000000000*
0080  3033 2E30 3030 3030 3731 3330 3030 3030  3030 3030 3034 2E30 3030 3030 3030 3030   *03.0000071300000000004.000000000*
00A0  3030 3030 3030 3030 3034 2E30 3030 3030  3030 3030 3030 3030 3030 3030 3032 2E30   *0000000004.00000000000000000002.0*
00C0  3030 3030 3030 3030 3030 3030 3030 3030  3031 2E30 3030 3030 3030 3030 3030 3030   *000000000000000001.0000000000000*
00E0  3030 3030 3030 2E30 3030 3030 3030 3030  3030 3030 3030 3030 3030 2E30 3030 3030   *000000.00000000000000000000.00000*

        **** BLOCK 0013 , LENGTH 0100 ****

0000  3037 3032 3232 3232 3232 3232 3232 3232  3232 3232 3232 3232 3232 3232 3232 3232   *07022222222222222222222222222222*
0020  3232 3232 3232 3232 3232 3232 3232 3232  3232 3232 3232 3232 3232 3232 3232 3232   *22222222222222222222222222222222*
0040  3232 3232 3232 3232 3232 3232 3232 3232  3232 3232 3232 3232 3232 3232 3232 3232   *22222222222222222222222222222222*
0060  3232 3232 3232 3232 3232 3232 3232 3232  3232 3232 3232 3232 3232 3232 3232 3232   *22222222222222222222222222222222*
0080  3232 3232 3232 3232 3232 3232 3232 3232  3232 3232 3232 3232 3232 3232 3232 3232   *22222222222222222222222222222222*
00A0  3232 3232 3232 3232 3232 3232 3232 3232  3232 3232 3232 3232 3232 3232 3232 3232   *22222222222222222222222222222222*
00C0  3232 3232 3232 3232 3232 3232 3232 3232  3232 3232 3232 3232 3232 3232 3232 3232   *22222222222222222222222222222222*
00E0  3232 3232 3232 3232 3232 3232 3232 3232  3232 3232 3232 3232 3232 3232 3232 3232   *22222222222222222222222222222222*

        ****  COS File Dump  ****
```

5D.

```
     **** BLOCK 0014 , LENGTH 0100 ****

0000  0301 0100 0000 0000 0100 0000 0000 0000  0000 0000 0078 7878 7878 7878 7878 7878   *.....................xxxxxxxxxxx*
0020  7878 7878 7878 7878 7878 7878 7878 7878  7878 7878 7878 7878 7878 7878 7878 7878   *xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx*
0040  7878 7878 7878 7878 7878 7878 7878 7878  7878 7878 7878 7878 7878 7878 7878 7878   *xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx*
0060  7878 7878 7878 7878 7878 7878 7878 7878  7878 7878 7878 7878 7878 7878 7878 7878   *xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx*
0080  7878 7878 7878 7878 7878 7878 7878 7878  7878 7878 7878 7878 7878 7878 7878 7878   *xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx*
00A0  7878 7878 7878 7878 7878 7878 7878 7878  7878 7878 7878 7878 7878 7878 7878 7878   *xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx*
00C0  7878 7878 7878 7878 7878 7878 7878 7878  7878 7878 7878 7878 7878 7878 7878 7878   *xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx*
00E0  7878 7878 7878 7878 7878 7878 7878 7878  7878 7878 7878 7878 7878 7878 7878 7878   *xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx*

