To the Graduate Council:

I am submitting herewith a thesis written by David Cummings Herrell II entitled "The Addition of an Integral Feedback Loop to an Existing Ratiometric Process Controller." I recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Electrical Engineering.

Robert E. Bodenheimer, Major Professor

We have read this thesis
and recommend its acceptance:

Donald W. Bouldin

Accepted for the Council:

Vice Chancellor
Graduate studies and Research

Thesis
79
.H477
cop.2

THE ADDITION OF AN INTEGRAL FEEDBACK LOOP TO AN EXISTING RATIOMETRIC PROCESS CONTROLLER

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

David Cummings Herrell II

June 1979

1389790

ABSTRACT

The purpose of this study was to evaluate the desirability of integrating an additional feedback loop into an existing ratiometric process controller to further reduce the errors inherent in the system design. Several design constraints were imposed on the proposed hardware design, including the consideration of maintainability, testability, and physical constraints of the existing system.

One of the primary constraints imposed was the completely designed, unmodifiable nature of the controller. By developing hardware for an unmodified controller, the upgraded capabilities of the system could be utilized in processors installed and operational prior to the new design. As well as basing the design on previously existing equipment, the integration of two systems of incompatible design must be performed into an inexpensive, but highly reliable system.

Since the mathematics of the system must also be modified, ideally in the minimal amount possible, the internal stored program must be changed. This modification should be consistent with the previously existing versions of the control algorithm, and must be compatible with all

other versions. Conclusions generated by the research indicate that the proposed modifications are possible and economically feasible.

System design will allow addition of the interface between the separate pieces of machinery without drastic modification of the individual components. Increased system accuracy will pay for the increased capital expenditure, with an increased savings in milk fat inventory. Any increase in system reliability will justify the purchase of the hardware improvement by the lessened probability of selling a product which does not meet the standards imposed by the Federal government on labeling of content.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

STATEMENT OF PROBLEM

During the span in which a product has market appeal, certain improvements in that product become necessary to remain competitive. Frequently, such development will enhance the product by the integration of another device into the system. For the project of this research, the primary product is a special-purpose, dedicated process controller. This product will be enhanced by the addition of an integral control feedback loop to the system to minimize the effect of several inherent process errors.

In order to describe the system, a brief historical review of the first methods used will prove helpful. One of the best reviews of this system was performed by Seiberling in his 1974 article [1]. This article also served to contrast modern system design to the earlier designs.

Early methods of homogenization and pasteurization of fat-controlled milk products utilized a batching environment to control formulation of the products, as shown in Figure 1. Raw milk was drawn from a series of large tanks and separated into skim milk and cream. These two entities were placed in separate tanks until a determination of the fat content of each tank was performed. Once these tests were complete, the personnel on the operating floor made the

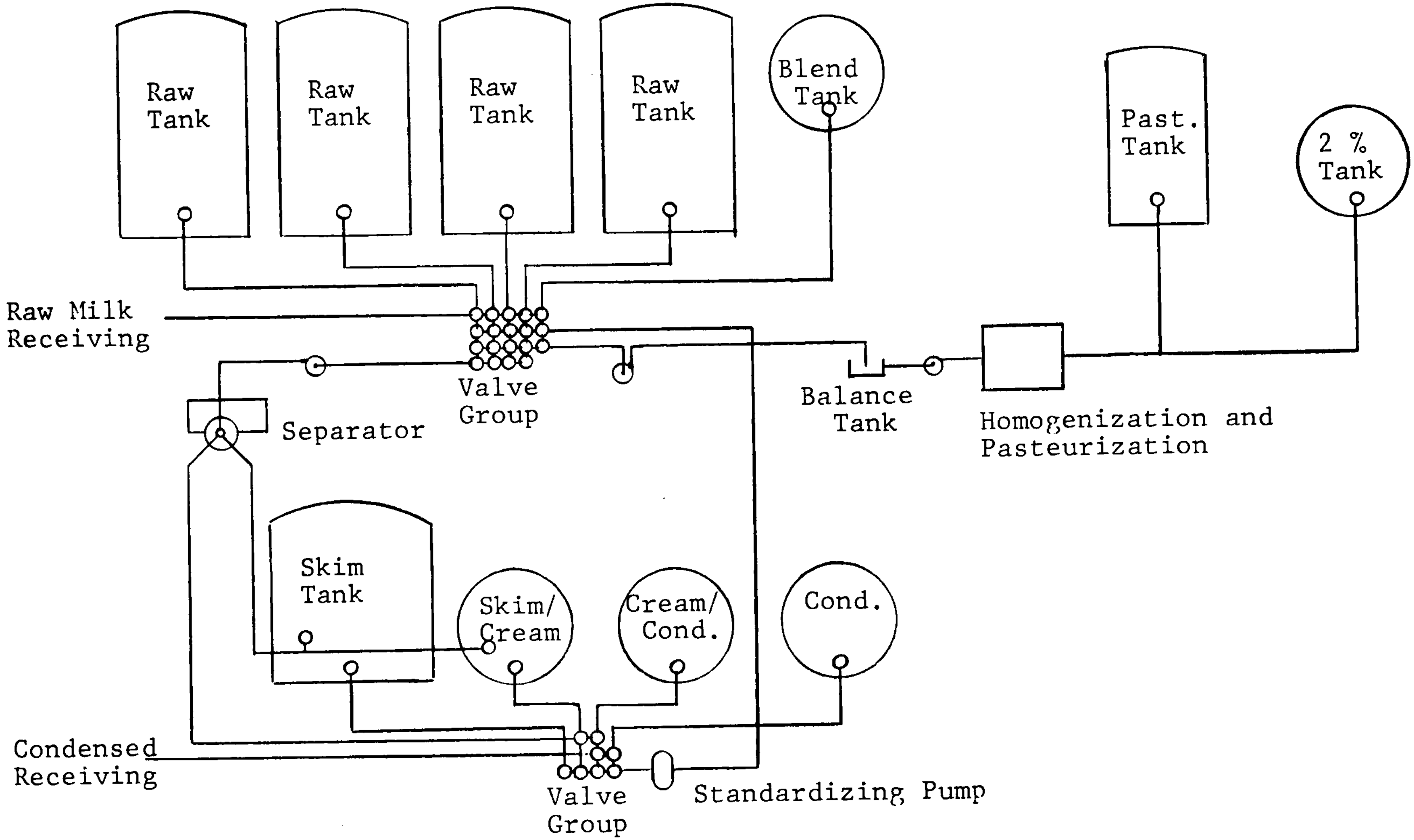

Figure 1. Milk Processing Plant Schematic

calculations necessary to create a blended, fat-controlled product. Then, roughly controlled amounts of each constituent were pumped into a blending tank.

After the resulting mixture was agitated for some set time, the percentage of milk fat was determined by accepted chemical tests. The required adjustment was then made by the addition of either low or high fat fluid. After adjustments to chemical formulation were successful, the resulting mixture was homogenized, pasteurized, and pumped into a final tank for packaging.

This process has several undesirable qualities which were described by such authorities as Elliott, [2]. The editorial referenced attempted to define desirable capabilities and functions which an integral in-line system for milk standardization should possess. Many schemes were devised in an attempt to increase the accuracy of constituent metering. Obviously, the more accurate the measurement, the less adjustment required in the finished product. Several systems have been developed which accurately measure precomputed ratios of constituents in a blending environment. Unfortunately, since the fat test of the constituents can not be accurately determined before the separation process is performed, and since the raw material must be used with as little delay as possible, the personnel performing the control must perform several complex calculations in at least three unknowns in order to set up the process correctly.

The batching process also involves the use of several large, expensive refrigerated tanks to hold intermediate stage products. Some means of agitation must also be provided in these tanks to speed the mixing process during the adjustment of the formulation of the product. The elimination of the need for this equipment should result in a significant reduction in capital investment.

Also, as the size of milk processing plants increased, a batching system required many chemical tests for fat and milk solids content in raw and finished product tanks for each step of the process. As the testing load on the laboratory staff increased, the repeatability and accuracy of the tests decreased radically. As this occurred, the number of adjustments required increased, at tremendous cost.

During this period, investigation in the field proved that raw milk could be separated more effectively at elevated temperatures. It was recognized that performing separation as a part of the homogenization and pasteurization process would cause a significant increase in plant productivity.

In an attempt to decrease capital equipment expense in a fluid milk plant, increase the productivity of the plant, and decrease the time spent moving milk between tanks, the design of an in-line system for computing the required control ratios and controlling the processing of the raw milk directly into the finished product tanks was undertaken. With this design, the addition of a means of carefully controlled blending of milk solids, which were not fat, into the product was considered. While not necessary for consumer acceptance or required by then current government regulation, the addition of milk solids to certain products increases nutrition and palatability.

At this point, the development of a computer-based process control system was initiated. Computer Concepts Corporation introduced the 1202 Digital Blending and Separating System, which will be referred to as the 1202, as the first practical control system to be widely accepted in the fluid milk industry. This system reduced the problems inherent in the manual system to a level at which product quality control and productivity were both maintained at acceptable levels in plants of any size.

The 1202 possessed several failure modes which allowed production of product of incorrect formulation. If a significant degradation occurred in a turbine meter being used to meter the flow rate in the piping, the calibration factor used by the 1202 to convert the stream of pulses from that meter into gallons was incorrect. This caused errors in the formulation of the finished product.

Also, when an operator failed to enter a number exactly into the front panel, a totally incorrect product was produced. In addition, when the human operator made an error while performing any chemical assay on the raw tank, the product was not correctly formulated.

In order to reduce the probability of occurrence of error in formulation of the finished product, the integration of a device capable of correcting the errors inherent in the system was undertaken. The original means of correction of error, as shown in the upper portion of Figure 2, utilized a Milko-Tester to determine the fat content of the finished product. This system depended on manual timing and information feedback to correct the inevitable errors in formulation. No automatic testing capability existed and operation tended to be labor-intensive. As such, the application of the system did not yield consistently corrected finished product due to erratic or non-existent feedback.

To increase the reliability of the system, Computer Concepts Corporation provided the capability of automating the testing process with their version of a process control system, shown in the center portion of Figure 2. This

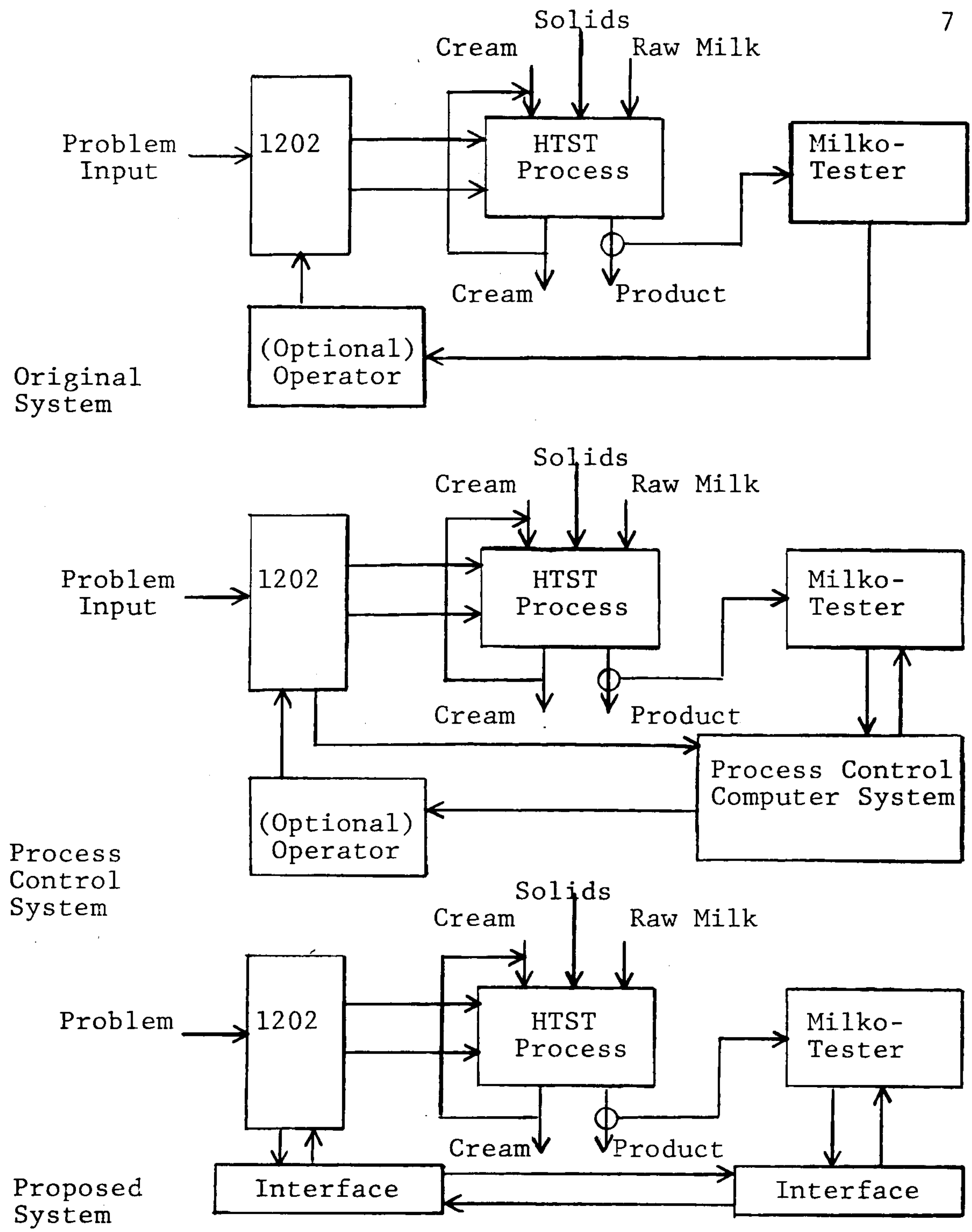

Figure 2. Information Flow Diagrams

system automatically sampled the fat content of the finished product, compared the result to the required parameter set on the 1202, and generated an audible alarm on significant deviation between the two values. Again, a human operator was required to change the values set into the 1202 to produce the required correction in formulation. With the addition of increased capital investment in a very complex process control system, note that this system still required operator intervention to achieve the correction required for proper formulation of products.

