ABSTRACT

The primary problem addressed by this project is to provide the Goldston Corporation construction management team with a real-time stand alone purchasing system which marries the traditional accounting data base with data files outside the normal realm of the accounting EDP systems. Secondary problems of time delay and invoice processing errors are also addressed.

The development of logical relationships between the traditional accounting accounts payable data base, a purchase order master file and a Cost Engineering Commitment file, make available to the Project Management team vital information not currently available from any other source, that is the amount of unpaid commitments. The utilization of computerized system at the field level will reduce the critical problem of the "Time Lag" between actual events and typical corporate reports. While also greatly increase the accuracy of "downstream" accounts payable verification and processing.

This system is designed to enforce good purchasing control technique through it's processing structure and interactive nature. It also has future potential to interact with Budget data files and Vendor master files to provide data output not currently available from any other systems.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>SECTION</th>
<th>PAGE NUMBERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>I.</td>
<td>1</td>
</tr>
<tr>
<td>II.</td>
<td>2</td>
</tr>
<tr>
<td>III.</td>
<td>2a</td>
</tr>
<tr>
<td>IV.</td>
<td>3-10</td>
</tr>
<tr>
<td>V.</td>
<td>11-23</td>
</tr>
<tr>
<td>VI.</td>
<td>24-29</td>
</tr>
<tr>
<td>VII.</td>
<td>30</td>
</tr>
<tr>
<td>VIII.</td>
<td>A1-A16</td>
</tr>
<tr>
<td></td>
<td>B1-B3</td>
</tr>
<tr>
<td></td>
<td>C1-C3</td>
</tr>
<tr>
<td></td>
<td>D1-D3</td>
</tr>
<tr>
<td></td>
<td>E1</td>
</tr>
<tr>
<td></td>
<td>F1-F4</td>
</tr>
<tr>
<td></td>
<td>G1-G5</td>
</tr>
<tr>
<td></td>
<td>H1-H2</td>
</tr>
</tbody>
</table>
IV. INTRODUCTION

A. THE NATURE OF THE PROBLEM:

1. GENERAL BACKGROUND: In the construction industry, computerized "SYSTEMS" have historically been the domain of the corporate executive offices, with day to day control strictly enforced thru the corporate accounting function. There were valid reasons for this organizational philosophy. The early hardware was, by today's standards, physically large, of small logical capacity, batch oriented, and expensive. Inevitably, these early systems were initially purchased to process corporate accounting data, generally perceived as the most pressing need due to the large numbers of transactions. The rigid discipline of an accounting function, also lent itself nicely to specific application programming. Most early EDP systems were really a set of uncommunicating or semi-communicating application programs, with batch transactions, two to three week lag times, and strict data entry procedures allowing little or no manipulation of data. These "early" accounting systems were very accurate at the expense of timeliness and data utility. They also opened up visions to top management of collecting and presenting for management scrutiny large quantities of data. Management then demanded, and generally received, massive new levels of accounting reports. Due to the nature of the accounting oriented systems, top management quickly found these "massive" data reports to be of little real value.
In the construction industry, the most potentially productive use of computerized systems would provide timely data on construction project performance. Initially the accounting system was utilized for these reports. At best, due to the nature and primary requirements of the accounting function, these evaluations proved to be historical evaluations. The timely data available for making daily management decisions on actual project performance still resided in the mind of the project superintendent, and in his intuitive interpretations of the many variables encompassing the construction plans and schedule. Because accounting cost data must trail the actual performance which incur those costs, the utilization of accounting data at any point in time often leads to erroneous assumptions and poor project management decisions unless this cost data is reconstructed in a manner which permits valid comparisons to job performance at a known instant in time. Of prime importance was the development of a construction system which was not only accurate, but concise, pertinent, and forward-looking.

In light of the revised criteria, management began looking beyond mere recompilations of their accounting systems. The need was perceived for a separate and distinct construction management system which would provide real-time cost and performance evaluations, based on often incomplete data. This would make feasible prompt management decisions to halt or reverse poor performance for the balance of a construction project. This construction management systems must be able to accurately evaluate construction performance on the
basis of four basic subsystem: labor costs, material costs, subcontract costs, and equipment costs. Each subsystem has numerous individual "cost pipelines", which will vary according to individual companies philosophies on the accounting definition of direct, indirect, and overhead costs.

While heavily "borrowing" from the accounting data base, accurate and timely forecasting requires the development of new data files outside the traditional scope of the accounting data base. Also the construction management system must employ "non real data" (i.e. data not based upon historical fact but merely based upon some objective evaluation process which has known probabilities of being erroneous).

The simplified theory behind the calculations in each major subsystem is an evaluation of actual accounting cost-to-date, contractually committed cost-to-date and performance to-date in such a manner that a reasonable assessment is made of 1) performance-to-date and 2) final projected costs and revenues at completion. The evaluation is made at the lowest functional levels and is summarized through a logical tier structure enabling project management to quickly isolate major variances, predict their forecasted future effects, and determine the root causes of these variances thru utilizing the report's hierarchical structure.