     **** BLOCK 0015 , LENGTH 0100 ****

0000  3030 3030 3030 3030 3030 2E30 3030 3030  3030 3030 3030 3030 3030 3030 3030 2E32   *0000000000.0000000000000000000.2*
0020  3833 3138 3530 3035 3030 3030 3030 3030  3031 2E34 3235 3030 3031 3031 3030 3030   *831850050000000001.4250001010000*
0040  3030 3030 3031 2E38 3034 3238 3034 3233  3030 3030 3030 3030 3030 2E30 3030 3030   *000001.8042804230000000000.00000*
0060  3030 3030 3030 3030 3030 3030 3030 2E32  3430 3030 3030 3430 3030 3030 3030 3030   *00000000000000.2400000400000000*
0080  3030 3030 3030 3030 3030 3030 3030 3030  3030 3030 3030 3030 3030 3030 3030 3030   *00000000000000000000000000000000*
00A0  3030 3030 3030 3030 3030 3030 3030 3030  3030 3030 3030 3030 3030 3030 3030 3030   *00000000000000000000000000000000*
00C0  3030 3030 3030 3030 3030 3030 3030 3030  3030 3030 3030 3030 3030 3030 3030 3030   *00000000000000000000000000000000*
00E0  3030 3030 3030 3030 3030 3030 3030 3030  3030 3030 3030 3030 3030 3030 3030 3030   *00000000000000000000000000000000*

     **** BLOCK 0016 , LENGTH 0100 ****

0000  0000 0100 0400 0600 0000 0000 0000 0000  0000 0028 0078 7878 7878 7878 7878 7878   *...................(.xxxxxxxxxxx*
0020  7878 7878 7878 7878 7878 7878 7878 7878  7878 7878 7878 7878 7878 7878 7878 7878   *xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx*
0040  7878 7878 7878 7878 7878 7878 7878 7878  7878 7878 7878 7878 7878 7878 7878 7878   *xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx*
0060  7878 7878 7878 7878 7878 7878 7878 7878  7878 7878 7878 7878 7878 7878 7878 7878   *xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx*
0080  7878 7878 7878 7878 7878 7878 7878 7878  7878 7878 7878 7878 7878 7878 7878 7878   *xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx*
00A0  7878 7878 7878 7878 7878 7878 7878 7878  7878 7878 7878 7878 7878 7878 7878 7878   *xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx*
00C0  7878 7878 7878 7878 7878 7878 7878 7878  7878 7878 7878 7878 7878 7878 7878 7878   *xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx*
00E0  7878 7878 7878 7878 7878 7878 7878 7878  7878 7878 7878 7878 7878 7878 7878 7878   *xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx*

     **** BLOCK 0017 , LENGTH 0100 ****

0000  3030 3030 3030 3030 3030 3030 3030 3030  3030 3034 3030 3030 3030 3030 3032 3030   *00000000000000000004000000000200*
0020  3030 3030 3030 3031 3131 3131 3131 3131  3131 3131 3131 3131 3131 3131 3131 3131   *00000001111111111111111111111111*
0040  3131 3131 3131 3131 3131 3131 3131 3131  3131 3131 3131 3131 3131 3131 3131 3131   *11111111111111111111111111111111*
0060  3131 3131 3131 3131 3131 3131 3131 3131  3131 3131 3131 3131 3131 3131 3131 3131   *11111111111111111111111111111111*
0080  3131 3131 3131 3131 3131 3131 3131 3131  3131 3131 3131 3131 3131 3131 3131 3131   *11111111111111111111111111111111*
00A0  3131 3131 3131 3131 3131 3131 3131 3131  3131 3131 3131 3131 3131 3131 3131 3131   *11111111111111111111111111111111*
00C0  3131 3131 3131 3131 3131 3131 3131 3131  3131 3131 3131 3131 3131 3131 3131 3131   *11111111111111111111111111111111*
00E0  3131 3131 3131 3131 3131 3131 3131 3131  3131 3131 3131 3131 3131 3131 3131 3131   *11111111111111111111111111111111*
     ****  COS File Dump  ****