Attempting to remove the restrictions imposed by the presence of a human operator, this research proposed the complete automation of information flow within the system, as shown in the lower portion of Figure 2. Information flow occurs at automatically generated, periodically timed intervals from the Milko-Tester to the 1202. Updated information regarding the current fat content at the outlet of the High-Temperature, Short-Time process will be utilized by the modified program within the 1202 to correct the formulation error without resorting to human intervention.

The next chapters are concerned with a description of the system integration process. These chapters cover the architecture of the 1202, the control mathematics used in flow calculations, the hardware design necessary to integrate the system, a synopsis of the operation of the unified system, a description of the simulation system designed to verify correctness of the modifications made to the stored program within the 1202, and a series of conclusions and possible improvements for the system.

CHAPTER 2

MATHEMATICAL BASIS OF OPERATION

The 1202 process control system was designed with the intent that the processor would be dedicated to process control in a fluid milk processing system. Since the process is controlled on the basis of ratios of flow rates within the system, and since these ratios are generated from the solution of either a three or four dimensional set of matrix equations, the architecture included substantial numeric manipulation capability. The requirements for branching and other program flow control are almost nonexistent. These requirements did not lead to the architecture of a conventional general-purpose computer, as is evident in the instruction set, found in Table 1.

Several major variations from the von Neumann architecture were introduced during the design process, as shown in the block diagram of Figure 3. Since the machine operates from an unmodifiable stored program, the instruction space was completely separated from the data space. This nine bit wide instruction space is implemented in programmable read-only memory, on a board containing the program counter and associated program flow control.

The data space was then separated into three distinct areas. Modifiable memory consists of four storage loca-

Table 1. System 1200 Instruction Set

Instruction Format

Bit Field	Function
8 to 5	Operation Code (Op code)
4 to 0	Operand

Instruction Set

Mnemonic	Op Code	Function
TN	0000	A and MQ = Thumbwheel(Operand)
SQ	0001	Memory(Operand) = MQ
SAC	0010	Memory(Operand) = A; A = 0
SAA	0111	Memory(Operand) = A
TI	1111	A and MQ = Inventory(Operand)
A1	0110	A and MQ = A + Thumbwheel(Operand)
S1	0011	A and MQ = A - Thumbwheel(Operand)
M1	1000	A and MQ = MQ * Thumbwheel(Operand)
D1	1110	A and MQ = MQ / Thumbwheel(Operand)
A2	0101	A and MQ = A + Memory(Operand)
S2	0100	A and MQ = A - Memory(Operand)
M2	1001	A and MQ = MQ * Memory(Operand)
D2	1010	A and MQ = MQ / Memory(Operand)
R	1011	Display A in Readout(Operand)
RB	1101	Blank Readout(Operand)
RR	1100	Display A in Readout(Operand); Repeat program

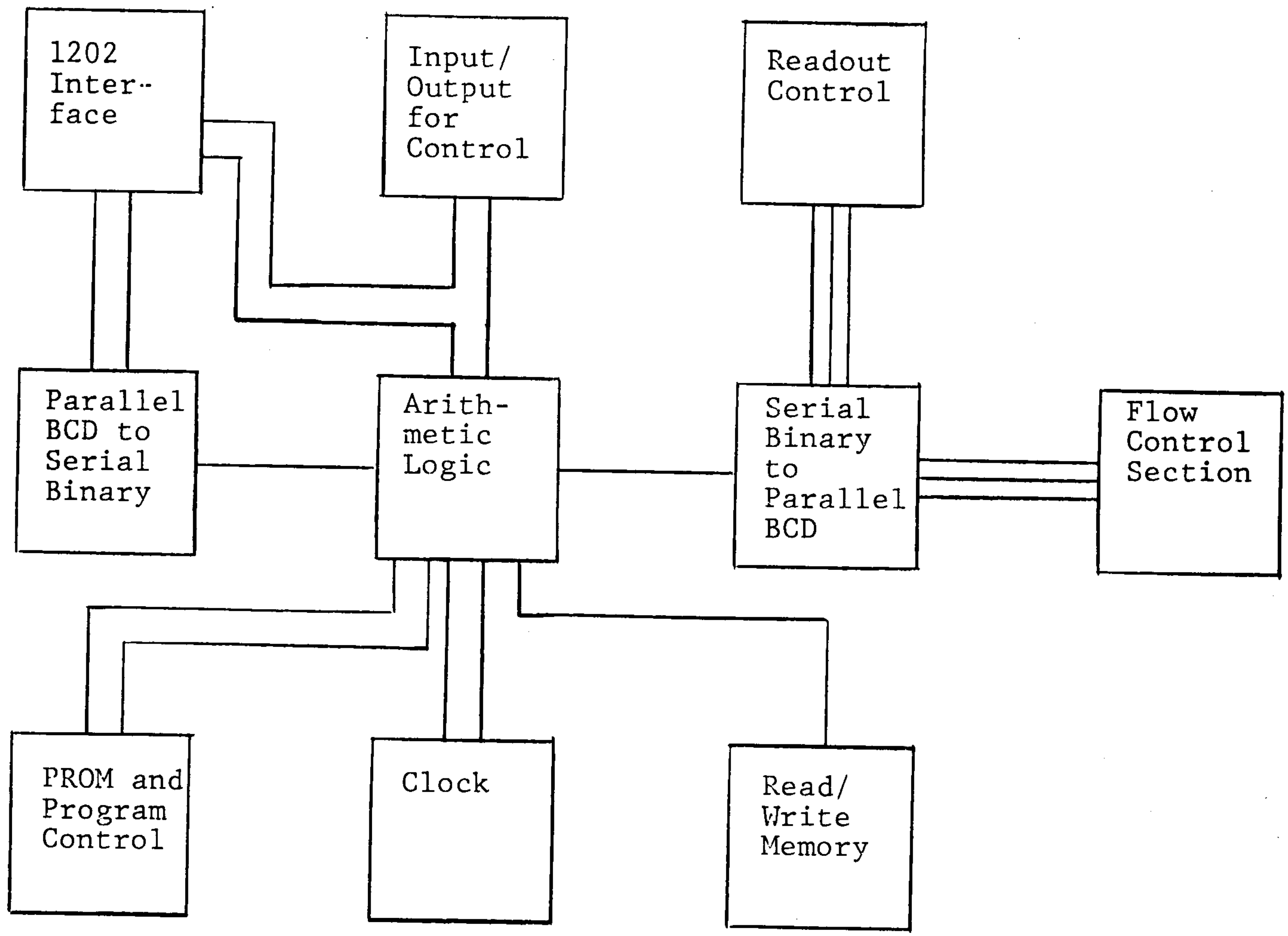

Figure 3. System Level Components of the 1202 Series Processor

tions, all of which are implemented as registers. The remaining data spaces are primarily associated with the man-machine interface. Read-only data, which is binary coded decimal (BCD), is obtained from thumbwheel switches set by the operator. Write-only data, which is placed on display for operator convenience and partially routed to the Flow Control Section for control purposes, is totally separate.

The arithmetic requirements require the processor to utilize 56 bits of BCD data to express all numeric information. In order to economically operate on a data word of this size, the arithmetic unit was implemented as a bit-serial processor, operating on binary data. The necessary conversion from BCD to binary is performed during the conversion between serial and parallel data flow.

In order to free the processor from the time-dependent operation on flow control data, a series of ratiometric flow control boards were designed. These boards have the capability of controlling a throttling valve directly from the two streams of meter pulses which represent two flows and the control ratio stored by the arithmetic unit. The collection of boards which perform this function is called the Flow Control Section of the processor, which operates totally independent of the program flow in the numeric calculations.

Many of the common architectural practices used during the period in which the processor was designed were incorporated in the design of the system. The arithmetic processor utilized an accumulator, A, and a multiplier quotient, MQ, for numeric manipulation. Due to the operation of the unit, the contents of the two are normally identical after any numeric operation. All data transfers to any point in the machine are routed through the accumulator.

The four temporary storage registers, C(0) through C(3), are available for storage of intermediate steps in the calculations. One additional register, C(6), is a write-only control location. That register is used to set the number of gallons of product to be produced for the problem input from the front panel. When the required number of gallons of product have been metered, the system halts further generation of any processed product. This state, called Recycle, will force changing the hardware to a configuration in which the present contents of the piping is continuously cycled through a closed loop in the system.

Four inventory registers, I(0) through I(3), exist as read-only locations. These are loaded from the Flow Control Section, and contain the number of gallons of the product, raw milk, milk solids which are not fat, and cream used or removed during the production of the current problem input from the front panel. These locations are reset when the system changes state from a Forward Flow condition to a Recycle state.

Data is entered into the machine by several banks of thumbwheel switches located on the front of the console. Each logical group of thumbwheels represents some specific parameter, defined in Table 2, and is assigned a unique hardware address. Up to 32 groups of thumbwheels are possible. The data defined by thumbwheels W(0) through W(2) consist of a single, mutually exclusive bit which serves to define the type of problem input by a rotary switch on the front panel.

Output data is displayed to the operator through four banks of seven segment readouts, R(0) through R(3). Each readout utilizes several back-lit captions to define the units of the data being displayed. Since the current program has the capability of displaying data which is grouped in eight logical sets, multiplexing of the four displays is performed for the 32 unique numeric results. Data to be displayed in the readouts is selected by means of seven user-selectable pushbutton inputs on the front panel. With the use of a pair of pushbuttons, the remaining data group may be selected. Table 3 lists the data grouping as currently defined.

Data displayed in the group selected as "Control Ratios" is utilized by the Flow Control Section as a basis for control. Precise interpretation of the ratios depends on

Table 2. Thumbwheel Functions

Octal Address	Mnemonic	Function
00	TENTH	Constant 10,000.
01	SS	Per cent solids in slurry
02	SP	Per cent solids to be added
03	FM	Per cent fat of milk
04	FC	Per cent fat, separated cream
05	FAC	Per cent fat, added cream
06	ONEHUN	Constant 100.
07	LPGAC	Lbs./Gal., added cream
10	LP	Lbs. of product
11	FP	Per cent fat of product
12	FS	Per cent fat of slurry
13	LPGC	Lbs./Gal., separated cream
14	LPGP	Lbs./Gal., product
15	LPGS	Lbs./Gal., slurry
16	MILKOD	Milko-Tester fat reading
17	MILKOC	Milko-Tester control
20	W0	Separator, cream added
21	W1	No separator, cream added
22	W2	Separator, no cream added
23	FK	Per cent fat, separator skim
24	LPGMIL	Lbs./Gal., raw milk
25	ONETH	Constant 1,000.

Table 3. Readout Format

Instruction Format

Bit Field	Function
8 to 5	Operation code (R, RR, or RB)
4 to 2	Operator selection mode
1 to 0	Readout select (R(0) to R(3))

Operator Mode Selection

Select Code	Data Pertains to Group
0	Inventory, in gallons
1	Product
2	Standard
3	Raw Milk
4	Slurry
5	Added cream
6	Separated cream
7	Control ratios

the current state of the blending system mode rotary switch on the front panel of the 1202. Readout R(0) was was defined as the ratio of the rate of addition of milk solids which are not fat in nature to the system to the rate of product flow out of the system. Readout R(1) specifies either the ratio of the rate of cream added to the rate of total cream separated, or the ratio of cream removed from the system to the total rate of cream separated, depending on the definition of the problem. Readout R(2) indicates the ratio of the rate of cream out of the separator to the rate of raw milk added to the system. Readout R(3) specifies the state of the 60/40 bit, which will be defined as the model for separator operation is developed.

