A LOW-COST CAD SYSTEM FOR MANUFACTURED HOUSING

Cyril M. Coupal
University of Saskatchewan; Saskatoon, Saskatchewan

ABSTRACT

This paper describes a low-cost CAD system designed and implemented at Designex Buildings Limited, North Battleford, Saskatchewan. The system was intended to assist the architectural planning department in the preparation and verification of detailed factory-process specifications, in the form of drawings and printed listings. The system successfully increased the production of these specifications by a factor of four, while reducing the introduction of human-error to almost nil.

1.0 INTRODUCTION

The period from 1973 to 1977 saw tremendous growth in the Canadian Housing Industry. This rapid growth taxed the abilities of many manufacturing companies attempting to meet the demand for new housing. Major firms with numerous plants spread across Western Canada could forecast and easily surpass 2500 housing-unit sales in a single year. Although this annual figure averages to 6.84 manufactured units per day, the majority of these would be manufactured between the months of April and September raising the peak figure somewhat higher. Even during peak periods it was necessary to supply sold units within a reasonable amount of time (it was not uncommon during peak periods to wait six weeks for complete blueprints and a similar delay before taking delivery of the unit at the construction site). It was this fertile economic climate which inspired many innovative concepts in manufacturing in an attempt to increase productivity, product quality, and competitiveness.

Our company was small by comparison with larger firms, having production figures of 150 to 200 units per year during this time period. We required increased production simply to meet local housing demands, notwithstanding the corporate desire to expand into larger markets elsewhere. Increasing production was not simply a matter of adding additional work-crews and running the manufacturing plant on work shifts around the clock. Our plant had the capacity for 5 units per day but our design planning staff could only produce 2 complete blueprints per day (on average), including the detailed plant drawings (the most time-consuming to produce). Seasonal changes in demand made adding additional drafting personnel unfeasible. As a solution, we intended to produce only houses which appeared in our brochure either unchanged or with slight modifications. This philosophy would ease the design bottleneck since the same blueprint could be used repeatedly, but would alienate a large portion of the housing market oriented toward unique styles; designs not found in the average housing brochure. Rather, we chose to modernize our design technology through computerization and devised a number of corporate objectives:

* supply manufactured housing to that segment of the market not serviced by standard designs at standard design costs and efficiency;
* increase productivity to meet increased demand without a correlative rise in design costs;
* increase product quality by eliminating product faults due to human error (dimensioning, size-requirement look-ups, simple arithmetic calculations);
* integrate the design-construction process of cost and material estimation, drafting, and manufacturing.

The following discussion presents the working results of our efforts to meet these objectives.

2.0 OVERVIEW OF THE CAD SYSTEM

The system was developed drawing on our own experience in house-manufacturing, the needs of our particular plant operations, and through study of existing CAD systems in the U.S. and Scotland. Virtually all systems studied, although meritorious, were designed for large mini-computers or mainframe machines which placed the cost beyond our means.

We eventually chose to use a Datapoint micro-computer and attempt to design and implement our own functional system. The Datapoint 1500 stand-alone system has a dedicated programmable CRT, 36K of RAM memory, and two 8" diskette drives capable of a total capacity of 512K bytes storage. To this we added a medium-speed bi-directional line printer and a 36" Calcomp drum plotter. The total investment was approximately $50,000, making this system significantly less (at the time) then even the least expensive graphics-oriented computer.

A number of diskettes form the core of the system which is principally self-managing. Each diskette contains a number of programs performing related functions (data creation, maintenance, report generation) and data files which control system operation (user ID, log files, contract information, etc.).

User identification and security are maintained in a transparent manner. When a user is added to the system, a password and security level is assigned. The password allows initial access to the system while the security code restricts access to various parts of the system. (Should a user repeatedly fail to enter the correct password, the system will shut itself off.) A new user maybe given a low security level allowing data creation, but not modification or deletion. In this way a new user may obtain hands-on practice and not fear serious consequences due to errors. When he becomes more experienced, a higher security level can be assigned, allowing access to a greater range of systems functions.

A series of data-libraries are maintained by the system either automatically (system libraries) or through user interaction (construction libraries). A library is simply a random access (ISAM) file maintained on diskette. The system libraries are user transparent, i.e. they can not be printed, edited, or otherwise tampered with, although the user may cause an update to occur through normal processing paths. All other libraries are under (restricted) user control. Through the proper program functions, the user may create, modify, and delete library entries. For example, the FRAMING-MATERIAL library contains the material name (2X4) and dimensions (38x89 or 1-1/2"x3-1/2") for all allowable framing members.