     **** BLOCK 0018 , LENGTH 0100 ****

0000  3030 3030 3030 3030 3031 2E34 3235 3030  3031 3031 3030 3030 3030 3030 3031 2E38   *0000000001.4250001010000000001.8*
0020  3537 3134 3334 3032 3030 3030 3030 3030  3030 2E30 3030 3030 3030 3030 3030 3030   *571434020000000000.0000000000000*
0040  3030 3030 3030 2E32 3430 3030 3030 3430  3030 3030 3030 3030 3030 2E30 3030 3030   *000000.2400000400000000000.00000*
0060  3030 3030 3030 3030 3030 3030 3031 2E35  3730 3730 3030 3133 3333 3333 3333 3333   *00000000000001.57070001333333333*
0080  3333 3333 3333 3333 3333 3333 3333 3333  3333 3333 3333 3333 3333 3333 3333 3333   *33333333333333333333333333333333*
00A0  3333 3333 3333 3333 3333 3333 3333 3333  3333 3333 3333 3333 3333 3333 3333 3333   *33333333333333333333333333333333*
00C0  3333 3333 3333 3333 3333 3333 3333 3333  3333 3333 3333 3333 3333 3333 3333 3333   *33333333333333333333333333333333*
00E0  3333 3333 3333 3333 3333 3333 3333 3333  3333 3333 3333 3333 3333 3333 3333 3333   *33333333333333333333333333333333*
```

5E.

```
     **** BLOCK 0019 , LENGTH 0100 ****

0000  0501 0100 0000 1000 0000 0A00 0000 0000  0000 003C 0100 B000 B000 B000 B000 B000   *...................<............*
0020  B000 B000 B000 B000 B000 B000 B000 B000  B000 B000 B000 B000 B000 B000 B000 B000   *................................*
0040  B000 B000 B000 B000 B000 B0B0 B000 B000  B000 B000 B000 B000 B000 B000 B000 B000   *................................*
0060  B000 B000 B000 B000 B000 B000 B000 B000  B000 B000 B000 B000 B000 B000 B000 B000   *................................*
0080  B000 B000 B000 B000 B000 B000 B000 B000  B000 B000 B000 B000 B000 B000 B000 B000   *................................*
00A0  B000 B000 B000 B000 B000 B000 B000 B000  B000 B000 B000 B000 B000 B000 B000 B000   *................................*
00C0  B000 B000 B000 B000 B000 B000 B000 B000  B000 B000 B000 B000 B000 B000 B000 B000   *................................*
00E0  B000 B000 B000 B000 B000 B000 B000 B000  B000 B000 B000 B000 B000 B000 B000 B000   *................................*

     **** BLOCK 001A , LENGTH 0100 ****

0000  3030 3030 3030 3030 3033 3030 3030 3030  3030 3134 3030 3030 3030 3030 3635 3030   *00000000030000000014000000006500*
0020  3030 3030 3030 3030 3030 3030 3030 3030  3030 3030 3030 3030 3030 3030 3030 3030   *00000000000000000000000000000000*
0040  3030 3030 3030 3030 3030 3030 3030 3030  3030 3030 3030 3030 3030 3030 3030 3030   *00000000000000000000000000000000*
0060  3030 3030 3030 3030 3030 3030 3030 3030  3030 3030 3030 3030 3030 3030 3030 3030   *00000000000000000000000000000000*
0080  3030 3030 3030 3030 3030 3030 3030 3030  3030 3030 3030 3030 3030 3030 3030 3030   *00000000000000000000000000000000*
00A0  3030 3030 3030 3030 3030 3030 3030 3030  3030 3030 3030 3030 3030 3030 3030 3030   *00000000000000000000000000000000*

00C0  3030 3030 3030 3030 3030 3030 3030 3030  3030 3030 3030 3030 3030 3030 3030 3030   *00000000000000000000000000000000*
00E0  3030 3030 3030 3030 3030 3030 3030 3030  3030 3030 3030 3030 3030 3030 3030 3030   *00000000000000000000000000000000*