With operator input capabilities defined, development of a system model from which a mathematical model may begin. The schematic flow diagram of Figure 4 will be used for this development. This represents a schematic flow diagram of a continuous High-Temperature Short-Time (HTST) process for milk fat and solids control in fluid milk, as developed in reference 3. Functionally, controlled amounts of raw milk, milk solids, and cream are combined to produce a product and possible excess cream. The homogenization and pasteurization of the product is performed during the blending or separation process.

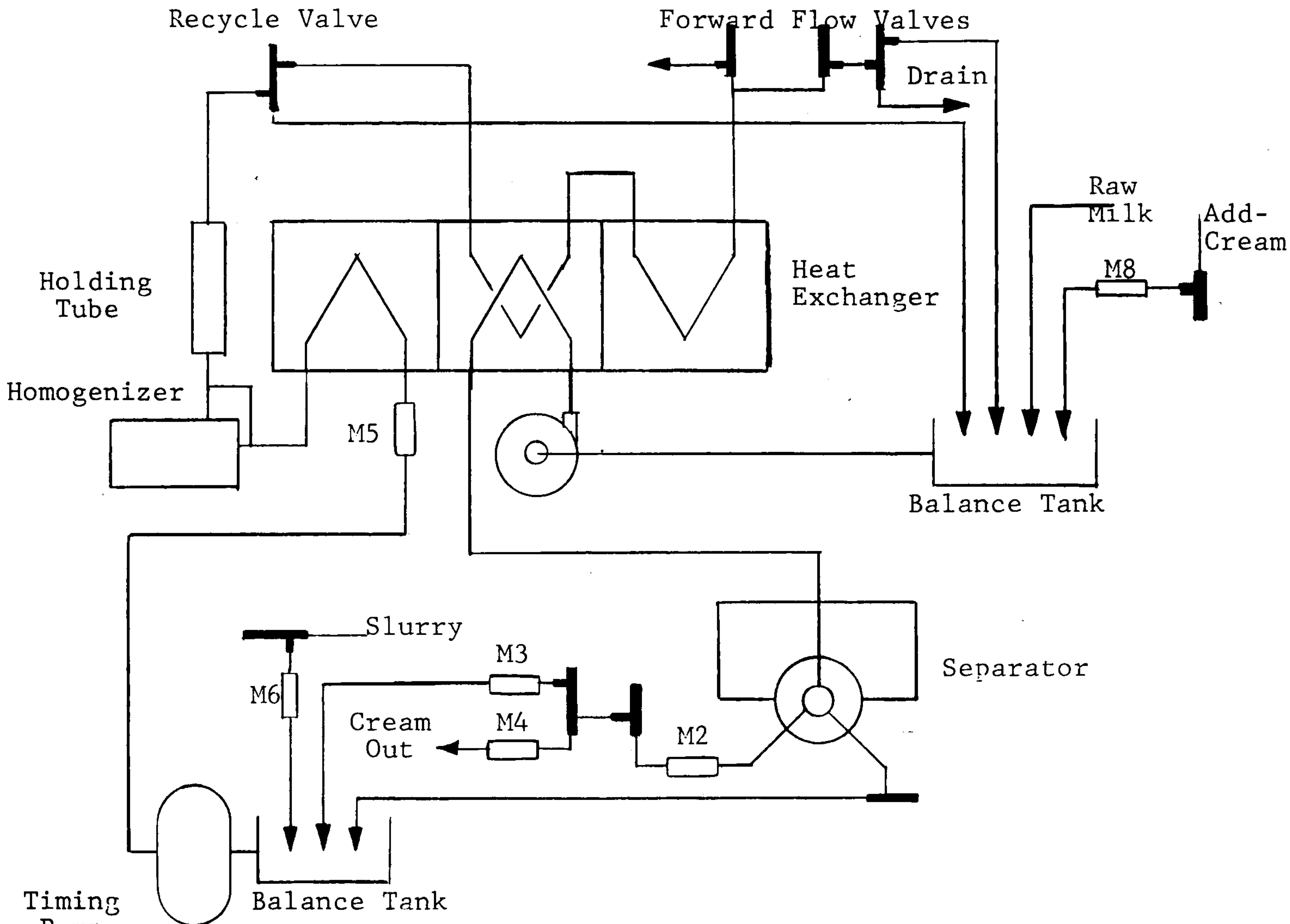

Figure 4. High-Temperature Short-Time Diagram for 1202 System

The basic process is similiar to the batching process, with the exception that no temporary storage is required. All corrections are performed as the milk flows through the system. Intermediate state storage and mixing tanks required in the batching process are no longer present. Multiple laboratory fat and solids tests for intermediate stage product determinations are no longer required. Operating costs for this system can thus be greatly reduced from the previous batching system.

During the examination of Figure 4, some explanation of terms should be performed. Slurry is a fluid milk product which has been fortified with dried milk solids which are not fat in nature. The slurry is used to increase the solids content of the milk to meet Federal regulations and increase consumer acceptance of certain products.

Add-cream is a milk liquid which is high in butterfat, and is produced during the separation of milk. The fat content of this fluid is normally in the 40 per cent milk-fat range, and will be produced when the final product has less fat content than the raw milk.

BT-1 and BT-2 serve as balance tanks. Both serve as a means of storing a small amount of fluid for surge protection during the process. The system must maintain both at a constant level. Until the system reaches steady-state conditions, the oscillation of fluid flow rates could introduce air into the fluid milk without the protection of the storage capacity of the balance tanks.

The symbols used on the figure are defined in Table 2. In addition to the mnemonics in that table, the following additional symbols are used to describe a model for the flow system. RS indicates the rate of flow of slurry into the system, as metered by M6. RP indicates the rate of flow of the product out of the system, as metered by M5. RC indicates the rate of flow of cream out of the separator. RCO indicates the rate of flow of cream out of the system, and RAC indicates the rate of flow of cream into the system.

Control of the system is based on one uncontrolled stream and ratios of flows to either the "wild" stream or to a fluid flow which has been referenced to that stream. Only the ratios of added or removed cream to the total flow of cream from the separator contribute to the correction of errors in fat test having small percentage values. During the initial phases of this development, it was arbitrarily decided that only those formulation errors of less than 10 per cent fat-test in the finished product need be considered for correction. From an examination of normal aging in turbine meters and other factors affecting the accuracy of the finished product, only small corrections are necessary for normal, non-catastrophic failure modes. This was numerically demonstrated in a letter written by the Economics Laboratories, Equipment Engineering Division in early 1973, [5].

After consideration of several methods of error correction, the design was based on a discrete time integration. This choice allowed the effective averaging of the results of several tests over an entire production run, thus minimizing the effect of possible erronious conversion of a single Milko-Tester sample. The control ratio for fat is initially set to a computed value. After the process is running, a correction is added or subtracted from the initial ratio correction. This value will be stored for later use at this point in the correction process. Each additional calculation, performed at discrete intervals, is averaged into the sum, thus preserving all previous corrections, while placing more emphasis on the most current test results.

The most mathematically elegant means of storing the modification factor was in terms of the percentage of the input fat test. This modification factor, M, could then be set, cleared, and modified under program control, based on the current machine state. This technique also served to minimize the changes to the previously existing code.

The upper portion of Figure 5 models the system when the fat-test of the finished product is to be increased, and considers only the boundary conditions at some steady state flow conditions. This model assumes that only slurry is used to add solids to the system. A complete development of the mathematics behind the process is developed in internal documentation written by Computer Concepts in 1972 [3]. Using electrical analogy and Kirchhoff's Current Law, the following set of matrix equations may be generated.

$$\begin{vmatrix} FM & FAC & FS \\ 0 & 0 & SS \\ 1 & 1 & 1 \end{vmatrix} * \begin{vmatrix} LPGMIL*RM \\ LPGAC*RAC \\ LPGS*RS \end{vmatrix} = \begin{vmatrix} FP \\ SP \\ 1 \end{vmatrix} *LPGP*RP$$

These equations generate the two control ratios, RAC/RP and RS/RP.

$$\frac{RAC}{RP} = \frac{LPGP*(FM*(SS-SP) + FS*SP - FP*SS)}{LPGAC*SS*(FM-FAC)}$$

$$\frac{RS}{RP} = \frac{LPGP*SP}{LPGS*SS}$$

These two equations are the basis for control of the system in Separator Out Mode. In this mode, a change in the control ratios RAC/RP is directly proportional to the resulting change in fat test of the product. Since an error percentage could be developed based on an actual fat test reading, a corrected ratio may be generated by multiplying

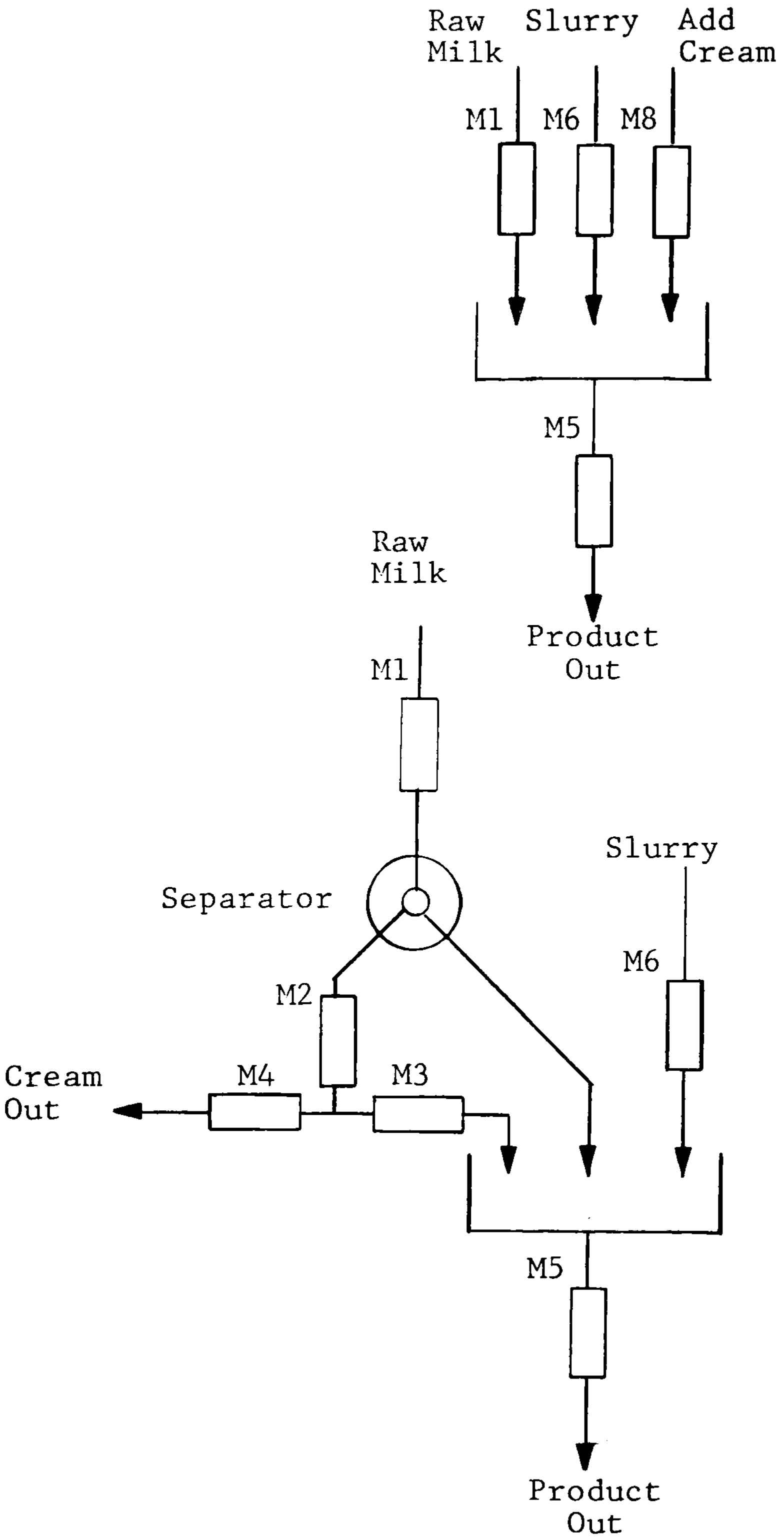

Figure 5. Schematic Flow Models

the percentage correction, M, adjusting the fixed point representation of the number, and using the modified ratio for control purposes.