2.1. THE USER INTERFACE

Since the micro-computer is a stand-alone system with a dedicated addressable CRT screen, we chose to implement the entire system as a menu-driven dialogue. The initial menu-screen is shown here:

1. SUPERVISOR OF OPERATIONS
2. APPLICATIONS OPERATOR

One may envisage the menu-structure as an n-ary tree. The ROOT node represents the initial menu-screen (which appears when the system is started). Each node has a number of branches equal to the number of selections available in the associated menu. Leaf nodes portray processing functions which may produce output or allow input creation (through display panels). All highlevel nodes are manipulated by a menu-processor using a function library. To add additional functions to the system one may simply edit this library, add the control information and pointers, then add the function program to the proper system diskette.

From a given menu-display, a user may make a numeric selection to progress further down the tree, or enter a sequence of '9's to return to the previous menu. Each branch of the tree depicts a partitioning of the system's functions; as one progresses down a particular branch-path, selections become more refined until a wanted function is available. In this way, a user need not remember a complex command-language syntax, nor worry about keyword or positional parameter specifications. All required arguments are requested via a message and input-space on the CRT. If a default is supplied, it is displayed allowing the user to accept it (by pressing the <ENTER> key) or over-ride the value by entering a new value.

We attempted to make the user interface as robust as possible: all user input is monitored and verified. Error detection (where practical) is immediate; the user is advised to retry befor performing subsequent data-entry steps. Where data-entry panels have multiple spaces, the user can retrace a portion or all of the entries to perform corrections. Special entries in any space will cause a portion of the screen to be re-initialized ('e' in alphanumeric spaces, '9' sequences in numeric spaces where '9'

would not be valid data).

Physical diskette manipulation (initialization, swapping between disk drives, etc.) is performed through a prompt-response-validate sequence. For example, should a diskette swap be required, a series of operation steps would be displayed. The system monitors each step as it is performed by the user. When the user enters his final response, a validation is performed. If any errors are detected, the sequence maybe repeated. A single control-function performs all necessary diskette manipulation (of this form) so that messages and the proper sequence become familiar to the user in a short time.

By using the above approach, we were able to impose housekeeping duties on the system rather than the user. The only significant duty the user must perform is a periodic back-up of all system diskettes (a weekly back-up would reflect all project additions and deletions made during the week). It is noteworthy that the diskette media (with the Datapoint drives) has proven to be a most reliable form of low-cost storage.

3.0 THE SYSTEM DATA MODEL

A house maybe interpreted as an horizontal 2-dimensional plane of co-ordinate points known as REFERENCE POINTS (REF) and coupling entities called PANELS (PNL). A two-storey house (or multi-level) would be processed as a series of horizontal 2-dimensional planes, each representing a single level. In addition, various descriptive elements called COMPONENTS (CMP) maybe created and assigned a location (by using a REF point).

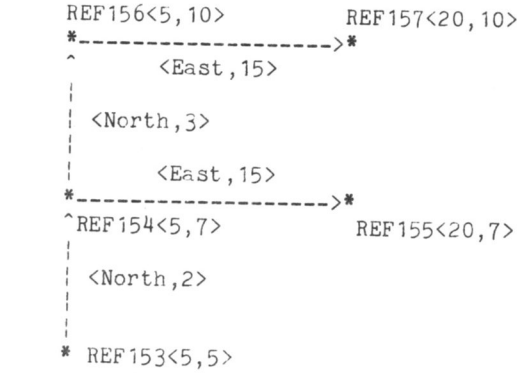
The basic data element is the REFERENCE POINT which labels a given co-ordinate in the floor plane. The house floor maybe equated to a cartesian co-ordinate system using only the <+x,+y> quadrant. A reference point requires three basic data elements for creation: a base reference point, a direction of travel, and a distance. The user may create a reference point at any location within a floor plan area to be used in later processing for locating other data elements. A base reference point is supplied initially, called 'PEG', at position <0,0>. All other reference points maybe created by a <direction,distance> specification from PEG or from another reference point already created. The system automatically generates REFnnn numbers and protects against duplicate entries freeing the user to concentrate on <direction, distance> values only.