     **** BLOCK 001B , LENGTH 0100 ****

0000  3030 3030 3030 3030 3030 2E30 3030 3030  3030 3030 3030 3030 3030 3030 3034 2E38   *0000000000.000000000000000004.8*
0020  3834 3031 3230 3033 3030 3030 3030 3030  3034 2E30 3834 3031 3230 3033 3030 3030   *840120030000000004.0840120030000*
0040  3030 3030 3030 2E30 3030 3030 3035 3732  3030 3030 3030 3030 3030 2E30 3731 3432   *000000.0000005720000000000.07142*
0060  3934 3433 3030 3030 3030 3030 3037 2E38  3530 3030 3033 3831 3030 3030 3030 3030   *94430000000007.85000381000000000*
0080  3035 2E39 3030 3731 3433 3430 3030 3030  3030 3030 3030 2E30 3030 3030 3030 3030   *05.900714340000000000.000000000*
00A0  3030 3030 3030 3030 3030 2E30 3030 3030  3030 3030 3030 3030 3030 3030 3030 2E30   *0000000000.0000000000000000000.0*
00C0  3030 3030 3030 3030 3030 3030 3030 3030  3030 2E39 3030 3030 3035 3732 3030 3030   *0000000000000000000.9000005720000*
00E0  3030 3030 3039 2E38 3731 3432 3934 3433  3030 3030 3030 3030 3033 2E30 3030 3339   *000009.8714294430000000003.00039*

     ****  GOS File Dump  ****


     **** BLOCK 001C , LENGTH 0100 ****

0000  3335 3437 2030 3030 3030 3030 3030 2E33  3337 3030 3030 3030 3030 3030 3030 3030   *3547-000000000.33700000000000000*
0020  3030 2E32 3033 3931 3835 3137 3030 3030  3030 3030 3031 2E33 3036 3030 3037 3930   *00.2039185170000000001.360000790*
0040  3030 3030 3030 3030 3035 2E39 3339 3238  3032 3332 3030 3030 3030 3030 3038 2E39   *0000000005.0392802320000000000.9*
0060  3835 3731 3439 3132 3030 3030 3030 3030  3035 2E32 3735 3734 3932 3837 2030 3030   *857149120000000005.275749287-000*
0080  3030 3030 3032 2E34 3834 3839 3138 3931  3030 3030 3030 3030 3031 2E38 3032 3530   *000002.4848918910000000001.80250*
00A0  3030 3030 3030 3030 3030 3030 3034 2E38  3834 3031 3230 3033 3333 3333 3333 3333   *00000000000004.88401200333333333*
00C0  3333 3333 3333 3333 3333 3333 3333 3333  3333 3333 3333 3333 3333 3333 3333 3333   *33333333333333333333333333333333*
00E0  3333 3333 3333 3333 3333 3333 3333 3333  3333 3333 3333 3333 3333 3333 3333 3333   *33333333333333333333333333333333*

     **** BLOCK 001D , LENGTH 0100 ****

0000  0201 0100 0000 0000 0000 0000 0001 0000  0000 0000 00A0 A0A0 A0A0 A0A0 A0A0 A0A0   *................................*
0020  A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0  A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0   *................................*
0040  A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0  A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0   *................................*
0060  A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0  A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0   *................................*
0080  A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0  A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0   *................................*
00A0  A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0  A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0   *................................*
00C0  A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0  A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0   *................................*
00E0  A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0  A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0   *................................*
```

5F.

```
        **** BLOCK 001E , LENGTH 0100 ****

0000  3030 3030 3030 3030 3030 2E30 3030 3030  3030 3030 3030 3030 3030 3030 3031 2E30    *0000000000.00000000000000000001.0*
0020  3030 3030 3030 3030 3030 3030 3030 3030  3030 2E30 3231 3432 3932 3533 3030 3030    *000000000000000000.0214292530000*
0040  3030 3030 3032 2E30 3339 3238 3030 3133  3030 3030 3030 3030 3030 2E30 3030 3030    *000002.03928000130000000000.00000*
0060  3030 3030 3030 3030 3030 3030 3030 2E30  3731 3432 3932 3533 3030 3030 3030 3030    *00000000000000.07142925300000000*
0080  3031 2E30 3137 3835 3735 3532 3030 3030  3030 3030 3030 2E30 3030 3030 3030 3030    *01.0178575520000000000.000000000*
00A0  3030 3030 3030 3030 3030 3030 3030 3030  3030 3030 3030 3030 3030 3030 3030 3030    *00000000000000000000000000000000*
00C0  3030 3030 3030 3030 3030 3030 3030 3030  3030 3030 3030 3030 3030 3030 3030 3030    *00000000000000000000000000000000*
00E0  3030 3030 3030 3030 3030 3030 3030 3030  3030 3030 3030 3030 3030 3030 3030 3030    *00000000000000000000000000000000*