The model for this process is simpler than the model required when the separator is utilized. In this problem, shown in schematic form in the lower portion of Figure 5, raw milk is separated into skim milk and cream. Some controlled amount of separated cream is added to the resulting skim. The remainder is stored for use in later products. As before, slurry may be added to improve the appearance, flavor, and nutritive value of the product. The slurry can make no contribution to the fat test of the finished product.

Modeling separator-based operation with an electrical analogy, the application of Kirchoff's Current Law yields the following set of matrix equations, with the addition of the assumptions required for flow continuity and other physical and operational assumptions.

$$\begin{vmatrix} (FC-FK) & 0 & -(FM-FK) & 0 \\ 0 & -(FC-FK) & (FM-FK) & FS \\ 0 & 0 & 0 & SS \\ 0 & -1 & 1 & 1 \end{vmatrix} * \begin{vmatrix} LPGP*RC \\ LPGC*RCO \\ LPGMIL*RM \\ LPGS*RS \end{vmatrix} = \begin{vmatrix} 0 \\ FP \\ SP \\ 1 \end{vmatrix} *LPGP*RP$$

This set of equations may then be solved for the three control ratios necessary in this physical problem.

$$\frac{RC}{RM} = \frac{(FM-FK)*LPGMIL}{(FC-FK)*LPGC}$$

$$\frac{RS}{RP} = \frac{SP*LPGP}{SS*LPGS}$$

$$\frac{RCO}{RC} = \frac{(FC-FK)*(FM*(SS-SP) + SP*FS - SS*FP)}{(FM-FK)*(FC*(SS-SP) + SP*FS - SS*SP)}$$

One additional bit of control information must be generated. Early in the development process for the original system, the necessity of basing control on the cream flow having the greater rate became apparent. Either M3 or M4 would be utilized for control, based on the projected flow through the two meters. A single bit control field, called either the 50/50 bit or the 60/40 bit depending on the relative age of the documentation, determines the meter to be used. In order to increase accuracy of measurement, the turbine meter having the greater flow rate is selected for control to force utilization of the meter having the more accurate, linear range. The 60/40 variant was introduced to force control change as a result of empirical knowledge of the controlled process. If at least 60 per cent of the total flow passes through M4, then the 60/40 bit will be set to one, M4 will be used for control, and the ratio will be unmodified. If this condition is not met, the bit will be set to zero, M3 will be used for control, and the ratio will be modified to the additive inverse of the present value (100.00 - RCO/RC) to increase the accuracy of the flow control.

In order to correct the control ratios for small errors, only the ratio controlling cream out of the system must be modified, since the only terms affecting the fat-test of the finished product are contained within this ratio.

Formulation of the modification factor must be changed to handle this physical phenomenon. The additive inverse of the modification percentage, M, was developed, maintaining the center of the new ratio at 100.00 per cent, to agree with the original. The equation utilized for the ratio correction is:

$$R' = RCO/RC * (200.00 - M)$$

This correction will suffice for the Standardize and Manual Separator In Modes. The Total Separation Mode, which forces the maximum removal of cream from the system, will not be modified. Since no fat test correction can be performed during this mode of operation, the sampling of fat test data was disabled. Results of the computations force the valve which would normally control the addition of fat back into the system to a completely closed state. This action will cause a product of minimum fat test to be generated.

CHAPTER 3

OPERATION AND DESIGN OF HARDWARE

In order to dynamically modify the fat test of the finished product, a mechanism to automatically sample the fat content of the fluid milk is required. One of the most widely used units developed for this task was the Milko-Tester, developed by A/S N. Foss of Denmark. This instrument is utilized by most of the dairies which have advanced to the 1202 control system. Also, complete data and device schematics were readily available at the time the design phase involving the machine was performed.

In the book written by Lampert [4], the system is described as operating by measuring the opacity, or scattering, of light by fat globules in a milk sample. A sample of specified volume is drawn into the machine, filtered, and warmed to approximately 60 degrees Centigrade. This sample is then mixed with a chelating reagent, which serves to prevent precipitation of various salts and to disperse the milk proteins.

The mixture is then homogenized in a small four-stage unit which develops a pressure of about two thousand pounds per square inch. This serves to roughly size the remaining fat globules. The liquid then passes through a carefully controlled beam of light. This beam is detected by a photometer, which has been calibrated in per cent fat. The data is then digitized and displayed on the front panel, as well as being placed on the unit's relay bus. The act of taking the next reading normally flushes any residue from the system, although samples approaching the extremes (0.00 per cent to 9.3 per cent fat) may produce erroneous readings.

The BCD data presented by the Milko-Tester can then be transferred from its remote location back to the 1202 processing unit, and used to correct the fat-test of the product. From physical constraints, the Milko-Tester and the 1202 processor may be separated by appreciable distances, which may introduce ground potential between the two units. To prevent problems, optical isolation should be provided.

A solution to these problems was determined to consist of an an interface for the Milko-Tester, an interface for the 1202, and a serial communication link between the two interfaces. To conserve power and space in the 1202 cabinet, any function which could be placed in either interface was designed into the Milko-Tester interface.

An internal power supply is required for the interface. Both power supply and interface may be mounted inside the Milko-Tester, utilizing the 220 volt a.c. power receptacle provided by the equipment manufacturer. A triple supply, capable of one ampere of five volt d.c. and 100 milliamperes each of +12 and -12 volt d.c., will be required for adequate power margin.

Referring to Figure 6, the interface to the Milko-Tester required conversion of an 25 volt d.c., active low relay bus to transistor-transistor logic (TTL) levels. At this point, optical isolation was provided to achieve some degree of noise rejection and freedom from ground noise with respect to the relay bus. The twelve data lines and the sample complete signal were converted.

All remaining communication with the Milko-Tester is performed using the start sample line. Power for this line was tapped from the Automatic Mode indicator line. This action allows the Milko-Tester sampling to occur only when the unit is in Automatic mode, thus yielding the maximum flexibility for the system.

Only three additional external control lines exist for this interface. Two of the three are data lines, used to transmit data and clock signals for the data back to the 1202 interface. These signals are transfered as EIA RS-232-C compatible levels. To avoid synchronization problems, the clock generated for the Universal Asynchronous Receiver / Transmitter, UART, at the Milko-Tester interface transmitter was also transmitted to the 1202 interface receiver.

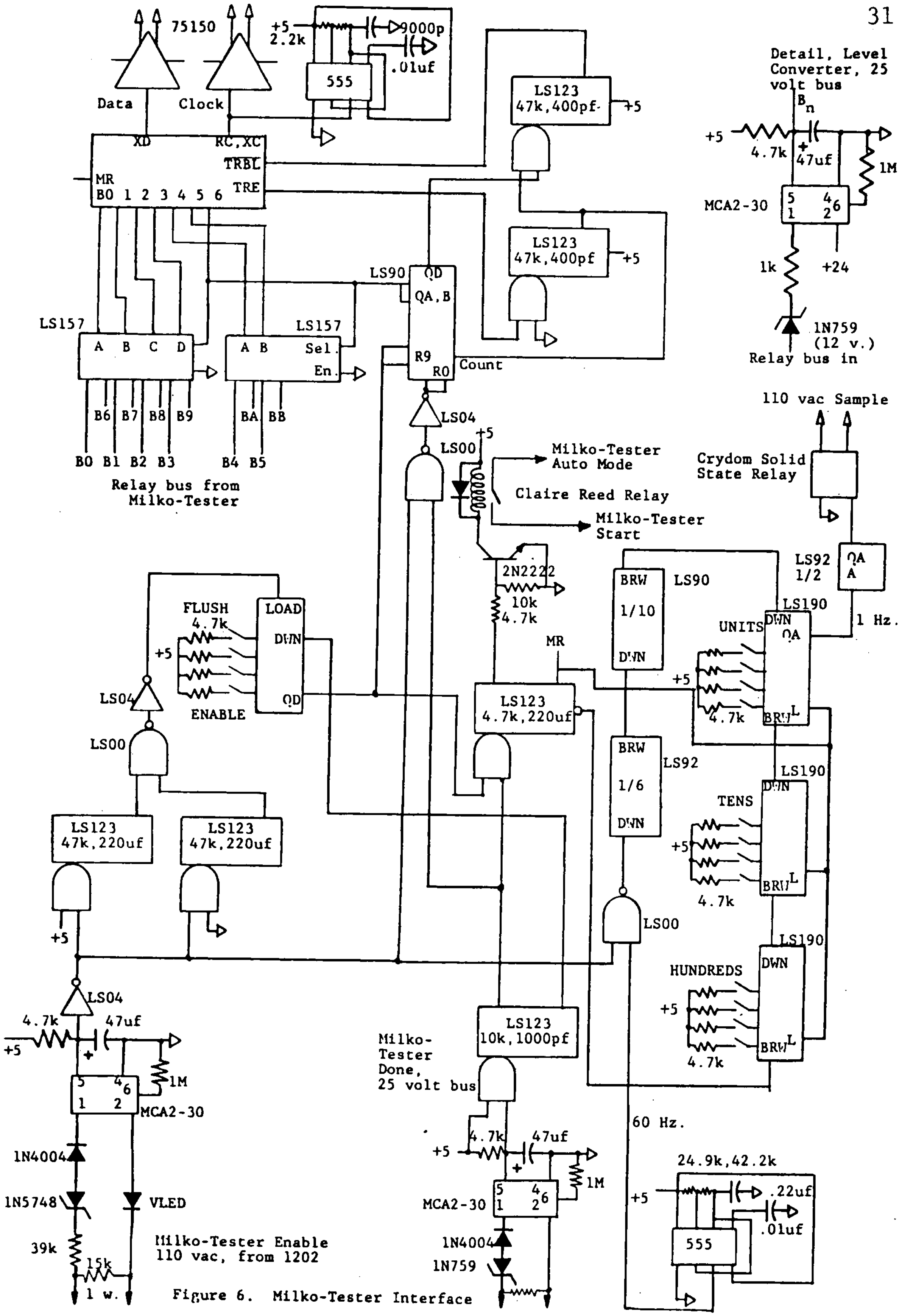

Figure 6. Milko-Tester Interface

The remaining control line into the Milko-Tester interface is an enable line. In order to maintain compatability with historical precedent in the fluid milk industry, the control line will be considered enabled when a 110 volt a.c. signal is present on that line. This control line will be considered disabled when no voltage is present on the line.

Since the test results may be invalid if the fat-test results change drastically from product run to product run, any change in state in the enable control line will also conditionally initiate a selectable number of Milko-Tester cycles, with no milk input to the machine. This will serve to flush any remaining traces of the last product from the system. During this operation, no fat test results will be transfered back to the 1202. This function is enabled by the switch labeled "Flush Cycle". The count of cycles required is determined by the remaining nibble of switches in that switch pack.

Also contained on this board is the parallel to serial conversion circuitry driver. The basis for the circuit is a UART package. When combined with the support circuitry, this device will be used to send the two six bit groups of data which represent the fat test of the fluid milk to the 1202 processor interface. This data will be transmitted a total of four times in case of possible interference. The

1202 interface will reject most transmission errors. Due to the low bit rate of transmission, the highly noise immune signal levels used, the use of parity in the transmission, and the repetition of the data, a high degree of reliability should be possible.

An additional function which the Milko-Tester interface performs is the timing of the period between samples. As distributed by Foss, the cycle time in automatic mode is slightly greater than 20 seconds. The cycle is generated by the completion of the last sample complete pulse. Assuming some reasonable process activity period of six hours each day for a six day week, the unit would cycle 6,480 times, or approximately 338,580 times in an average year. Such rapid sampling is neither desirable nor useful, since most High-Temperature Short-Time systems, as shown in Figure 4, page 19, have a much larger transport and settling time than 20 seconds.

Thus, an adjustable timing circuit is provided, with a 999 second range, to allow propagation of the changing conditions through the piping system to the Milko-Tester sampling port. Changes in the operating point of the system should not be made at a rate greater than that time required to sample, move the data back to the 1202 interface, process the data within the 1202, adjust and stabilize the valve positions, and have stable composition fluid flow at the sample port. Changes at a rate greater than the minimum will cause system oscillation, and possible system instability.

The only function remaining is derived from the timing chain. A two second period pulse train is generated when the system is enabled. This signal will be used to generate 110 volt a.c. bursts to control the operation of the sample port. Several electrically-actuated solenoid valves exist which require two second pulse trains to generate flow compatable with the sampling requirements of a Milko-Tester.