Directions are designated by the cartographic directions North, South, East and West (N,S,E,W) derived from the practice of labeling architectural drawings (when labeling or identification was necessary, we turned to familiar architectural practices to make the system easier to learn).

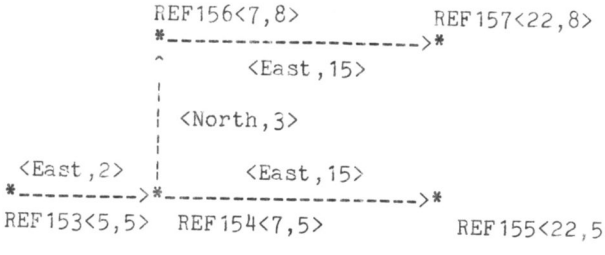
The house must exist such that its West-most wall aligns with the X=0 plane, while its South-most wall aligns with the Y=0 plane. This will ensure that the drawings produced during output generation will position properly on the frame.

Once all REFERENCE points are created, the system will begin a background process to convert all relational dimensions (REF points not created from PEG) into absolute dimensions. The data file will contain both the relational vectors and absolute dimensions. If no conflicts are detected the system returns control to the user who may now proceed with PANEL creation.

It is possible, by judicious specification, to move an entire block of REF points by adjusting only one <direction,distance> vector. For example:



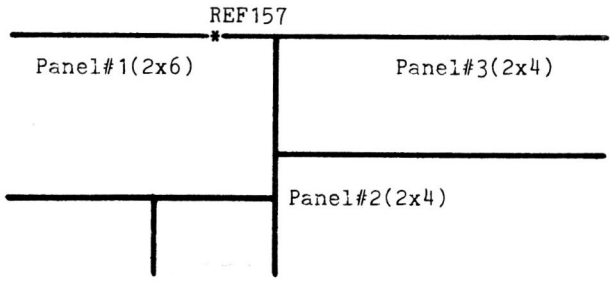
if the vector <North,2> between REF153 and REF154 is modified to <EAST,2> the result would be:



A second data element, the PANEL, is used to create wall sections. A panel is a manufactured unit which comprises a complete wall, and contains window/door openings, and stud-

groups for intersecting walls. Since many panels have equivalent characteristics, a global specification function allows the user to specify parameters such as stud-spacing, framing size, and wall height, once for an entire series. Should a parameter change, the same function can be invoked a second time. The actual panel entries are made by specifying the terminating REFERENCE points. The panel is automatically assigned a direction originating with the first REF point and terminating with the second REF point. If the panel is an exterior wall, it will be automatically aligned so that the exterior edge and the plane joining the two end REF points are allineated. For interior panels, it is the panel center-line which is allineated with the REF plane.

Each panel end may require adjustment if additional panels also terminate at the same REF point. To do this, a user may specify an OFFSET vector <direction, amount>. The direction values are the same as for REF creation (N,S,E,W). The amount is a nominal designation which maybe: none (-), panel-width (W) or panel half-width (C). In each case, the panel dimension used to calculate the OFFSET amount is the largest framing size to terminate at that REF point at 90 degrees to the current panel.



In this example, panel #1 is used to calculate the offset (South) of panel #2. However, it is the size of panel #2 which affects the offset (EAST) of both panels #1 and #3.

The final data element is the COMPONENT. This element is used to depict a construction-unit at some location within a panel. For example, windows and doors are considered as COMPONENTs: the opening plus all framing material used to create the opening. (In the case of windows/doors, the actual values are obtained from a library; the user simply calls the component by name and locates it at a given REF point). Framing groups used to back walls intersecting a given panel between its end-points are special COMPONENTs called POCKETS. Pockets are different from window/door components in that they are created by the user during the panel creation (not retrieved from a library). The user builds a POCKET one stud at a time until the desired configuration is made, then locates it at the correct REF point.

Each panel can contain any number of components so long as each component does not conflict with other components in the same panel. A background process (initiated automatically when the PANEL creation function is terminated) resolves all panels and components, similar to the REF resolution. Panel overlaps and incorrect component specifications (ex: at a REF which is not within a panel boundary) are detected during this process. Should no errors be found, the user is allowed to continue to the next phase which performs the actual output generation.

A practiced user with sound construction knowledge can easily prepare the data for an average bungalow (about 96.6 sq. meters, containing approximately 40 panels and 60 REF points) in about 1-1/2 hours.