        **** BLOCK 001F , LENGTH 0100 ****

0000  0201 0100 0000 0000 0000 0C00 0001 0000  0000 0000 00A0 A0A0 A0A0 A0A0 A0A0 A0A0    *................................*
0020  A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0  A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0    *................................*
0040  A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0  A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0    *................................*
0060  A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0  A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0    *................................*
0080  A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0  A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0    *................................*
00A0  A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0  A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0    *................................*
00C0  A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0  A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0    *................................*
00E0  A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0  A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0 A0A0    *................................*
        ****  COS File Dump  ****


        **** BLOCK 0020 , LENGTH 0100 ****

0000  3030 3030 3030 3030 3030 2E30 3030 3030  3030 3030 3030 3030 3030 3030 3031 2E30    *0000000000.00000000000000000001.0*
0020  3030 3030 3030 3030 3030 3030 3030 3030  3030 2E30 3037 3134 3330 3231 3030 3030    *000000000000000000.0071430210000*
0040  3030 3030 3032 2E30 3731 3432 3932 3533  3030 3030 3030 3030 3030 2E30 3030 3030    *000002.07142925300000000000.00000*
0060  3030 3030 3030 3030 3030 3030 3030 2E30  3032 3835 3730 3333 3030 3030 3030 3030    *00000000000000.00285703300000000*
0080  3031 2E30 3037 3134 3330 3231 3030 3030  3030 3030 3030 2E30 3030 3030 3030 3030    *01.0071430210000000000.000000000*
00A0  3030 3030 3030 3030 3030 3030 3030 3030  3030 3030 3030 3030 3030 3030 3030 3030    *00000000000000000000000000000000*
00C0  3030 3030 3030 3030 3030 3030 3030 3030  3030 3030 3030 3030 3030 3030 3030 3030    *00000000000000000000000000000000*
00E0  3030 3030 3030 3030 3030 3030 3030 3030  3030 3030 3030 3030 3030 3030 3030 3030    *00000000000000000000000000000000*

        **** BLOCK 0021 , LENGTH 0010 ****

0000  5448 4953 2049 5320 5448 4520 454E 442E                                             *THIS IS THE END.                *
0020                                                                                      *                                *
```

5G.