The interface for the 1202 is thus left three functional requirements. That interface will be responsible for determining when the Milko-Tester should be enabled, synchronized storage of the periodically updated fat-test, and informing the 1202 program of the new results.

The required power for the interface can be generated by the present 1202 power supplies. Adequate power margin is present to allow the use of an additional quarter ampere of five volt d.c. and 20 milliamperes of both +12 volt and -12 volt d.c. from existing supplies. The required 12 volt supplies are generated by zener regulation of the 15 volt d.c. supplies.

Mechanical mounting of the unit could be accomplished using the hardware developed for the System 1500 Process Control interface. This would place the interface assembly in the back of the processor cabinet. Ribbon cable and individual wiring would then be run to the several points required for interface to the system, as indicated in Figure 7, the schematic for the interface.

One of the functions of the 1202 interface is the determination of Milko-Tester enabling. This will be performed by examination of the state of the Forward Flow bit and the Total Separation Mode indicator bit. If the unit is in Forward Flow and not in Total Separation, the Milko-Tester will be enabled. If either condition is not met, the Milko-Tester will disabled.

The 110 volt a.c. solid-state relay output will be fed to the Milko-Tester. As a means of operator control of the system, a three position switch would be provided in this control path. One position, Auto, gives complete control of the system to the 1202 interface. The two remaining positions give manual control to the operator. Manual Off would totally stop sampling in the Milko-Tester. Manual On would run the Milko-Tester at the sample rate set by switches in the Milko-Tester interface. This would allow Milko-Tester sampling during periods when the 1202 would not be in use or when automatic sampling would be disabled.

During the period when automatic sampling is enabled, the 1202 interface will receive two six bit data items which

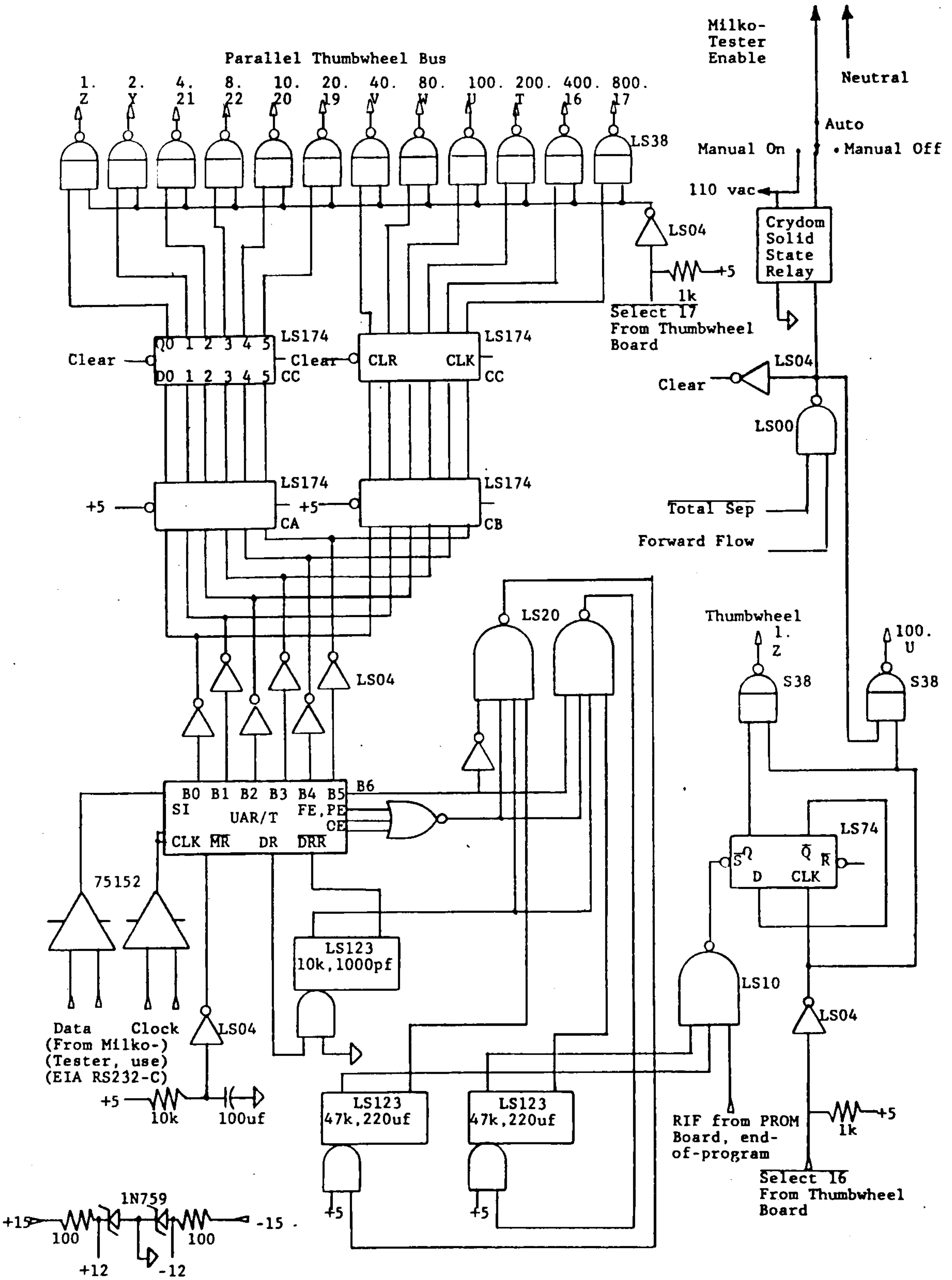

Figure 7. 1202 Interface Schematic

represent the BCD value of the fat test of the fluid milk. When each is received free of transmission errors, that part of the data will be gated into a buffer register. A monostable multivibrator will be used to indicate valid data present for each transmitted section. If both are received within an appropriate period and the end of program in the arithmetic unit is reached, the data will be gated into the output register. If one data item is never successfully transmitted, the successful item transfer flag will be cleared.

The act of transferring data into the output register will set the new data available flag, bit zero of thumbwheel 17. This will serve to indicate the presence of new data to the running program. This flag will be cleared after thumbwheel 17 is read. Consideration should be held for this factor in program design.

One additional bit is defined in thumbwheel 17. The numeric value 100 will be present when the Milko-Tester is enabled and milk sampling should be performed. This bit will be cleared when either of the previously mentioned conditions disable fat testing. This bit is generated by the same logic which generates the enable, and will not be affected by read operations on the thumbwheel.

Generation of the modification factor may now be completely defined. An explanation of the utilization of this quantity was given in Chapter 2. The generation of the factor should be considered in three separate stages. If the process is not in Forward Flow, then the fat test ratio should not be modified. In this case, M should be set to 100.00 per cent, which represent the unmodified setting of control ratios. Using an inverted Forward Flow bit, the quantitiy may be developed as shown.

$$M' = (1.00 - MILKOC/100) * 100.00$$

If the process is in Forward Flow, then the last M should be preserved. The initial constant, 100.00 per cent, should be disabled. Since MILKOC is now set, the initial set point will be disabled. To allow the retention of the old data, the last M should be added to the previous equation to yield an equation of the form shown.

$$M' = (1.00 - MILKOC/100) * 100.00 + M * (MILKOC/100)$$

When a new fat test reading is presented, the M data must be updated. This update should consist of a signed percentage error between the fat test, set in the front panel by the operator, and the actual fat test, generated by the Milko-Tester. This quantity should then be superimposed on the composite equation last shown to yield the final equation for the modification control percentage, M.

After simplification, the following equation is generated.

$$M' = (1.00 - MILKOC/100) * 100.00 + (MILKOC/100) * (M + 100.00 * (FP\text{-}MILKOD)/FP)$$

This equation is sufficient for complete specification of all modification necessary to implement the correction within the 1202.

CHAPTER 4

DESIGN AND VERIFICATION OF SIMULATOR

In order to determine the effects of slight modification of the control code on the behavior of the system, to allow simple, rapid changes in the control code, to check the modifications made for correctness in the minimum time, and to gain a complete working understanding of the processing unit, generation of a software simulation of the 1202 was undertaken. To allow rapid completion of the design and debug phases of the development process, the Fortran IV language was utilized on a DECsystem-10 time-sharing system. The development process served to divide the system into three major levels and four logical divisions of code.

To preserve the dynamic range possible when using the target machine, the use of floating point mathematics was required. This choice allowed the use of the complete 14 digit range of the 1202, but introduced a problem for small numbers. Integer arithmetic defines a truncation or rounding process at the minimum possible value of fixed-point scaling used. Floating point notation retains as much significance as possible. In order to reconcile this dicotomy, the subroutine TRUN was developed to round floating point numbers symmetrically, in a simulation of the operation of the 1202 processor.

To insure intelligible communication of data between subroutines in a logical manner, a Blank Common was developed which completely defines the current processor state. To further simplify the task of indexing through thumbwheels, temporary storage, and inventories, each of these was defined as a vector. Since standard Fortran does not allow the use of zero as a valid index, the octal index used by the 1202 required translation. This translation was performed by using the octal value directly as decimal, and adding the value one to the decimal representation. These defined constants were then set into a set of descriptive labeled constants to ease the problems of coding and usage.

Read-outs were simulated, using the same indexing techniques, in an eight by four array. The column index is used to store the data for each of the four displays which would be visible on the front panel at any given time. The row index was used to store the data for each of the eight display modes. The interpretation of the eight mode indices is listed in Table 3, page 17.

The bottom level of the simulation, listed in the Appendix as TN, must consist of a group of subroutines which simulate the operation of the individual machine instructions. Each of these subroutines will perform checks on the input calling parameters. If any errors are found, a message defining the errror is output. After validity checks are completed, each subroutine will manipulate the machine state and return for further processing of 1202 instructions.

As a means of decreasing the overhead associated with the simulator, two deviations from complete simulation of the target processing system were allowed. The most important of these deviations is in the TN instruction. The TN instruction is the only instruction with the capability of resetting the new data available flag in the Milko-Tester Control thumbwheel. All instructions capable of using thumbwheel data should be capable of resetting this flag. By knowing in advance that only this instruction would be used to input the required data, this deviation could be tolerated.

The other deviation required the use of a Return statement at the end of the simulated program. Rather than generate some means of using the RR instruction to generate the required control, such control must be added by the programmer. This deviation also implies that only one pass through the simulator will be performed for each call to the simulation subroutine.

Two code sections exist at the next level of the simulation. The initialization of the thumbwheel data forms a small subroutine, called INIT in the Appendix. The other is the 1202 program simulation, which uses subroutine calls to implement the actual program flow.

In order to provide easy modification of the input data from thumbwheel switches, initialization of this data was separated into a small subroutine, INIT. This localized operator input to a single program area, and simplifies the task of finding the data for purpose of verificiation or modificiation.

Fixed-point scaling will be employed for any input representing either a percentage or a volume to weight conversion, using a factor of 100. Thus, the per cent solids in slurry, SS, represents 1.00 per cent in this routine, and not the 100 which is apparently set. This fixed-point scaling must be considered at all times within the program and must be corrected at the point in the calculations when the least error will be introduced in the final results.

The other subroutine at this level, listed as PROGRA in the Appendix, is the simulation of the actual 1202 program. Two copies of the subroutine exist in the Appendix. Each is identified in the opening comments as being either the original or the heavily modified, newer version.

The modification was primarily concerned with the addition of the control modification. As a part of coding practice, the duplicated code was removed, existing code was simplified where possible, and the use of C(3) was removed.

No precise definition of the extensive changes necessary was listed. Since an original and the modified version are present, the changes may be found by careful comparison of the two.

The entry point and the highest level of code for the simulator is listed as MAIN in the Appendix. This code serves to initialize the state of the machine for input and output. Then, each of the three modes is exercised in the order Standardize, Separator Out, and Total Separation.

For each, the problem is run and the results of that run are output, as shown in Tables 4, 5, and 6. The tabulated results must agree with the data published for the given problem in the Maintenance Manual for the 1202 (Computer Concepts, 1973), reference 6. Careful comparison of the results generated against the results published in that manual indicate the correct function of the simulator.

At this point, no more calculations prove necessary if Total Separation is invoked. This would mark the end of the problem run, since no correction can be performed for fat content in this control mode.