(See Exhibit 1)
DIVISIONAL PROJECTS
STATUS REPORT

JOB COST SUMMARY REPORT
BY FUNCTION CLASSES & AREAS

PROJECT JOB COST COMMITMENT REPORT
BY INDIVIDUAL COST CODE

LABOR SUBSYSTEM  EQUIPMENT SUBSYSTEM  MATERIAL SUBSYSTEM  SUBCONTRACT SUBSYSTEM

FORECASTS

Various "Cost Pipelines"
Actual Costs
Committed Costs

Evaluation Process Against

Budgeted Costs
Total Work to be Performed

Work Performed to-Date

Evaluation Performed Within Individual Cost Codes

EXHIBIT I  CONSTRUCTION MANAGEMENT
CONCEPTUAL MODEL

-5A-
2. Goldston Corporation Specific Background. The general background previously described is accurate in respect to The Goldston Corporation. In late 1980 and early 1981 the corporation decided to pursue the above philosophies and actively develop the construction management system as schematically represented in Exhibit II. Thru early 1982 much was accomplished towards fulfilling Exhibit II. The complex labor reporting system was designed and implemented and is now routinely used. New purchasing procedures and new purchase order forms were developed to enhance control and accountability, and a purchase order master file was designed for the IBM SYSTEM 34 to capture all purchasing information. Logical reports from this purchase order master file provide material and subcontract commitment reports by project and by corporate general ledger overhead codes. However, these commitment reports do not include any payment information. A computerized inventory system was never implemented, and the project cost engineers are still using the accounting department’s equipment reports. Job Cost Comparison Reports are being manually developed with the information being manually transferred from the various lower tier reports. These manual forecasts based upon a partially automated system are providing an adequate degree of accuracy for the current recessionary levels of operations. For this reason, further development of corporate Construction Management systems have been temporarily suspended. But there exist several areas potentially serious errors could occur, especially in large complicated fast track projects with massive daily
transactions. The current system has no cross referencing between accounts payable and purchase order commitments. The overlap of these two data streams must be known to determine unpaid commitments. The calculation is vital in accurate forecasting within the material and subcontract subsystems. These large construction projects often by necessity often have their own purchasing and clerical staffs in order and maintain pace with the tremendous inflow of paper documents. These documents are all processed in a manual paper-file operation. Errors and omission can and do occur too frequently, especially when the personnel involved are not often familiar with good purchasing procedures. It is for this type of large construction project, with a semi-closed environment, that this system was developed.

IV. E. APPROACH TO THE SOLUTION

The system developed for this project is designed to meet the specific application niche described above. What is needed is a stand alone, low-cost system that would solve these major problems presently existing in the Goldston project management environment:

1. The inability to match vendor invoice payments against
   A. Individual purchase orders
   B. cost codes within purchase orders. (This will greatly increase the accuracy & reliability of the cost engineering functions.)

2. Speed up the processing and coding of:
   A. Purchase Orders
   B. Vendor applications for payments

3. Increase the accuracy of the field purchasing function through the system requirements of good purchasing techniques.
The approach chosen to achieve solutions to the above problems was to develop a microcomputer based system using the IBM Personal Computer. (This particular microcomputer system was chosen primarily because it was currently available within the Goldston Corporation).

This project involves the systematic, logical integration of several data base files. The BASIC program which produces this integration is user-friendly, it utilizes a menu driven concept to lead the user through whatever operation he selects. The system program automatically provides the integration between the data files at the point of data entry in a manner invisible to the user. The user can inquiry access the data files at any time, with the system directly accessing that data requested. The field management team can input and edit data on purchase order, payments and cost control at any time, with standard system checks at the point of data entry to eliminating typical omission and coding errors.

The major drawback to the use of this particular IBM equipment was the single disk drive containing only 150,000 bytes of information. (This is relatively small by most standards.) The optimum approach would have all purchase order information, and the associated cost code files and invoice files "on line" and always directly accessible to inquiry. The tiny IBM diskette size prohibits immediate access to all purchase orders at all times. However, most of the size limitation involved can be overcome with the good operational procedures. Most construction companies have several types of purchase
orders, each designed for distinct purchasing functions.

Goldston uses 3 types of purchase orders:

Type "A" Major Material Purchase Orders- a seven part contractual purchase order for purchases over $2500.00.

Type "B" Field Purchase Order - a four part Purchase Order with limited contractual conditions to be used for rapid purchases of under $2000.00.