4.0 OUTPUT FUNCTIONS

The CAD system produces various printed reports and plotted drawings. Printed reports maybe generated from libraries (component, material, project) to obtain current status and contents for onpoing maintenance and update. The reports used by the manufacturing plant come in three forms of printed listing and three forms of plotted drawings.

4.1. PRINTED REPORTS

A user may, at any time, generate a listing of the current input values for a house project. This listing produces a formatted copy of the original data file, without dimensional resolution. Should the diskette file become damaged or otherwise rendered useless, one can quickly re-create the data file by re-entering the values in this report. However, it has proven most utile during the actual data creation phase to verify what was entered when discrepancies are detected by the system without obvious causes.

Two printed reports are used in the manufacturing process. The MATERIAL LIST can be used to cut and sort material before actual panel assembly (Fig. 4.1.1.). It presents a summation, in order of increasing length and framing size, of all material needed to manufacture the wall sections for the house. This list includes wall plates (the horizontal members at the top and bottom of the wall), studs (regular, cripple,

and king) and header members (structural and non-structural spanning construction). All component material requirements are obtained from the component library. Framing requirements are determined by the panel characteristics and the panel framing-size specification (the panel framing size also overrides the default 2x4 size for component construction; a 2x6 wall will contain 2x6 components even though they were originally created as 2x4).

The second printed report presents a panel-by-panel decomposition (Fig. 4.1.2.). Each panel is listed on a separate page; overall panel characteristics (height, stud-spacing, length) form the page header. A number is assigned to every panel to assist in erecting the house on site (the floor-plan plot also lists panel numbers to indicate their location relative to each other). Following the header is a listing of all pieces required to build the panel, dimensioned from the left end of the panel (the left/right ends are determined from the direction vector assigned to the panel: a panel proceeds from left ----> right). The panel order is the same as the order in which they were created (i.e. although the panel numbers increase numerically, their relative position in the house relys on the original creation order).

The print processor is a function which uses previously resolved data files, computes and organizes the reports. Various temporary files, used by the processor, are removed when the reports are completed. In this way, the reports will consistently reflect all changes made to a house regardless of the previous status of the house data.

4.2. PLOTTED DRAWINGS

Because of our design approach and hardware configuration, the graphics function is a passive background process. The Datapoint 1500 is not powerful enough to drive an interactive graphics terminal (raster scan) although it maybe powerful enough to drive a storage-tube display. The processor CRT has no graphics capability. Consequently, all graphics output is produced on the Calcomp plotter.

One may generate drawings at any time during the data creation process. For example, after creating a number of REF points, one may wish to inspect their relative positions by comparing them to the source sketch or drawing. It maybe advantageous to obtain the resolved dimensions to assist in producing PANEL elements. The drawing in Fig. 4.2.1. is only one combination of possible output options available:

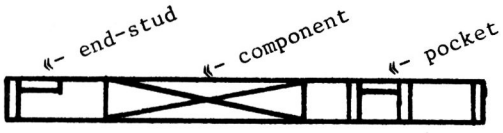
* cross hatch of reference points;

* label REF points;

* dimension REF points;

The user may select any combination of the above options depending on his drawing requirement (if all are rejected, an outline of the structure with no labeling or dimensioning is obtained. This is a useful drawing for preparing joist and roof-truss layouts, site plans, etc.).

Fig. 4.2.2. shows a floor plan plot which locates and labels all panels. Each panel contains end-stud configurations, pocket groups, and a cross-hatch of the area occupied by a component:



The floor-plan drawing maybe plotted with the following options:

* dimensions;

* panel-number labels;

* component labels;

The scale of the drawing is computed during the output process and as such allows for differently scaled drawings to be produced from the same house data file. Dimension conversion (from the internal file representation) is also performed at this time allowing the same data file to generate drawings with imperial (foot, inch) or metric (millimeter) dimensioning. For example, a house file maybe drawn at 1:48 scale with imperial dimensions, and at 1:50 scale with metric dimensions.

The graphics processor itself is a two-stage function. The first stage performs the conversions required by the user-selected options, and prepares a temporary file (which is removed when plotting is completed). The file contains ASCII data records; each record begins with an operation code followed by either character string data (dimension values, labels) or plotter parameters (distance increments, pen movement controls). The second stage of the function reads the plot file and simply interprets each record, performing the indicated plotting func-

tion. For example, a panel (simple rectangle) is designated by a particular numeric code followed by the numeric parameters for length, width, and orientation (N,S,E,W). From each record, the entity to be plotted is determined, and the proper control sequences are generated directly to the plot-controller interface. This is accomplished through various octal codes produced by the programming language PRINT statement (the plot controller shares the printer port).