If the mode is either Standardize, summarized in Table 4, or Separator Out, summarized in Table 5, the operation of the blending system program changes would be exercised. An indication was given that the Forward Flow state in the system existed. A set of calculations for the set mode

Table 4. Standardize Mode Simulation

	Product	Std	Milk	Slurry A	Cream S	Cream	Ratio
	Startup						
R0	66667.	100750.	50375.	4000.	0.	3071.	600.
R1	5000.	10000.	10075.	300.	0.	5375.	5335.
R2	1000.	10750.	-1.	1000.	-1.	-1.	1143.
R3	0.	10000.	-1.	-1.	-1.	-1.	1.
	Forward Flow, No Milko Data						
R0	66667.	100750.	50375.	4000.	0.	3071.	600.
R1	5000.	10000.	10075.	300.	0.	5375.	5335.
R2	1000.	10750.	-1.	1000.	-1.	-1.	1143.
R3	0.	10000.	-1.	-1.	-1.	-1.	1.
	One Pass, 0.10% High						
R0	66667.	100750.	50375.	4000.	0.	3071.	600.
R1	5000.	10000.	10075.	300.	0.	5375.	5228.
R2	1000.	10750.	-1.	1000.	-1.	-1.	1143.
R3	0.	9800.	-1.	-1.	-1.	-1.	-1.
	Second Pass, No New Data						
R0	66667.	100750.	50375.	4000.	0.	3071.	600.
R1	5000.	10000.	10075.	300.	0.	5375.	5228.
R2	1000.	10750.	-1.	1000.	-1.	-1.	1143.
R3	0.	9800.	-1.	-1.	-1.	-1.	-1.
	Third Pass, 0.10% High						
R0	66667.	100750.	50375.	4000.	0.	3071.	600.
R1	5000.	10000.	10075.	300.	0.	5375.	5122.
R2	1000.	10750.	-1.	1000.	-1.	-1.	1143.
R3	0.	9600.	-1.	-1.	-1.	-1.	-1.
	Startup						
R0	66667.	100750.	50375.	4000.	0.	3071.	600.
R1	5000.	10000.	10075.	300.	0.	5375.	5335.
R2	1000.	10750.	-1.	1000.	-1.	-1.	1143.
R3	0.	10000.	-1.	-1.	-1.	-1.	-1.
	Milko Data, 0.10% Low						
R0	66667.	100750.	50375.	4000.	0.	3071.	600.
R1	5000.	10000.	10075.	300.	0.	5375.	5442.
R2	1000.	10750.	-1.	1000.	-1.	-1.	1143.
R3	0.	10200.	-1.	-1.	-1.	-1.	-1.

Table 5. Separator Out Mode Simulation

	Product	Std	Milk	Slurry	A Cream	S Cream	Ratio
	Startup						
R0	66667.	42222.	21111.	4000.	15926.	0.	600.
R1	5000.	10000.	4222.	300.	478.	0.	2389.
R2	1000.	47778.	-1.	1000.	-1.	-1.	1143.
R3	490.	10000.	-1.	-1.	-1.	-1.	0.
	Forward Flow, No Milko Data						
R0	66667.	42222.	21111.	4000.	15926.	0.	600.
R1	5000.	10000.	4222.	300.	478.	0.	2389.
R2	1000.	47778.	-1.	1000.	-1.	-1.	1143.
R3	490.	10000.	-1.	-1.	-1.	-1.	0.
	One Pass, 0.10% High						
R0	66667.	42222.	21111.	4000.	15926.	0.	600.
R1	5000.	10000.	4222.	300.	478.	0.	2437.
R2	1000.	47778.	-1.	1000.	-1.	-1.	1143.
R3	490.	9800.	-1.	-1.	-1.	-1.	0.
	Second Pass, No New Data						
R0	66667.	42222.	21111.	4000.	15926.	0.	600.
R1	5000.	10000.	4222.	300.	478.	0.	2437.
R2	1000.	47778.	-1.	1000.	-1.	-1.	1143.
R3	490.	9800.	-1.	-1.	-1.	-1.	0.
	Third Pass, 0.10% High						
R0	66667.	42222.	21111.	4000.	15926.	0.	600.
R1	5000.	10000.	4222.	300.	478.	0.	2485.
R2	1000.	47778.	-1.	1000.	-1.	-1.	1143.
R3	490.	9600.	-1.	-1.	-1.	-1.	0.
	Startup						
R0	66667.	42222.	21111.	4000.	15926.	0.	600.
R1	5000.	10000.	4222.	300.	478.	0.	2389.
R2	1000.	47778.	-1.	1000.	-1.	-1.	1143.
R3	490.	10000.	-1.	-1.	-1.	-1.	0.
	Milko Data, 0.10% Low						
R0	66667.	42222.	21111.	4000.	15926.	0.	600.
R1	5000.	10000.	4222.	300.	478.	0.	2341.
R2	1000.	47778.	-1.	1000.	-1.	-1.	1143.
R3	490.	10200.	-1.	-1.	-1.	-1.	0.

Table 6. Total Separation Mode Simulation

	Product	Std	Milk	Slurry	A Cream	S Cream	Ratio
	Startup						
R0	66667.	112500.	56250.	4000.	0.	6429.	600.
R1	5000.	10000.	11250.	300.	0.	11250.	0.
R2	1000.	22500.	-1.	1000.	-1.	-1.	1143.
R3	490.	10000.	-1.	-1.	-1.	-1.	0.

would be performed. These calculations would then be identified and output. Examination of the output showed no change in control ratio for the change in control state.

Then, the next step will feed a 5.10 per cent product test into the system with a 5.00 per cent product expected. A set of calculations for the set mode would be performed. These calculations would then be identified and output. Examination of the output showed the required change in control ratios, as defined in Chapter 2, for the cream control ratios.

Now a program computational pass will be performed with perturbed product fat test, but no new data. Examination of the output of the calculations showed the control ratios and modification ratios to be unchanged, as required. The fat test of the product is apparently wrong, but no update of the control ratio in R(7,1) was performed since no new fat test data exists.

Again, a test step will define a new product fat test of 5.10 per cent into the system, with a 5.00 per cent fat test expected. Results of the calculations performed showed the expected change in value of control code and modification value, based on an identical error in the results returned from the last two fat tests. As more tests are accumulated, the change in control ratio will continue to reflect the perturbations induced by all test results. This will allow long term averaging of test results in the action of the feedback loop.

Entry into the next pass of the 1202 program models the system leaving Forward Flow state into a Recycle state. The results of the calculations showed the correct restoration of the control ratios in preparation for the next run. This would allow the system to produce a totally new product, unaffected by the state of the last product runs.

The final step of the verification process will define a product fat test of 4.90 per cent, with a 5.00 per cent fat test expected. A set of calculations for the set mode would be performed. Examination of the results of the calculations showed correct changes, as defined in Chapter 2, on the affected control ratios.

Thus, all possible state transitions in the modification of the program were tested and proven correct. The program modificiations and additions are thus shown to be effective in the proposed system integration. With the developed system, the properties of the product are being used to generate an integral error signal to be used in conjunction with the data set in the problem in the front panel to correct the control ratios in the required amounts to minimize the error signal. This minimization process will serve to minimize the difference in formulation between the product produced and the product which was to have been manufactured.

CHAPTER 5

CONCLUSIONS

The purpose of this design was to integrate two previously existing process control machines into a unified system. Processing capabilities of both machines would be extended by this addition. The desired goal of this design process was to decrease the operating cost of a fluid milk processing plant and increase the reliability of the formulation of the products.

Integration of the two systems has been shown to be possible. All initial goals were achieved in this system. Results of the timed conversion of fat test were fed back into the ratiometric process controller, and the operating point of the system was changed based on the changing results. Averaging of the data over a complete run was implemented in the discrete integration performed by the software algorithm.

An external means was provided for operator control in an overriding mode. The system designed is totally standalone, and requires no additional computer support. Changes and modifications in control ratios are performed totally automatically.

Possible extensions and additions to the system should now be considered. The inclusion of an alarm of some type to warn of corrections exceeding the limited range desirable in the system should be considered. Also, some limit on the maximum excursion possible for the correction factor might be worthwhile.

Another valid extension would be averaging the Milko-Tester readings over some short-term period, as well as the long-term average currently performed. This would have several samples taken in the minimum time frame, and averaged. This should also include the rejection of samples outside some tight bounds around the mean of the short-term collection of samples. This would allow the rejection of invalid data from the Milko-Tester. Currently, the design will use any data being returned from the fat testing device. This could lead to serious errors, in the event of a Milko-Tester failure.

Another valid change would be to adjust the fat test of the raw milk entering the system based on the Milko-Tester reading generated from the raw milk. This was utilized in at least one fluid milk processing plant, using the System 1500 Process Control System. When the system was designed, this alternative was not considered. Several later references suggest the desirability of control based on the raw milk data. This should be studied, possibly in an operating environment.

These conclusions and possible extensions lead to the possiblity that additional work in the area would be necessary to determine the most economical, effective approach for maximization of the benefit of the system, while minimizing the cost of implementation.

BIBLIOGRAPHY

1. Sieberling, Dale A., "Computer Controlled Digital Separating / Standardising / Blending of Fluid Milk Products." Dairy Industries, London, England, January/February, 1974.

2. Elliott, Robert A., Editorial, American Dairy Review, November, 1977.

3. Computer Concepts Corporation, Internal Design Documentation, 1973.

4, Lampert, Lincoln M., Modern Dairy Products, Chemical Publishing Company, New York, 1975, pp. 394-395.

5. Economics Laboratory, Equipment Engineering Division (ELEED), Correspondence between Dale A. Sieberling and Ross Duncan, July 10, 1973.