Type "C" Subcontract Purchase Order—a seven part purchase order with complete subcontract terms and conditions. These Purchase Orders are pre-numbered for control according to the following schema:

<table>
<thead>
<tr>
<th>DIV NO.</th>
<th>PC TYPE</th>
<th>PO SEQUENCE NO:</th>
</tr>
</thead>
</table>

i.e. 7B-0008 is Type "B" Purchase Order from construction division no. 7 PO sequence no. 0008

For control purposes, these Purchase Orders are issued to divisional buyers in sequential blocks of 500 purchase orders each, with the divisional project buyer responsible to account for each F.O. issued. A large project would be issued sequential "blocks" of each PO type. It is these project issued PO "blocks", and their associated cost code and invoice records, which this system interactively manages. The system maintains complete track of all Purchase Orders of a particular type on one diskette, along with the directly associated invoice (accounts payable) records and cost code records. The system uses the Purchase Order sequence number as the system "key" providing direct system access to the Purchase Order Master file. Then, according to the operation interactively selected by the user, the system also indirectly accesses the invoice and cost code records associated with a Purchase Order record.
IV. C. References and Bibliography

1. IBM "BASIC" Software Reference Manual

2. IBM "DOS" Software Reference Manual
V. DETAILS OF THE PROJECT

A. LOGICAL DESIGN. The logical design of the P.O. system is shown in Exhibit III. It involves the construction of the random access (direct access) primary data files:

1. Purchase Order Master file
2. Invoice file
3. Cost Code file

The Purchase Order Master file is the logical master for the entire system. It uses the P.O. sequence number as the key to the individual Purchase Order records. It also contains pointer fields to both the invoice file and the cost code file. These pointers create the "logical" files of invoices for each Purchase Order and cost codes for each purchase order. These logical relationships are automatically built and maintained by always accessing the other two files through the Purchase Order Master file. The Purchase Order Master file also continually maintains summary information of the invoice records and of cost codes records which are logically subsidiary to the Purchase Order Master file. It should be noted here that the pointer chains are forward link pointers. The P.O. master file has two pointer fields which identify the first invoice or first cost code associated with that purchase order. Each invoice record or cost code record then "points" to the next invoice or cost code associated in the logical "chain". The last record in the chain has an END-OF-FILE indicator. There is no backward link pointer chains. Logical security is provided by burying the relevant Purchase Order number in each record in both the invoice and the cost code files. This would make recovery
EXHIBIT III-LOGICAL DESIGN PURCHASE ORDER MANAGEMENT SYSTEM

PURCHASE ORDER
MASTER FILE

P.O. #

INVPTR

CCDPtr

INVOICE MASTER
FILE

INVOICE RECORD# P.O.# VENDOR INVOICE#

COST CODE MASTER FILE

COST CODE RECORD# P.O.# COST CODE NUMBER
possible, if for some reason, the logical pointer chains were broken. See Exhibit IV for a list of the data fields in the Purchase Order Master file.

The Invoice file is physically a chronological file, with invoices added in sequence in the order of input. Except when the entire invoice file is printed, invoices are always accessed through the Purchase Order Master file. While technically a random file with direct access, the "key" to these random records is an invoice number (assigned at input). It is this invoice number which the pointer fields contain.

The Cost Code file is logically set up and maintained in a similar fashion to the invoice file. It consists of multiple subsets of cost code records "belonging" to the various Purchase Orders in the Purchase Order Master file. The purchase order record contains the amount and sales summary information for all the cost code records associated with that specific purchase order record.

The program logic is comprised of main module centered upon a main menu of eleven standard operations. Upon entering the program, the 6 files are opened and the buffer fields in each are defined to the system. The user is immediately shown the menu and is prompted to make his choice of which subroutine function he desires to perform. Upon entering his choice, the program then "walks" the user through the desired operation, prompting the user where necessary. Upon finishing any operation, the user is always returned to the main menu. The user can only exit the program from th
### EXHIBIT IV