By implementing a two-stage function, it is possible to interface other plotting hardware by adjusting the second stage decoding routine. However, the plotter hardware must be compatible with the Datapoint 1500 printer port, i.e. it must use a buffered input and accept data at 9600 baud (with the ability to signal buffer full when the buffer is within 10-20 characters of its actual limit). The Calcomp controller has a buffered input mechanism but does not signal when the buffer is almost full. For this reason we patched into the display panel LEDs which do indicate a buffer-full condition. However, this condition is raised only when the buffer is FULL. Therefore, each control sequence sent to the Calcomp controller is padded with octal 000s to ensure that valid data is properly captured. In this way the characters lost during the interval when the buffer is full and the 1500 detects the condition, are all null (000) and do not affect the resultant plotter output.

5.0 PROJECT DATABASE FILE STRUCTURE

A project database is maintained as an ISAM file, one project per diskette. The system catalogues all project diskettes in the project library (kept on a system diskette) and can retrieve a project using a user-supplied 4 character ID code (drawing number, factory order-number, etc.).

ISAM file records are accessed through a key composed of a data-descriptor (REF, PNL, CMP) and a system generated sequence number nnn. Although the key is created automatically as data is entered, the user may use the key to access information about a particular element. For example, by entering 'PNL 5' the user can inspect the current attributes of the fifth panel created.

The REFnnn record contains vector information about its location and what panels begin or terminate there. The information includes pointer keys to related records.

The PNLnnn record contains pointers to the REF records for its origin and terminal points along with various panel description fields (height, stud spacing, etc.).

The CMPnnnmm record contains various fields describing a particular component. Since a component exists within a panel, the CMP key is composed of the panel number (nnn) and a component sequence number (mm). Thus a single panel may have a maximum of 99 components.

The ISAM file is managed by the Datapoint DOS system releaving application programs from all but simple I/O routines. Periodically the CAD system will determine that the ISAM file has deteriorated (due to excessive updates) and will invoke a DOS utility to restructure the file. From the user's perspective, it means a 5 minute wait during which he may get a cup of coffee and be totally beyond reproach from his supervisors.

ISAM creation/restructuring and file sorting are processes serviced by DOS utilities. To invoke a DOS utility, the application program is suspended and a DOS call is issued. When the utility is finished, the application program resumes execution at the call point. Since each utility causes a suspension in terminal activity, the processes have been located such that the user is not interrupted during a task (such as data creation). The utility steps are usually invoked between system functions (ex. when REF point creation is finished and PNL creation is about to begin).

6.0 SUMMARY

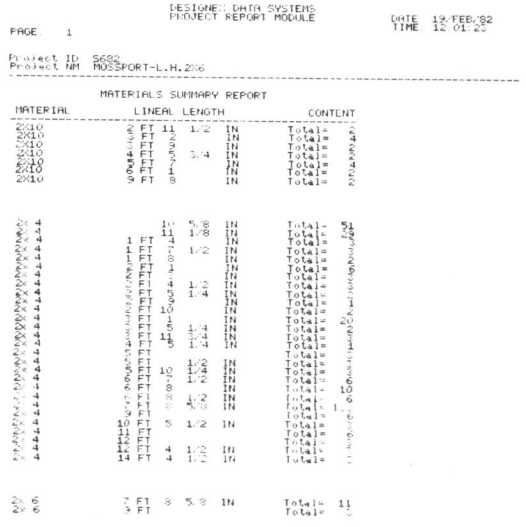
The full development and implementation process spanned two years (1976/77 - 1978/79). During that time the system underwent many revisions and modifications, stabilizing into the present configuration. The original concepts of system architecture have proven to be sound ones. A menu-driven user interface simplified learning the system dialogue, detecting and correcting errors, and providing user assistance. The user's guide can devote space to help develop a user concept of the system (without any knowledge of computers or programming) and yet provide step-by-step guidance through various functions. The present system has been in full production for over two years and has met our original expectations.