6. Computer Concepts Corporation, 1202 Digital Separating and Blending System Maintenance Manual, 1972.

APPENDIX

```
C DAVID C. HERRELL       REVISION: 12-APR-79
C
C       ENTRY POINT FOR SIMULATOR.
C
C       MODULES REQUIRED:
C       MAIN               ENTRY POINT AND DRIVER
C       INIT               INITIALIZATION OF INPUT
C       PROGA              ACTUAL 1202 PROGRAM
C       IN                 SIMULATION OF 1202 INSTRUCTIONS
C
C       RESULTS WRITTEN TO FORTRAN LOGICAL UNIT NUMBER 21
C
C       SET UP COMMON BLOCK OF MACHINE STATE
C
        REAL TW,C,I,A,MQ,RO
        COMMON TW(28),C(7),I(4),A,MQ,RO(8,4)
C
C       SET UP POWER-ON FOR MACHINE, ZERO ALL MEMORY,
C       ZERO ALL INVENTORIES, SET READOUTS TO INVALID
C
        DO 11 K=1,7
11      C(K)=0.
        DO 12 K=1,4
        I(K)=0.
        DO 13 J=1,8
13      RO(J,K)=-2.
12      CONTINUE
C
C       INITIALIZE THUMBWHEELS
C
        CALL INIT
C
C       PERFORM: SEPARATOR IN, SEPARATOR OUT,
C                TOTAL SEPARATION CALCULATIONS
C
        DO 30 KO=0,2
C
C       NOTE THAT WO, W1, AND W2 ARE MUTUALLY EXCLUSIVE
C
        TW(21)=0.
        TW(22)=0.
        TW(23)=0.
        TW(KO+21)=1.
C
C       IDENTIFY WHICH MODE OF OPERATION
C
        IF( KO .EQ. 0) WRITE (21,2000)
        IF (KO .EQ. 1) WRITE (21,2001)
        IF (KO .EQ. 2) WRITE (21,2002)
2000    FORMAT('1',//////,18X,'TABLE 4.  ',
     & 'STANDARDIZE MODE SIMULATION',/)
2001    FORMAT('1',//////,17X,'TABLE 5.  ',
     & 'SEPARATOR OUT MODE SIMULATION',/)
2002    FORMAT('1',//////,16X,'TABLE 6.  ',
     & 'TOTAL SEPARATION MODE SIMULATION',/)
        WRITE (21,199)
```

```
199     FORMAT(10X,'PRODUCT',5X,'STD',4X,'MILK  SLURRY',
     &  ' A CREAM S CREAM',3X,'RATIO')
C
C       NOW SET FOR CALCULATION IN RECYCLE
C
        WRITE (21,1000)
1000    FORMAT(13X,'STARTUP')
        TW(18)=0.
C
C       PERFORM 1202 SIMULATION
C
        CALL PROGRA
C
C       OUTPUT RESULTS
C
        CALL WR
C
C       IF IN TOTAL SEPARATION MODE, QUIT
C
        IF (KO .EQ. 2) GO TO 30
C
C       SET FORWARD FLOW
C
        TW(18)=100.
C
C       CALCULATE, OUTPUT RESULTS
C
        CALL PROGRA
        WRITE (21,1010)
1010    FORMAT(13X,'FORWARD FLOW, NO MILKO DATA')
        CALL WR
C
C       SET FORWARD FLOW, NEW MILKO-TESTER DATA
C
        TW(18)=101.
C
C       SET MILKO-TESTER DATA 0.1% HIGH
C
        TW(17)=TW(12)+10.
C
C       CALCULATE, OUTPUT RESULTS
C
        CALL PROGRA
        WRITE (21,1020)
1020    FORMAT(13X,'ONE PASS, 0.10% HIGH')
        CALL WR
C
C       CALCULATE SECOND PASS WITH NO NEW DATA
C
        WRITE (21,1025)
1025    FORMAT(13X,'SECOND PASS, NO NEW DATA')
        CALL PROGRA
        CALL WR
C
C       SET NEW DATA, REPEAT CALCULATION AND OUTPUT
C
```

```
      TW(18)=101.
      CALL PROGRA
      WRITE (21,1030)
1030  FORMAT(13X,'THIRD PASS, 0.10% HIGH')
      CALL WR
C
C     SET RECYCLE MODE
C
      TW(18)=0.
C
C     CALCULATE, OUTPUT RESULTS
C
      CALL PROGRA
      WRITE (21,1000)
      CALL WR
C
C     SET FORWARD FLOW, NEW DATA, 0.1% LOW
C
      TW(18)=101.
      TW(17)=TW(12)-10.
      CALL PROGRA
      WRITE (21,1050)
1050  FORMAT(13X,'MILKO DATA, 0.10% LOW')
      CALL WR
30    CONTINUE
      STOP
      END

COMMON BLOCKS

/.COMM./(+111)
TW       +0       C        +34      I        +43      A        +47      MQ       +50
FO       +51

SUBPROGRAMS CALLED

WR       INIT     PROGRA

SCALARS AND ARRAYS   "*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED

*K       1        *KO      2        *J       3        .S0003 4         .S0002 5
 .S0001 6         .S0000 7          *KO      10

 MAIN.    NO ERRORS DETECTED
```

```
      SUBROUTINE WR
C
C     THIS SUBROUTINE IS USED TO OUTPUT DATA
C     TO THE PRINT FILE IN A STANDARDIZED MANNER.
C
C     IT WILL BE CALLED ONLY FROM MAIN.
C     OUTPUT FORMAT WILL BE:
C
C     MODE1 MODE2 MODE3 MODE4 MODE5 MODE6 MODE7
C R0  X     X     X     X     X     X     X
C R1  X     X     X     X     X     X     X
C R2  X     X     X     X     X     X     X
C R3  X     X     X     X     X     X     X
C
      REAL TW,C,I,A,MQ,RO
      COMMON TW(28),C(7),I(4),A,MQ,RO(8,4)
      DO 30 K=1,4
      K1=K-1
30    WRITE(21,200) K1,(RO(J,K),J=2,8)
200   FORMAT(7X,'R',I1,7F8.0)
      RETURN
      END


COMMON BLOCKS

/.COMM./(+111)
TW      +0      C       +34     I       +43     A       +47     MQ      +50
RO      +51

SUBPROGRAMS CALLED


SCALARS AND ARRAYS   "*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED

*K1     1       *K      2       *J      3       .S0001 4        .S0000 5


WR      NO ERRORS DETECTED
```

```
      SUBROUTINE INIT
C
C     PROGRAM PURPOSE:
C
C     TO INPUT TO THE THUMBWHEEL VECTOR THE
C       CONSTANTS SET BY THE OPERATOR FOR A
C       PROBLEM PUBLISHED BY THE MANUFACTURER
C       AS A TEST CASE FOR THE MACHINE.
C     MOST MNEMONICS WERE DEVELOPED BY THE
C       MANUFACTURER, AND WERE USED FOR EASE
C       OF HISTORICAL COMPARISON.  ALL MNEMONICS
C       ARE LISTED IN TABULAR FORM WITHIN THE
C       BODY OF THE THESIS.
C
C     SUBROUTINE FUNCTIONS AS A LARGE BLOCK
C       DATA SUBROUTINE.
C
      REAL TW,C,I,A,MQ,RO
      COMMON TW(28),C(7),I(4),A,MQ,RO(8,4)
C
      INTEGER TENTH,SS,SP,FM,FC,FAC,ONEHUN,LPGAC,LP,FP,
     & FS,LPGC,LPGP,LPGS,MILKOD,MILKOC,WO,W1,W2,FK,LPGMIL,
     & ONETH,CO,C1,C2,C3,EOB,IO,I1,I2,I3
      DATA TENTH/1/,SS/2/,SP/3/,FM/4/,FC/5/,FAC/6/,
     & ONEHUN/7/,LPGAC/8/,LP/11/,FP/12/,FS/13/,LPGC/14/,
     & LPGP/15/,LPGS/16/,MILKOD/17/,MILKOC/18/,WO/21/,
     & W1/21/,W2/23/,FK/24/,LPGMIL/25/,ONETH/28/
C
      TW(TENTH)=10000.
      TW(SS)=1000.
      TW(SP)=100.
      TW(FM)=1000.
      TW(FC)=5000.
      TW(FAC)=100.
      TW(ONEHUN)=100.
      TW(LPGAC)=300.
      TW(LP)=100000.
      TW(FP)=500.
      TW(FS)=300.
      TW(LPGC)=350.
      TW(LPGP)=150.
      TW(LPGS)=250.
      TW(MILKOD)=0.
      TW(MILKOC)=0.
      TW(FK)=0.
      TW(LPGMIL)=200.
      TW(ONETH)=1000.
      RETURN
      END
```

COMMON BLOCKS

```
/.COMM./(+111)
TW      +0      C       +34     I       +43     A       +47     MQ      +50
RO      +51
```

```
SUBPROGRAMS  CALLED


SCALARS  AND  ARRAYS    "*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED

%IO              W1       1     %CO              LPGMIL  2      WO       3
 FK      4       LP       5     %EOB             FP      6      ONEHUN   7
 SS      10      LPGS     11     ONETH   12      MILKOC  13     MILKOD   14
%I3              LPGAC    15    %C3             %I2             FC       16
%C2              FAC      17    %I1              FS      20     TENTH    21
 LPGC    22      W2       23    %C1              LPGP    24     FM       25
 SP      26

INIT       NO ERRORS DETECTED
```

```
      SUBROUTINE PROGRA
C
C     THIS IMPLEMENTS THE SIMULATION OF THE
C     EXECUTION OF 1202 INSTRUCTIONS.  THIS IS THE
C     MODIFIED VERSION OF THE PROGRAM.  INSTRUCTIONS
C     ARE IMPLEMENTED AS SUBROUTINE CALLS.  SHEET
C     REFERENCES ARE MADE TO THE ORIGINAL PROGRAM
C     LISTINGS.
C
C     THIS IS THE STORED PROGRAM OF A 1202 PROCESSOR,
C     AS MODIFIED TO SUPPORT THE MILKO-TESTER
C     ADDITIONS.  UNFORTUNATELY, THE REQUIRED
C     DOCUMENTATION OF CHANGES WAS NOT MAINTAINED.
C
      REAL TW,C,I,A,MQ,RO
      COMMON TW(28),C(7),I(4),A,MQ,RO(8,4)
      INTEGER TENTH,SS,SP,FM,FC,FAC,ONEHUN,LPGAC,LP,FP,
     & FS,LPGC,LPGP,LPGS,MILKOD,MILKOC,WO,W1,W2,FK,LPGMIL,
     & ONETH,CO,C1,C2,C3,IO,I1,I2,I3,EOB
      DATA TENTH/1/,SS/2/,SP/3/,FM/4/,FC/5/,FAC/6/,
     & ONEHUN/7/,LPGAC/8/,LP/11/,FP/12/,FS/13/,
     & LPGC/14/,LPGP/15/,LPGS/16/,MILKOD/17/,
     & MILKOC/18/,WO/21/,W1/22/,W2/23/,FK/24/,
     & LPGMIL/25/,ONETH/28/,CO/1/,C1/2/,C2/3/,C3/4/,
     & IO/1/,I1/2/,I2/3/,I3/4/,EOB/7/
C
C     COMPUTE MODIFICATION FACTOR, STORE IN C3
C USE FORMULA:
C    MNEW=10000*(1-FF) + FF(M+10000*NEW*(FP-MILKOD)/FP)
C     WHERE:  FF IS THE 100'S IN MILKOC
C             AND NEW IS THE 1'S IN MILKOC
      CALL TN (MILKOC)
      CALL S1 (ONEHUN)
      CALL SAA (CO)
      CALL TN (FP)
      CALL S1 (MILKOD)
      CALL M2 (CO)
      CALL M1 (TENTH)
      CALL D1 (FP)
      CALL A2 (C3)
      CALL SAA (CO)
      CALL TN (MILKOC)
      CALL D1 (ONEHUN)
      CALL SAA (C1)
      CALL M2 (CO)
      CALL SAA (CO)
      CALL TN (ONEHUN)
      CALL D1 (ONEHUN)
      CALL S2 (C1)
      CALL M1 (TENTH)
      CALL A2 (CO)
      CALL SAA (C3)
C     SHEET 1
C     CONTROL
C     COMPUTE RATIO CREAM/MILK
      CALL TN (FC)
```

```
      CALL S1 (FK)
      CALL M1 (LPGC)
      CALL SQ (CO)
      CALL TN (FM)
      CALL S1 (FK)
      CALL M1 (LPGMIL)
      CALL M1 (TENTH)
      CALL D2 (CO)
      CALL R (7,2)
C     COMPUTE RATIO SLURRY/PRODUCT
      CALL TN (SS)
      CALL M1 (LPGS)
      CALL SQ (CO)
      CALL TN (SP)
      CALL M1 (LPGP)
      CALL M1 (TENTH)
      CALL D2 (CO)
      CALL R (7,0)
C     COMPUTE RATIO (ADD CREAM/CREAM OR SEP CREAM/CREAM)
      CALL TN (SS)
      CALL M1 (FP)
      CALL SQ (CO)
      CALL TN (FS)
      CALL M1 (SP)
      CALL S2 (CO)
      CALL SQ (C1)
      CALL TN (SS)
      CALL S1 (SP)
      CALL M1 (FC)
      CALL A2 (C1)
      CALL SQ (CO)
      CALL TN (FM)
      CALL S1 (FK)
C     SHEET 2
      CALL M2 (CO)
      CALL SQ (C2)
      CALL TN (SS)
      CALL S1 (SP)
      CALL M1 (FM)
      CALL A2 (C1)
      CALL SQ (C1)
      CALL TN (FC)
      CALL S1 (FK)
      CALL M2 (C1)
      CALL M1 (TENTH)
      CALL D2 (C2)
      CALL M2 (C3)
      CALL D1 (TENTH)
      CALL SAA (CO)
C     STORE SEP CREAM/CREAM
C     COMPUTE 60/40 BIT
      CALL A1 (ONETH)
      CALL M1 (WO)
      CALL D1 (TENTH)
      CALL R (7,3)
      CALL SAC (C2)
```

```
C        60/40 BIT, C2, INTERACTION WITH COMPUTED RATIOS
C        D IS THE 60/40 BIT (=0 OR =1), AND R IS THE RATIO
C
C        (10000-R)(1-D) + RD        ORIGINAL EQUATION
C        (2R-10000)D + R - 10000 ALGEBRAIC REDUCTION.
         CALL A2 (CO)
         CALL A2 (CO)
         CALL S1 (TENTH)
         CALL M2 (C2)
         CALL S2 (CO)
         CALL A1 (TENTH)
         CALL M1 (WO)
         CALL SAC (C2)
C FINISH COMP., ADD CREAM
         CALL TN (FM)
         CALL S1 (FAC)
         CALL M1 (SS)
         CALL M1 (LPGAC)
         CALL SQ (CO)
C        SHEET 3
         CALL TN (TENTH)
         CALL M1 (W1)
         CALL M1 (LPGP)
         CALL M2 (C1)
         CALL D2 (CO)
         CALL SAC (CO)
         CALL TN (TENTH)
         CALL A1 (TENTH)
         CALL S2 (C3)
         CALL M2 (CO)
         CALL D1 (TENTH)
         CALL A2 (C2)
         CALL R (7,1)
C        STD
         CALL TN (FAC)
         CALL M1 (W1)
         CALL S1 (FM)
C        SHEET 4
         CALL SQ (CO)
         CALL TN (WO)
         CALL A1 (W2)
         CALL M1 (FC)
         CALL A2 (CO)
         CALL M1 (SS)
         CALL SQ (C1)
         CALL TN (FP)
         CALL M1 (SS)
         CALL SQ (CO)
         CALL TN (FS)
         CALL M1 (SP)
         CALL S2 (CO)
         CALL SQ (CO)
         CALL TN (WO)
         CALL A1 (W1)
         CALL M2 (CO)
         CALL SQ (C2)
```

```
      CALL TN (WO)
      CALL A1 (W2)
      CALL M1 (FC)
      CALL SQ (CO)
      CALL TN (FAC)
      CALL M1 (W1)
      CALL A2 (CO)
      CALL SQ (CO)
      CALL TN (SS)
      CALL S1 (SP)
      CALL M2 (CO)
      CALL A2 (C2)
      CALL M1 (LP)
      CALL D2 (C1)
C     SHEET 5
      CALL R (2,0)
      CALL SAA (CO)
C     MILK
      CALL M1 (ONEHUN)
      CALL D1 (LPGMIL)
      CALL R (3,0)
      CALL TN (FM)
      CALL M2 (CO)
      CALL D1 (TENTH)
      CALL R (3,1)
      CALL RB (3,2)
      CALL RB (3,3)
C     NOW CONTINUE WITH SLURRY
      CALL TN (FAC)
      CALL M1 (W1)
      CALL SQ (CO)
      CALL TN (W1)
      CALL S1 (WO)
      CALL S1 (W2)
      CALL M1 (FM)
      CALL S2 (CO)
      CALL SQ (CO)
      CALL TN (WO)
      CALL A1 (W2)
      CALL M1 (FC)
      CALL A2 (CO)
      CALL M1 (SS)
      CALL SQ (C1)
      CALL TN (SS)
      CALL S1 (SP)
      CALL M1 (FM)
      CALL A2 (C2)
      CALL M1 (LP)
      CALL D2 (C1)
      CALL SAA (C1)
      CALL R (2,2)
      CALL TN (LP)
      CALL M1 (SP)
      CALL D1 (SS)
      CALL R (2,1)
      CALL RB (2,3)
```

```
C     SHEET 6
C     SLURRY
      CALL TN (SS)
      CALL M1 (LPGS)
      CALL SQ (CO)
      CALL TN (LP)
      CALL M1 (SP)
      CALL M1 (ONEHUN)
      CALL D2 (CO)
      CALL R (4,0)
      CALL TN (TENTH)
      CALL M1 (SS)
      CALL SQ (CO)
      CALL TN (LP)
      CALL M1 (SP)
      CALL M1 (FS)
      CALL D2 (CO)
      CALL R (4,1)
      CALL RB (4,3)
C     SHEET 7
C     ADD CREAM
      CALL TN (W1)
      CALL M1 (ONEHUN)
      CALL M2 (C1)
      CALL D1 (LPGAC)
      CALL R (5,0)
      CALL TN (W1)
      CALL M2 (C1)
      CALL M1 (FAC)
      CALL D1 (TENTH)
      CALL R (5,1)
      CALL RB (5,2)
      CALL RB (5,3)
C     SEP CREAM
      CALL TN (WO)
      CALL A1 (W2)
      CALL M2 (C1)
      CALL SQ (C1)
      CALL TN (ONEHUN)
      CALL M2 (C1)
      CALL D1 (LPGC)
      CALL R (6,0)
      CALL TN (FC)
      CALL M2 (C1)
      CALL D1 (TENTH)
      CALL R (6,1)
      CALL RB (6,2)
      CALL RB (6,3)
C     SHEET 8
C     PRODUCT
      CALL TN (LP)
      CALL M1 (ONEHUN)
      CALL D1 (LPGP)
      CALL SAA (EOB)
      CALL R (1,0)
      CALL TN (LP)
```

```
      CALL M1 (FP)
      CALL D1 (TENTH)
      CALL R (1,1)
      CALL TN (LP)
      CALL M1 (SP)
      CALL D1 (TENTH)
      CALL R (1,2)
      CALL R (4,2)
      CALL SAC (CO)
      CALL A2 (C3)
      CALL R (2,3)
      CALL TN (MILKOD)
      CALL R (1,3)
C     COUNT
      CALL TI (I0)
      CALL R (0,0)
      CALL TI (I1)
      CALL R (0,1)
      CALL TI (I2)
      CALL R (0,2)
      CALL TI (I3)
      CALL RR (0,3)
      RETURN
      END
```