**FILE NO 1**  
**PURCHASE ORDER MASTER FILE**  
"FO MASTR.DAT"  
**LENGTH=120**

<table>
<thead>
<tr>
<th>FIELD</th>
<th>NAME</th>
<th>SIZE</th>
<th>DATA</th>
<th>DATA</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>74</td>
<td>91</td>
<td>1</td>
<td>ALPHA</td>
<td>----</td>
<td>RECORD USES INDICATOR</td>
</tr>
<tr>
<td>75</td>
<td>DIVNO</td>
<td>2</td>
<td>NUMERIC</td>
<td>INTEGER</td>
<td>DIVISION NUMBER</td>
</tr>
<tr>
<td>76</td>
<td>PURCHOR</td>
<td>1</td>
<td>ALPHA</td>
<td>----</td>
<td>PURCHASE ORDER TYPE</td>
</tr>
<tr>
<td>77</td>
<td>PURCHSD</td>
<td>2</td>
<td>NUMERIC</td>
<td>INTEGER</td>
<td>PURCHASE ORDER SEQUENCE NUMBER</td>
</tr>
<tr>
<td>78</td>
<td>SDNO</td>
<td>7</td>
<td>ALPHA</td>
<td>----</td>
<td>PROJECT (S/O) NUMBER</td>
</tr>
<tr>
<td>79</td>
<td>GLENR</td>
<td>2</td>
<td>NUMERIC</td>
<td>INTEGER</td>
<td>GENERAL LEDGER NUMBER</td>
</tr>
<tr>
<td>80</td>
<td>VENDR</td>
<td>5</td>
<td>ALPHA</td>
<td>----</td>
<td>VENDOR NUMBER</td>
</tr>
<tr>
<td>81</td>
<td>DESCRN</td>
<td>40</td>
<td>ALPHA</td>
<td>----</td>
<td>PURCHASE ORDER DESCRIPTION</td>
</tr>
<tr>
<td>82</td>
<td>IDCNO</td>
<td>8</td>
<td>NUMERIC</td>
<td>DOUBLE</td>
<td>PRICE NUMBER</td>
</tr>
<tr>
<td>83</td>
<td>CHGORDR</td>
<td>1</td>
<td>NUMERIC</td>
<td>INTEGER</td>
<td>CHANGE ORDER NUMBER</td>
</tr>
<tr>
<td>84</td>
<td>PCHAMT</td>
<td>8</td>
<td>NUMERIC</td>
<td>DOUBLE</td>
<td>PURCHASE ORDER AMOUNT</td>
</tr>
<tr>
<td>85</td>
<td>INVCNT</td>
<td>8</td>
<td>NUMERIC</td>
<td>DOUBLE</td>
<td>INVOICE SALES IN DATE</td>
</tr>
<tr>
<td>86</td>
<td>INVACD</td>
<td>8</td>
<td>NUMERIC</td>
<td>DOUBLE</td>
<td>INVOICE ACCOUNTS PAYABLE TO DATE</td>
</tr>
<tr>
<td>87</td>
<td>INVRTR</td>
<td>8</td>
<td>NUMERIC</td>
<td>DOUBLE</td>
<td>INVOICE RETURN TO DATE</td>
</tr>
<tr>
<td>88</td>
<td>INVPOIN</td>
<td>2</td>
<td>NUMERIC</td>
<td>INTEGER</td>
<td>INVOICE POINTER (1 TO FIRST INVOICE)</td>
</tr>
<tr>
<td>89</td>
<td>FUTURE</td>
<td>2</td>
<td>NUMERIC</td>
<td>INTEGER</td>
<td>For future use</td>
</tr>
<tr>
<td>90</td>
<td>DESCRTN</td>
<td>2</td>
<td>NUMERIC</td>
<td>INTEGER</td>
<td>DESCRIPTION POINTER TO FIRST CODE RECORD</td>
</tr>
<tr>
<td>91</td>
<td>DELMTN</td>
<td>1</td>
<td>NUMERIC</td>
<td>INTEGER</td>
<td>DELIVERY/START DATE MONTH</td>
</tr>
<tr>
<td>92</td>
<td>DELDAY</td>
<td>2</td>
<td>NUMERIC</td>
<td>INTEGER</td>
<td>DELIVERY/START DATE DAY</td>
</tr>
<tr>
<td>93</td>
<td>DELYEAR</td>
<td>2</td>
<td>NUMERIC</td>
<td>INTEGER</td>
<td>DELIVERY/START DATE YEAR</td>
</tr>
<tr>
<td>94</td>
<td>FUTURE</td>
<td>2</td>
<td>NUMERIC</td>
<td>INTEGER</td>
<td>For future use</td>
</tr>
</tbody>
</table>

### FILE NO 2

**PURCHASE ORDER HEADER FILE**  
"HEADR.DAT"  
**LENGTH=4**

<table>
<thead>
<tr>
<th>FIELD</th>
<th>NAME</th>
<th>SIZE</th>
<th>DATA</th>
<th>DATA</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>95</td>
<td>96</td>
<td>2</td>
<td>NUMERIC</td>
<td>INTEGER</td>
<td>DIVISION NUMBER</td>
</tr>
<tr>
<td>97</td>
<td>PURCHOT</td>
<td>2</td>
<td>ALPHA</td>
<td>----</td>
<td>PURCHASE ORDER ORDER TYPE</td>
</tr>
</tbody>
</table>
the main menu. The appropriate program flowcharts for the main menu and the various subroutines are contained in the Appendix Part B. The following briefly outlines the logical design of each major subsystem.

Choice 1. "INITIALIZE A PC FILE" The purpose of the subroutine is to initialize all six program file structures. Upon entering this routine, the user is immediately asked if he really desires to initialize all the files (as this would destroy any data previously stored thereon). If the user enters an affirmative, these records are initialized by setting the appropriate pointer fields and sales fields to zero. The Purchase Order Master file has an indicator field showing whether that record is currently holding any valid purchase order data. All these indicators are set to the "unused status".