One fault with the system is that the 1500 processor is too small for an industrial environment. The Datapoint 1500 was originally offered as a sophisticated stand-alone data-entry

system which could front-end a larger mini or mainframe computer. Hence certain processes of the CAD system (all background) require more time than is desireable (all interactive processes such as error detection, data retrievals, etc., are quite fast). Obvious through hindsite, a better approach would be to feed all prepared data into a larger micro or mini computer which could service multiple work stations (possibly through a local area network) and produce printed reports and plotted drawings concurrently. The cost of such a configuration was prohibitive, especially since we originally conceived the project as an experiment.

The experiment proved successful and, although installed in only the single site, continues to function well. Through careful redesign of certain CPU-bound processes, and the addition of new lower-cost graphics hardware, we maybe able to extend the abilities of the system to provide interactive graphics at all levels of the design phase, generate material takeoffs and pricing estimates for the entire structure, and generally make the system more comprehensive. The ground-work has been done.

****



DESIGNEX DATA SYSTEMS
PROJECT REPORT MODULE

PAGE 1

DATE 19/FEB/82
TIME 12 01 23

Project ID: S682
Project NM: MOSSPORT-L.H.2X6

MATERIALS SUMMARY REPORT

MATERIAL	LINEAL LENGTH	CONTENT
2X10	6 FT 11 1/2 IN	Total= 2
2X10	3 FT 2 IN	Total= 4
2X10	3 FT 9 IN	Total= 2
2X10	4 FT 6 3/4 IN	Total= 2
2X10	7 FT 7 IN	Total= 4
2X10	6 FT 1 IN	Total= 2
2X10	9 FT 8 IN	Total= 2
2X 4	10 5/8 IN	Total= 51
2X 4	11 1/8 IN	Total= 24
2X 4	1 FT 4 IN	Total= 3
2X 4	1 FT 7 1/2 IN	Total= 3
2X 4	1 FT 8 IN	Total= 3
2X 4	2 FT 1 IN	Total= 6
2X 4	2 FT 4 IN	Total= 3
2X 4	2 FT 4 1/2 IN	Total= 3
2X 4	2 FT 5 1/4 IN	Total= 3
2X 4	2 FT 8 IN	Total= 1
2X 4	2 FT 10 IN	Total= 2
2X 4	3 FT 1 IN	Total= 20
2X 4	3 FT 5 1/4 IN	Total= 6
2X 4	3 FT 11 3/4 IN	Total= 6
2X 4	4 FT 5 1/4 IN	Total= 1
2X 4	5 FT	Total=
2X 4	5 FT 1 1/2 IN	Total=
2X 4	5 FT 10 1/4 IN	Total=
2X 4	6 FT 7 1/2 IN	Total= 6
2X 4	6 FT 8 IN	Total= 10
2X 4	7 FT 8 1/2 IN	Total= 6
2X 4	7 FT 8 5/8 IN	Total= 1
2X 4	9 FT	Total= 6
2X 4	10 FT 5 1/2 IN	Total= 3
2X 4	11 FT	Total= 6
2X 4	12 FT	Total= 4
2X 4	13 FT 4 1/2 IN	Total= 2
2X 4	14 FT 4 1/2 IN	Total= 1
2X 6	7 FT 8 5/8 IN	Total= 11
2X 6	9 FT	Total= 5

Fig. 4.1.1: Material List Report

DESIGNEX DATA SYSTEMS
PROJECT REPORT MODULE

PAGE 26

DATE 19/FEB/82
TIME 12 01 23

Project ID: S682
Project NM: MOSSPORT-L.H.2X6

Panel No 24 Type INTERIOR Material: 2x 4
Length 14 FT 4 1/2 IN
Spacing 1 FT 4 IN OC
Height Left= 8 FT 1 1/8 IN Right= 8 FT 1 1/8 IN
Plates Top= 2
Btm= 1
Studs 7 @ 7 FT 8 5/8 IN

Left

Edge
Flat Up 1 1/2 IN

Pocket

Edge 2 FT 3 IN
Flat Up 2 FT 4 1/2 IN

Component # 1 @ 2 FT 8 IN
2 King-studs < 2x 4 > 7 FT 8 5/8 IN
0 Cripples < 2x 4 > 6 FT 8 1/2 IN
1 Header < 2x 4 > 1 FT 8 IN
3 Cripple/top < 2x 4 > 10 5/8 IN

Pocket

Edge 4 FT 7 1/2 IN
Flat Up 4 FT 9 IN
Edge 5 FT 1/2 IN

Pocket

Edge 11 FT 7 IN
Flat Down 11 FT 8 1/2 IN
Flat Up 11 FT 8 1/2 IN
Edge 12 FT

Right

Flat Down 13 FT 11 1/2 IN
Edge Dado 14 FT 3 IN

Fig. 4.1.2: Panel Decomposition Report

(Printed reports and Plotted drawings courtesy of
Designex Buildings Ltd., North Battleford, Sask.)