```
THIS IS THE IGES FILE CREATED BY A POST-PROCESSOR USING A               S0000001
 MEC COMPUTER INDEPENDENT FILE AS INPUT.                                S0000002
,,DK1.TEST.DAT,CIS,TEKTRONIX.V1.LØ2,1,16, 8,32, 8,64,DLKTEST,1.,1,INCH, G0000001
1,.Ø14Ø,Ø52881,12Ø533,Ø.,1Ø23.,D.L.KMETZ,B.S.CORP.                      G0000002
     116     1     1     1     Ø     1     Ø     Ø          ØDØØØØØØ1
     116     Ø     1     1     Ø     Ø     Ø  POINT         5DØØØØØØ2
     116     2     1     1     Ø     1     Ø     Ø          ØDØØØØØØ3
     116     Ø     1     1     Ø     Ø     Ø  POINT         5DØØØØØØ4
     4Ø2     3     1     1     Ø     1     Ø     Ø          ØDØØØØØØ5
     4Ø2     Ø     1     1     1     Ø     Ø  GROUP         5DØØØØØØ6
     116     4     1     1     Ø     1     Ø     Ø          ØDØØØØØØ7
     116     Ø     1     1     Ø     Ø     Ø  POINT         5DØØØØØØ8
     11Ø     5     1     1     Ø     1     Ø     Ø          ØDØØØØØØ9
     11Ø     Ø     1     1     Ø     Ø     Ø   LINE         4DØØØØØ1Ø
     1ØØ     6     1     1     Ø     1     Ø     Ø          ØDØØØØØ11
     1ØØ     Ø     1     2     Ø     Ø     Ø CIRCLE         6DØØØØØ12
     1Ø2     8     1     1     Ø     1     Ø     Ø          ØDØØØØØ13
     1Ø2     Ø     1     1     Ø     Ø     ØCOMPOSIT        8DØØØØØ14
     1Ø4     9     1     1     Ø     1     Ø     Ø          ØDØØØØØ15
     1Ø4     Ø     1     3     1     Ø     Ø  CONIC         5DØØØØØ16
     1ØØ    12     1     1     Ø     1     Ø     Ø          ØDØØØØØ17
     1ØØ     Ø     1     2     Ø     Ø     Ø CIRCLE         6DØØØØØ18
     112    14     1     1     Ø     1     Ø     Ø          ØDØØØØØ19
     112     Ø     1     6     Ø     Ø     Ø SPLINE         5DØØØØØ2Ø
     11Ø    2Ø     1     1     Ø     1     Ø     Ø          ØDØØØØØ21
     11Ø     Ø     1     1     Ø     Ø     Ø   LINE         4DØØØØØ22
     11Ø    21     1     1     Ø     1     Ø     Ø          ØDØØØØØ23
     11Ø     Ø     1     1     Ø     Ø     Ø   LINE         4DØØØØØ24
     4Ø2    22     1     1     Ø     1     Ø     Ø          ØDØØØØØ25
     4Ø2     Ø     1     1     1     Ø     Ø  GROUP         5DØØØØØ26
```

FIGURE 6A. THE MODEL AS MAPPED IN IGES

```
116,   3.00000,   5.00000,.00, 0,1,05;                                          1P0000001
116,   3.00000,   7.00000,.00, 0,1,05;                                          3P0000002
402, 2,03,01;                                                                   5P0000003
116,   4.00000,   6.00000,.00, 0;                                               7P0000004
110,   3.00000,   5.00000,.00,   3.00000,   7.00000,.00,1,13;                   9P0000005
100,.00,   3.00000,   6.00000,   3.00000,   5.00000,                           11P0000006
100,   3.00000,   7.00000,1,13;                                                11P0000007
102, 2,11,09;                                                                  13P0000008
104,   1.00000,   0.00000,   4.00000,  -4.00000, -16.00000,                    15P0000009
104,  76.00000,.00,   6.00000,   4.00000,   6.00000,   4.00000                 15P0000010
104;                                                                           15P0000011
100,.00,   1.42500,   1.86429,   1.67500,   1.86429,                           17P0000012
100,   1.67500,   1.86429;                                                     17P0000013
112,4,2,2, 2,                                                                  19P0000014
112, 0.00, 1.31,                                                               19P0000015
112,   6.90000,  -0.96366,   0.65319,  -4.67699,                               19P0000016
112,   9.87143,  -0.43005,   0.31618,  -7.25770,                               19P0000017
112,   5.93929,  -1.56685,   1.32472,  -5.02145,                               19P0000018
112,   8.98571,  -2.17247,  -1.16017,   4.79865;                               19P0000019
110,   6.82143,   2.03929,.00,   8.07143,   1.01786,.00,1,25;                  21P0000020
110,   8.00714,   2.07143,.00,   6.99286,   1.00714,.00,1,25;                  23P0000021
402, 2,23,21;                                                                  25P0000022
S0000002C0000002D0000026P0000022                                                 T0000001
#
```

6B.

VITA

Diane Louise Kmetz was born to Theresa and Joseph Donatelli on Aug. 26, 1955 in Roseto, Pa. She received her undergraduate degree in Mathematics from Bloomsburg State College, Bloomsburg Pa. in 1977 and accepted a programmer/analyst position at the Homer Research Laboratory of Bethlehem Steel Corporation, Bethlehem, Pa. She received her M.S. degree in Computer Science at Lehigh University, Bethlehem, Pa. in 1981.