<table>
<thead>
<tr>
<th>File #</th>
<th>File Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Purchase Order Master file</td>
</tr>
<tr>
<td>2</td>
<td>Purchase Order Header file</td>
</tr>
<tr>
<td>3</td>
<td>Invoice file</td>
</tr>
<tr>
<td>4</td>
<td>Invoice Header file</td>
</tr>
<tr>
<td>5</td>
<td>Cost Code file</td>
</tr>
<tr>
<td>6</td>
<td>Cost Code Header file</td>
</tr>
</tbody>
</table>

The header file for each of the main data files contain system information on the number of currently used records in each main data file. Due to the overly restrictive nature of a single 160,000 byte disk drive, and for the purpose of this project, this initialization routine sets up 100 Purchase Order records, 300 Invoice records, and 300 Cost Code records. Historical data indicates a 3 to 1 ratio between these files is sufficient to meet almost any anticipated requirements.
As part of this initialization process, the user designates the diskette for the particular division, the specific purchase order type, and a specific Purchase Order block. This data is stored in the Purchase Order header file. Upon entering any of the eleven main menu subroutines, the system questions the user as to the division purchase order type, and the purchase order sequence block which he is attempting to process against. Only if this information agrees with that stored in the PO Header file is the user allowed to proceed. Otherwise the user is notified he is attempting to process against the wrong purchase order series diskette.

Choice I. "ENTER A NEW PURCHASE ORDER" This routine not only creates the Purchase Order Master file, but also creates the cost code files associated with that purchase order master file. Upon entering the routine, the user is prompted for the purchase order he desires to enter. The system then:

1. Checks to ascertain if the appropriate diskette is in place in the drive unit. If not, an error message occurs.
2. Prompts the user to enter the desired purchase order data, including job number, and general ledger number, delivery date, purchase order description, etc.
3. If there are multiple cost codes involved in the purchase order, the system prompts the user to enter "######". If there is only one cost code involved in the purchase order, that code is entered.
4. If there is only one cost code, a cost code record is automatically created and linked to the purchase order master record, using the same data in the Purchase Order Master record.
5. If there is to be more than 1 cost code, the program enters a cost code entry subroutine, prompting the user to build each cost code record. The system automatically logically links these

-14-
cost code records to each other and to the Purchase Order Master file. Control then returns to the purchase order entry routine.

4. The user enters the total purchase order amount. This must be equal to the sum of all cost codes, or the cost code entry routine and purchase order amount will be re-entered.

5. Return to Main Menu.

Choice 3. "DISPLAY A PURCHASE ORDER". After entering this routine from main menu, and entering the desired PO number, the system checks for the correct diskette, then displays the current purchase order record. This will contain the summary information of Purchase Order amount to date, (including all change order addition or deductions), sales to date, retainage to date and accounts payable to date. The ability to directly access Purchase Order records provides immediate response to these inquiries.

Choice 4. "EDIT A PURCHASE ORDER FILE". The three editing routines are all selective; that is, they do not allow complete freedom in editing. Each edit routine is designed to prevent the user from creating erroneous data, or creating discrepancies between the summary data in the Purchase Order Master file and its associated invoice and cost code chains. Each edit routine begins with a set of instructions, informing the user what he can edit and how to edit it correctly.

The user is then presented a menu listing the edit functions available in this subroutine.

Exhibit V is a printout of the Purchase Order Edit instructions as they would appear to the user on his CRT screen.

Note the system does not allow the user to change the
EXHIBIT V

PURCHASE ORDER EDIT INSTRUCTIONS


   The above areas must not vary within individual cost codes.

   When a purchase order is changed, these items in the cost codes are required by subsidiary to this purchase order by using the cost code edit routine.

   The PURCHASE ORDER AMOUNT SHOULD ONLY BE MODIFIED THROUGH THE COST CODE EDIT ROUTINE.

   All sales, returns, and accounts payable data in the PC master file can only be changed via the invoice edit routine.

   Invoice data is never modified, but only balanced with correction invoices.

2. USER IDENTIFICATION YOU DESIRE TO EDIT

   PC ID: [Enter PC ID Number]
   PC SN: [Enter PC Serial Number]
   PC DUE: [Enter Due Date]
   PC DIVISION: [Enter Division Name]
   PC LOCATION: [Enter Location]
   PC OPERATOR: [Enter Operator]
   PC DATE: [Enter Date]
sales, retainage, or accounts payable fields except through the invoice edit subroutine, and the purchase order amount can not be edited except through the cost code edit subroutine. This forces the user to edit these items by traversing the invoice or cost code chain. Only after properly inserting and linking the edit information within these two subfiles, is the correct summary information moved into the Purchase Order Master file. The Purchase Order edit routine then "walks" the user through the edit procedure with appropriate system prompts. If the Purchase Order job number is changed, the system automatically modifies the job number in every associated cost code record. If the actual purchase order number is modified, then a new purchase order record is set up, the old purchase order is logically "cleared" , the cost code pointer chain is transferred to the new purchase order record, and the purchase order numbers within each associated cost code is modified to the purchase order number. Note: The logical invoice chain is not transferred to the new purchase order record, but the chain is left attached to the old 'voided' purchase order record. The user is instructed if there are any current invoices, that there must be credit and debit correction invoices added through the ENTER AN INVOICE subroutine. This method is consistent with good accounting practices.