COMMON BLOCKS

/.COMM./(+111)

TW	+0	C	+34	I	+43	A	+47	MQ	+50
RO	+51								

SUBPROGRAMS CALLED

S2	M2	SAC	D2	A2	S1
M1	D1	A1	RB	SQ	R
SAA	TI	RR	TN		

SCALARS AND ARRAYS "*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED

I0	1	W1	2	CO	3	LPGMIL	4	WO	5
FK	6	LP	7	EOB	10	FP	11	ONEHUN	12
SS	13	LPGS	14	ONETH	15	MILKOC	16	MILKOD	17
I3	20	LPGAC	21	C3	22	I2	23	FC	24
C2	25	FAC	26	I1	27	FS	30	TENTH	31
LPGC	32	W2	33	C1	34	LPGP	35	FM	36
SP	37								

PROGRA NO ERRORS DETECTED

```
      SUBROUTINE TN(K)
C
C     THIS IS A COLLECTION OF SUBROUTINE ENTRIES
C     WHICH IMPLEMENT THE SIMULATION OF OPERATION
C     OF INDIVIDUAL INSTRUCTIONS IN THE 1202
C     PROCESSOR.
C
C     THIS ENTRY SIMULATES THE READING OF A
C     THUMBWHEEL VALUE, ADDRESS K.
C
      REAL TW,C,I,A,MQ,RO
      COMMON TW(28),C(7),I(4),A,MQ,RO(8,4)
C
      IF (K.LT.1 .OR. K.GT.28) TYPE 100,K
100   FORMAT(' TW CALLED WITH K=',I12)
      A=TW(K)
      CALL TRUN (A)
      MQ=A
      IF (TW(18) .EQ. 101. .AND. K .EQ. 18) TW(18)=100.
      RETURN
C
      ENTRY SQ(K)
C
C     THIS ENTRY SIMULATES THE STORE MQ
C     AT REGISTER K
C
      CALL TRUN (MQ)
      IF (K.LE.0 .OR. K.GT. 7) TYPE 200,K
200   FORMAT(' SQ CALLED WITH K=',I12)
      C(K)=MQ
      RETURN
C
      ENTRY SAC(K)
C
C     THIS ENTRY SIMULATES THE STORE ACC
C     AT REGISTER K, AND CLEAR ACC
C
      CALL TRUN (A)
      IF (K.LE.0 .OR. K.GT. 7) TYPE 300,K
300   FORMAT(' SAC CALLED WITH K=',I12)
      C(K)=A
      A=0.
      RETURN
C
      ENTRY SAA(K)
C
C     THIS ENTRY SIMULATES THE STORE ACC
C     AT REGISTER K, AND RETAINS THE ACC
C
      CALL TRUN (A)
      IF (K.LE.0 .OR. K.GT. 7) TYPE 400,K
400   FORMAT(' SAA CALLED WITH K=',I12)
      C(K)=A
      RETURN
C
      ENTRY TI(K)
```

```
C
C       THIS ENTRY SIMULATES THE READ INVENTORY
C       REGISTER K INTO ACC
C
        IF (K.LE.0 .OR. K.GT. 4) TYPE 500,K
500     FORMAT(' TI CALLED WITH K=',I12)
        A = I(K)
        MQ=A
        RETURN
C
        ENTRY AI(K)
C
C       THIS SIMULATES THE ADDITION OF
C       THUMBWHEEL K TO ACC
C
        IF (K.LT.1 .OR. K.GT.28) TYPE 600,K
600     FORMAT(' AI CALLED WITH K=',I12)
        A=A+TW(K)
        CALL TRUN (A)
        MQ=A
        RETURN
C
        ENTRY SI(K)
C
C       THIS SIMULATES THE SUBTRACTION OF
C       THUMBWHEEL K FROM THE ACC
C
        IF (K.LT.1 .OR. K.GT.28) TYPE 700,K
700     FORMAT(' SI CALLED WITH K=',I12)
        A=A-TW(K)
        CALL TRUN (A)
        MQ=A
        RETURN
C
        ENTRY MI(K)
C
C       THIS SIMULATES THE MULTIPLICATION OF
C       THE MQ BY THUMBWHEEL K
C
        IF (K.LT.1 .OR. K.GT.28) TYPE 800,K
800     FORMAT(' MI CALLED WITH K=',I12)
        A=MQ*TW(K)
        CALL TRUN (A)
        MQ=A
        RETURN
C
        ENTRY DI(K)
C
C       THIS SIMULATES THE DIVISION OF MQ
C       BY THUMBWHEEL K
C
        IF (K.LT.1 .OR. K.GT.28) TYPE 900,K
900     FORMAT(' DI CALLED WITH K=',I12)
        A=MQ/TW(K)
        CALL TRUN (A)
        MQ=A
```

```
      RETURN
C
      ENTRY A2(K)
C
C     THIS SIMULATES THE ADDITION OF
C     REGISTER K TO ACC
C
      IF (K.LE.0 .OR. K.GT. 6) TYPE 1000,K
1000  FORMAT(' A2 CALLED WITH K=',I12)
      A=A+C(K)
      CALL TRUN (A)
      MQ=A
      RETURN
C
      ENTRY S2(K)
C
C     THIS SIMULATES THE SUBTRACTION OF
C     REGISTER K FROM THE ACC
C
      IF (K.LE.0 .OR. K.GT. 6) TYPE 1100,K
1100  FORMAT(' S2 CALLED WITH K=',I12)
      A=A-C(K)
      CALL TRUN (A)
      MQ=A
      RETURN
C
      ENTRY M2(K)
C
C     THIS SIMULATES THE MULTIPLICATION OF
C     MQ BY REGISTER K
C
      IF (K.LE.0 .OR. K.GT. 6) TYPE 1200,K
1200  FORMAT(' M2 CALLED WITH K=',I12)
      A=MQ*C(K)
      CALL TRUN (A)
      MQ=A
      RETURN
C
      ENTRY D2(K)
C
C     THIS SIMULATES THE DIVISION OF
C     MQ BY REGISTER K
C
      IF (K.LE.0 .OR. K.GT. 6) TYPE 1300,K
1300  FORMAT(' D2 CALLED WITH K=',I12)
      A=MQ/C(K)
      CALL TRUN (A)
      MQ=A
      RETURN
C
      ENTRY RR(K,J)
C
C     THIS SIMULATES THE DISPLAY OF DATA
C     INTO READOUT J DURING MODE K.
C     THIS INSTRUCTION ALSO CLEARS THE PC,
C     THUS STARTING A NEW CYCLE.
```

```
C
      ENTRY R(K,J)
C
C     THIS SIMULATES THE DISPLAY OF DATA
C     INTO READOUT J DURING MODE K.
C
      IF (K.LT.0 .OR. K.GT.7 .OR. J.LT.0 .OR. J.GT.7)
     & TYPE 1400,K,J
1400  FORMAT(' RR OR R CALLED WITH K,J='3I2)
      RO(K+1,J+1)=A
      RETURN
C
      ENTRY RB(K,J)
C
C     THIS SIMULATES THE BLANKING OF
C     READOUT J DURING MODE K
C
      IF (K.LT.0 .OR. K.GT.7 .OR. J.LT.0 .OR. J.GT.7)
     & TYPE 1500,K,J
1500  FORMAT(' RB CALLED WITH K,J='3I2)
      RO(K+1,J+1)=-1.
      RETURN
      END

COMMON BLOCKS

/.COMM./(+111)
TW      +0      C       +34     I       +43     A       +47     MQ      +50
FO      +51

SUBPROGRAMS CALLED

TRUN

SCALARS AND ARRAYS   "*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED

*K      1       *J      2

 MAIN    NO ERRORS DETECTED
```

```
      SUBROUTINE TRUN (X)
C
C     THIS SUBROUTINE IS USED TO SIMULATE INTEGER
C     ROUNDING ON THE TARGET MACHINE
C
      IF (ABS(X) .GT. 2.0**33) RETURN
      IX = INT(X)
      IX = IX + IFIX (X-FLOAT(IX) + 0.5)
      X = FLOAT (IX)
      RETURN
      END

SUBPROGRAMS CALLED

INT.
IFIX.    FLOAT.   ABS.

SCALARS AND ARRAYS    "*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED

*X       1        *IX      2

TRUN        NO ERRORS DETECTED
```

VITA

David Cummings Herrell II was born in Knoxville, Tennessee on December 27,1952. He attended elementary schools in that area and graduated from Karns High School in June 1970. He then entered the University of Tennessee, Knoxville in June 1970.

In June 1974, he received a Bachelor of Science in Electrical Engineering with Honors. He began study toward a Master's degree in Summer 1974. He became a Teaching Assistant in the Electrical Engineering Department during Winter 1975.

He accepted employment with Computer Concepts Corporation during the summer of 1976 and was employed through September 1978. He then accepted employment with Technology for Energy Corporation, and will continue there after graduation. He received the degree of Master of Science, with a major in Electrical Engineering in June 1979.