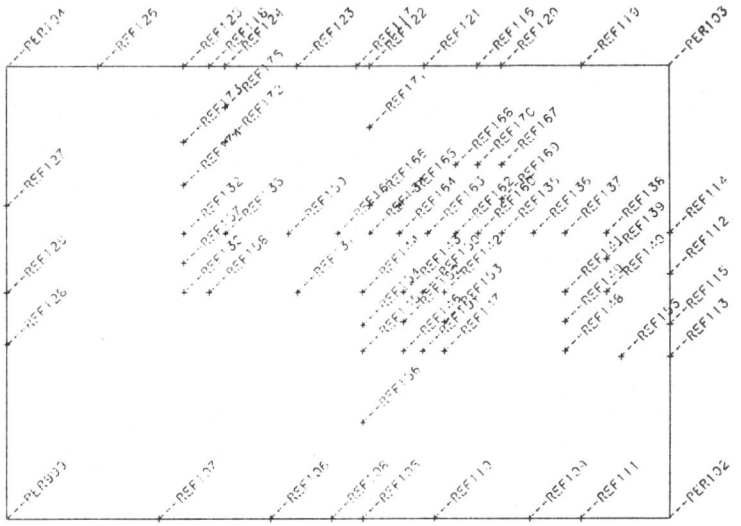


Fig. 4.2.1: Reference Point Plot.

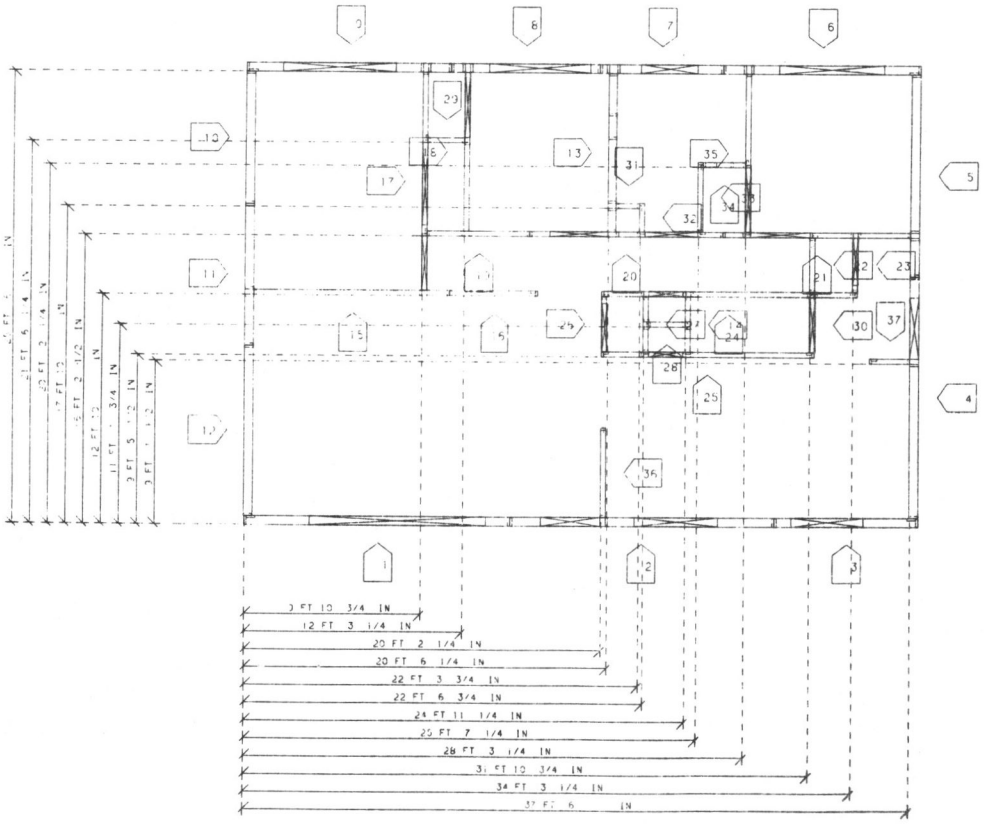


Fig. 4.2.2: Floor Plan Plot.