Choice 5. "DISPLAY COST CODES BY PO NUMBER" This routine displays on the CRT screen all the cost codes associated with the purchase order number submitted by the user. This routine simply follows the cost code chain and displays each cost code in the order of input. An example of
this output is contained in the user manual.

Choice 6 "ENTER ADDITIONAL COST CODES OR MODIFY(EDIT) PREVIOUS COST CODES". This routine performs two similar but separate functions. The first adds more cost codes after the original cost code(s) are set up in Choice 2. This function will allow the system to accept change order increases or deductions in the purchase order amount. These additional cost codes are physically added in input order at the tail end of the logical cost code chain. The sum of all additional cost codes are automatically added to the purchase order amount in the Purchase Order Master file when the user is finished inputting all additional cost codes. These additional cost codes will be separate records, even if they are identical to cost codes previously utilized. Previous cost code records are not changed.

The second function in this subroutine will actually modify previously created cost code records within certain limits inherent in the program central philosophy. The purchase order amount can not be changed through this edit (additional negative or positive cost codes must be added to effectively change a single cost code totals). The user can not change sales information concerning invoice payments through either of these two cost code edit functions, rather he must enter the invoice edit function to change any payment data. Eventually, the user is limited to editing description fields. Exhibit VI shows the CRT instructions given to the user upon entering this subroutine.
If existing cost code records are desired to be changed, then use this routine to change cost code number, cost code description, cost code amount, or cost code general ledger no.

If cost code number is incorrect, user should also change
the cost record until the master file using the PC EDIT routine.

If sales data in a cost code record needs to be changed,
then use function return to main menu, and use the invoice
function.

If invoice order no. is incorrect, edit and use the PC
EDIT routine.

If adding a change order reduction, it is preferred to add a
new cost code with negative amounts. Both the original cost
and new cost reduction cost code will print on the reports.

On exit existing data in cost code files.

Return to adding a cost code subroutine.
Exit cost code edit routines and return to main menu.
Choice 7 "ENTER A NEW INVOICE". This subroutine stores a new invoice in the next empty record and then logically links this new invoice to all previous invoices against that purchase order. The sales amount is input by each cost code within that purchase order, and the sum total of all cost code sales must be equal to the sum of the total invoice retainage and accounts payable amounts, else the user is forced to re-input. Credit invoices (such as to clear retainage accounts) are permissible. Payments to-date summary data is also transferred automatically to the Purchase Order Master file.

Choice 8 "DISPLAY AN INVOICE BY PURCHASE ORDER". This subroutine, upon entry of a valid purchase order number, will display chronologically all invoices against that purchase order number by following the invoice pointer chain, and sequentially displaying each invoice record. This display routine is extremely useful during the processing of new invoices. An example of this display is given in the user manual.

Choice 9 "EDIT INVOICE DATA". This edit subroutine is the most straightforward. Only the invoice number and the invoice date (both descriptions) can actually be modified in any existing invoice record. All numerical changes must be made through correction invoices, which debit or credit the appropriate general ledger codes within a job number. These correction invoices are logically added to the end of the previously existing invoice chain, and the sum of these correction entries is added to the sales data in the Purchase Order Master file to ensure
correspondance between the two files. This correction
entry method does not change data which may have been
previously entered into the corporate accounting system,
and hence maintains general accounting control standards.
The user is again given detailed CRT instructions upon
entering this subroutine as to what he can and cannot edit,
and how to properly do so.

Choice 10 "RUN VARIOUS REPORTS" Upon entering this sub-
routine the user is immediately presented with another
menu listing the various report options available to him.
(See Exhibit VII for a printout of the "RUN REPORTS" main
menu). The brief description of the various reports and
their program logic follows.

1. PRINT THE PURCHASE ORDER MASTER LOG This report
option prints the entire Purchase Order Master Log in
numerical order by passing sequentially through the
Purchase Order Master file. Since the Purchase Order
Master file is always in numerical sequence order, each
purchase order record is printed as it is accessed. This
report would be very useful to the field purchasing team
for both control purposes and for editing and processing
further system data in other subroutines.

2. PRINT THE COST ENGINEERING MASTER LOG This report
becomes a primary tool of the field cost engineer. It
lists all job charged cost code records after sorting
them according to this criteria:

   a.) By job number in sequential order
   b.) By cost codes (in ascending order) within
       each job number
   c.) By accounting general ledger number within
       each cost code
Each cost code record prints the cost code number, cost code description, cost code amount, sales to date against that cost code record, and most importantly, the amount of unpaid commitments. This last vital number is available from no other source than this system. For an example of this report, please see the Test Results section in Section F of the appendix.

The sort mechanism does not involve the physical creation of a sorted cost code file. Instead, due to the limited disk space, the logical sorting is done entirely within main memory. Four arrays are set up in main memory, each with a length equal to the number of occupied cost code records currently within the cost code file. The cost code file is sequentially read and the following data stored in the arrays. The first array contains the cost code's file position, the second array contains the cost code job number, the third array contains the cost code number, the fourth array contains the general ledger number. The four arrays are then sorted within main memory according to the above criteria, using a "bubble sort". The cost code records are then accessed in the same sequence as found in the sorted array set. Subtotals are provided for each cost code number and job number. An example of this report is found in Section F of the Appendix.

1. LIST ALL INVOICES. This report prints all invoices records currently entered into the system, in chronological order of entry. It is mainly an editing tool, although it could be used as a summary report for forward-
ing invoice applications to the accounting department.

An example of this report is given in Section F of the Appendix.

4. LIST ALL OVERHEAD PURCHASES  The first part of this report would list all purchases made against the Inventory General Ledger account number 1459. Accounting coding procedures dictate that these charges are not charged against a project number, so all inventory purchases will be grouped under a "blank" pseudo-job number "0000-00". For these entries the cost code field is used to contain the inventory code. These records are all part of the cost code file, and the sorting and printing logic is almost identical to that for the cost engineering report, with the exception of the sorting criteria.

now being:

a) By job number (all "blank" coded)
b) By general ledger number within job number "blank"

c) By inventory code within the general ledger number.

The second part of this report prints all those cost code records charged to overhead (i.e. to "blank" pseudo-job) and not charged to the specific general ledger inventory number 1457. The printing logic is similar to the above, with an additional sort sequence not needed, as the same sort will serve both report functions.

Choice 1: END APPLICATION  This will exit the user from the system program after closing all the files.

V. B. PHYSICAL DESIGN  This system was designed and runs using the following equipment:

1) An IBM Personal Computer Model 5150 with:
   a. one 160,000 byte disk drive
   b. 128,000 bytes of main memory
2) An IBM E151 Monochrome Display terminal
3) An Epson MX-100 printer.
This unit is easily transportable to remote jobsite locations. All the data files used in this system are random files with direct access based upon inquiry by key. Of the six files, three header files contain only one record, whose fields contain status information on the data file associated with it. Of the three data files (the purchase order master file, the invoice master file and the cost code file) only the invoice file is typically addressed by key inquiry. Access to the other two files is generally done logically by following pointers imbedded in the Purchase Order Master file. A physical summary of each file is given in Appendix C.

V. C. USER ENVIRONMENT. The physical user environment is anticipated to be field construction office, often a trailer. The user will typically either be a purchasing clerk/agent with limited formal training in the purchasing function or in computer systems. The purchasing accounts payable functions on such a large project would generally follow these guidelines:

1. All purchase orders chargeable to that function will be issued through the project manager at the job site.

2. The project will receive three "blocks" of Purchase orders in a known numerical sequence.

   1 block of type "A" purchase orders Major Material
   4 blocks of type "B" purchase orders Field
   1 block of type "C" purchase orders Subcontract

   The project will be responsible for all purchase orders issued to it.

3. All vendor invoices against purchase orders issued to a job site will be verified, approved, and coded (both accounting general ledger code and project management cost code) at the jobsite before being forwarded to corporate accounting for payment.
While the Project Management team will receive a number of cost reports processed through the Corporate main offices, there are no reports which will give the project management team information on committed but unpaid costs. Also the project purchasing function is under continual daily pressure to process massive amounts of purchasing information. They also need to be able to recall on summarized this data quickly upon demand. Part VIII Section 6 contains a Field Users' Manual for this system.
VI. SUMMARY AND CONCLUSIONS.

A. DESCRIPTION OF PROJECT RESULTS. The work done under this project has resulted in the following:

1. The development of standard Purchase Order Contracts of these types:

   Type A. Major Material Purchase Order Contract
   Type B. Field Purchase Order Contract
   Type C. Subcontract Purchase Order Contract.

   (See Section E of the Appendix for photocopies of the purchase orders developed under this project.)

2. The development of a stand alone, microcomputer based, Purchase Order Control System for field use on large construction projects. The menu-driven system routinely builds and maintains data in three areas:

   Purchase Order Master file
   Cost Code file
   Invoice (Accounts Payable) file

This information can be input into the system directly from Purchase Orders, Vendor Invoices, and Change Orders to a Purchase Order Contract. The system is always available for maintenance or updating of these files, and the user prompt mechanism forces good system procedure during such maintenance or updating.

This system, when used routinely, will significantly increase the workload capacity of the field purchasing function, while significantly reducing the rate of error through its automatic editing function during data entry.

3. The system routinely outputs these reports:

   a) Purchase Order Master Report:
      (all purchase orders by PO type, in numerical sequence)

   b) A Project Purchasing Commitment Report
      (all purchase orders by purchase order type, sorted by job, within job sorted by cost code, and within cost code sorted by general ledger number)
c: An Inventory Commitment Report
(all purchase orders written to various inventory accounts; by Inventory account, and within inventory account by inventory code)

d: An Overhead Commitment Report
(all purchase orders charged to overhead accounts, sorted by general ledger code).

An example of each of these is contained in Section F
"Test Results" of the Appendix.

1. OBJECTIVE EVALUATION OF THE PROJECT RESULTS: This project was conceived to fill a specific, narrow, niche in the existing operations of a specific major construction company. It was not intended to be a massive, generic application program, but to output a specific set of documents under given conditions, documents which would contain information not available from any other source. The system would output these results while enhancing user purchasing throughput and decreasing user errors. The system described in the project indeed meets these specifications. Given the physical limitations of diskette storage size, I am well satisfied with the system's performance. It must be noted here that due to the data limitation inherent in the single sided drive used for the system development, this system was developed and tested using a maximum file size of 100 records for the Purchase Order Master file, 300 records for the Invoice file and 500 records for the Cost Code file. A single-density diskette will hold up to 250 Purchase Order records and their associated invoice and cost code records. To be truly effective, the system should have a minimum 500 Purchase Order records and 1000 to 1500
invoice and cost code records (depending upon the Purchase order type). Ideally blocks of 1000 Purchase Orders would be preferred. Calculations show a single double sided, double density disk drive of 640,000 bytes would hold Purchase Order blocks of 1000 and all the associated invoice and cost code records. Two double sided disk drives would work just as well, (probably better, because it will be easier to backup the data files). In short, before this system is actively used in the field, a second disk drive should be purchased and installed in the system.

The system has been made as user friendly as possible, with copious comments and notes in the interactive mode. Most users, if basically familiar with the corporate processing procedures, could be comfortable with the system after a couple of hours of "hands-on" experience without any extensive outside coaching.

Another objective of the system was to force the user into maintaining good purchasing and accounting procedures. This the system does extremely well, especially in editing modes, only allowing editing with certain restrictions and under a set of guidelines imposed by the system. These system checks preventing poor input procedures could be more extensive with future additional programming.

The report routine presently generated fulfills the requirements of the project proposal, however, there
are several very useful reports that could be generated if the information data fields were expanded to include such items as actual delivery dates, invoice dates, and others.

The actual program coding used in this project is primarily subroutine orientated. As new subsystems were added, it became obvious there were many coding sequences which were endemic to a variety of the system operations. In some instances the author rewrote program sections to make them stand alone subroutines which could be easily used throughout the program. This often leads to subroutine tiers three or four deep, and can be potentially confusing to any future programmers, so in many cases identical coding blocks were repeated for the author’s own programming clarity. There is little doubt, however, that the program length could be shortened further by “subroutining” wherever possible. The present block arrangements of the codes is as developed during the course of project programming. It is somewhat restrictive, and the program should be renumbered and the subroutine blocks rearranged for additional clarity.

One very important contribution of the project is the conceptual ground it lays for the development of a corporate system, based on the same philosophy and guidelines as this small field system. In that case, the Invoice file would truly be the Accounting Accounts Payable file. This “pilot” project could save significant future programming and development costs if the Goldston
Corporation ever decided to implement a similar system in its' mainframe system. In such a future development, this project system might still serve a valid field function when telecommunicatively linked to the mainframe unit, allowing faster processing, direct inquiry, and reduction in input data error while not tying up the corporate mainframe.

In summary, this system, when applied with an upgrade in the hardware disk capacity, will perform as desired, and has the potential to pay for itself (both new hardware and the software development costs) on a single major construction project within the Goldston organization.

3. RECOMMENDATIONS FOR FURTHER SYSTEM DEVELOPMENT.

1. Expand the hardware storage capacity to allow "blocks" of 1000 purchase order records and the associated cost code and invoice records on file at one time. A hard disk could be used to maintain an even larger data base.

2. Renumber and rearrange the actual coding blocks to improve clarity and enhance efficiency.

3. If the sort routine on the cost code file used for the printing of the Project Commitment Report becomes too slow, another pointer field could be added to the cost code record, which would be used to logically link the cost code records together in report order as they were input.
4. Of significant value would be to expand the number of files to eight, through the addition of a Change Order Master file and a Change Order Header file. These files would be nearly identical to the Purchase Order Master file, but would be used to summarize the series of new cost code added under a Change Order label (as it would actually occur in the field). This would clarify even further these code additions or deductions by grouping the same under the appropriate change order.

5. With the development of a sophisticated screen entry routine, the line prompt system could be dropped in lieu of a "fill in the blanks" and screen representation of a purchase order. With a high impact printer, there is no reason why Purchase Order forms could not be printed through the computer if the forms were purchased with tearaway sprocket field edges.

6. The internal system checks on the data entry routine could be expanded even further to include automatic checks on the validity of the inputed general ledger number against the purchase order division code and the Purchase Order type. (This check function would need an override, as these coding rules are not always observed.) Other checks could prohibit entry of an invoice which would overpay either a cost code or the total purchase order, and give the appropriate system message to the user.
BIBLIOGRAPHY

1. IBM "BASIC" Software Reference Manual
2. IBM "DDS" Software Reference Manual
3. IBM "GUIDE TO OPERATIONS" Software Reference Manual