August 12, 1957

MEMORANDUM TO: Mr. J. J. Ingram

SUBJECT: Stored-Program, Variable Word-Length, Core Storage Accounting Machine Proposal.

Furman file

The enclosed description represents my first effort in the Advanced Accounting Machine area. It is, of course, inaccurate and incomplete; the intent is merely to indicate the general form of the machine system I have in mind, and to provide a record of such effort at the earliest date.

Further development of the machine system must be preceded by a fairly extensive programming study, which I am presently pursuing.

F. O. Underwood

FOU:mak

cc: Messrs. M. E. Bear
P. M. Brannen
L. W. Brehm
R. G. Mork
J. K. Mosser
F. Saltz
J. H. Wellburn

Enc.

Description of Stored Program Variable-Word-Length Accounting Machine System

OBJECTIVES

The objectives to be met for the new Accounting Machine are:

1) 300 LPM List
2) Very high reliability
3) Calculating Ability
4) Functions and Rental to be such as to ensure a large market and a good profit to IBM

In essence, the machine is to be considered as the "next step" beyond the 407. Some of the secondary objectives are:

1) Variable word length
2) Complete Stored Program(No control panels)
3) Programming ease comparable with WWAM:
4) Mechanically simple Front Printer
5) Low diagnostic time
6) Adequate checking

Some of the secondary objectives perhaps need justification.

1) Variable word length-

The type of work performed by Accounting Machines is of a Variable word length nature, there being no standard or even "average" word length. No fixed word length machine can do a good job with Accounting problems. Once the variable word-length concept is accepted, many of the secondary objectives become easier to obtain, such as Read checking, print editing and the complete elimination of the control panel.

2) Complete stored program-

No way is known of providing a good checking scheme for control panels, so that it would seem desirable to eliminate all control panels, provided that all necessary functions can be provided with ease. Also, it is known that a good deal of equipment is necessary to drive electronic signals through a control panel, particularly when a large but indeterminate amount of logic may be wired on the board.

A third item of interest is the consideration of storage and maintenance of control panels in the customers office. The panels are large, heavy, awkward to handle, and require expensive space for storage. The contacts are not 100% reliable, and wires do come loose from the board and are often plugged in wrong holes.

It is also interesting to note the following: The cost of the manual control panels and plug wires that are furnished to the customer are considered as part of the manufacturing cost of the machine, and is, say, X% of the manufacturing cost. X% of the rental, then, is attributable to these control panels and wires, and may amount to something like $80 - $100 per month.

Over and above this cost is the additional investment that the customer must make for additional control panels and plug wires.

Another aspect of control panels is that a very effective limit is set on the expandability of the machine functions, because of the physical limits on the number of holes available on the panels.

These and other minor considerations point out quite strongly the desirability of a machine system which requires no control panels. It is admitted that control panels provide flexibility and that customers are generally familiar with plugged program techniques, but on the other hand, sufficiently powerful stored-program machines are also very flexible, and are coming into more widespread usage.

3) Programming ease comparable with WWAM.

The WWAM system provides considerable improvements in programming over machines such as the 604 and the 407.

WWAM is essentially a three-address-program machine, with data being transferred serially by digit. Such a system permits the execution of a program with a minimum of steps, and also permits the print edit functions to be performed with a minimum of plug wires.

By duplicating the scheme (in a practical way) by stored-program techniques, a machine system can be developed which meets all objectives of reliability, ease of programming and (hopefully) cost.

4) Mechanically Simple Front Printer

In order to achieve the high order of reliability, the mechanical components should be reduced to the simplest minimum, since it is expected that the mechanical portions of the machine will require the greater part of service time.

A front printer is considered mandatory, since it appears that the printing quality is considerably better than for a comparable back-printer.

It is also more adaptable to Bill Feed operation.

5) Low diagnostic time

If the reliability goal that we are seeking is really approachable, the Customer Engineer will have little contact with the machine circuits, so that when trouble does occur, powerful diagnostic aids must be available.

It is felt that such aids are more easily achieved with a stored-program type of machine system than with any other, particularly so if the machine system is designed with such operation in mind.

6) Adequate checking

"checking" is always a controversial subject, and who can really define what "adequate" checking is? A minimum approach is to provide Read checking and Print checking, and perhaps a check on the arithmetic, since it requires little, if any, additional cost to achieve.

Additional checking can be at the customers discretion, since it is possible for him to program what ever checks he feels are economically desirable.

Whether or not the proposed machine system meets the cost objectives cannot be determined without a great deal of effort. It does seem promising, however, when one considers that a considerable saving can be expected due to the elimination of the control panel and all of the attendent equipment.

The functional requirements will most certainly be met, either by the machine system as it stands or by minor modifications to it.

GENERAL DISCUSSION

The Stored-program, Variable word length Accounting Machine system to be described is based to a great extent upon the logic of WWAM. The major differences are that no control panels are used, and an entirely different printer mechanism is proposed.

All functional control of the machine resides in the core memory in the form of the "stored program". The program format is such that pilot selectors, co-selectors, line selectors, digit emitters, column splits, expanders, branch units, coincidence switches, electronic selectors, etc. are non-existent, but their functions are easily achievable.

The Storage arrangement, the addressing scheme and the instruction word format are all chosen to give the greatest degree of compatibility among all aspects of the functional components. All of core storage is alphanumeric (7-bit, 1248 ABC) with no functional assignment, that is, data and instructions may be stored anywhere within memory.

Core storage is arranged in blocks of 100 characters each, only for the purpose of providing efficient addressing. The machine is planned for a maximum of 48 blocks or 4800 alphanumeric characters of storage. This limit is set only by the addressing scheme.

A variable-word length machine requires some sort of "word-mark" to indicate the manner in which words are assigned in Storage. The device used in WWAM is the "Storage-Split", where as this machine is provided with an eighth plane of cores in the core memory, which is designated as the 'Word-mark' plane. A core in this plane is set to "1" at the location of the high-order character of each word in storage. Setting of these cores is accomplished by the program.

An "instruction word" consists of ten characters with four parts:

1) Read-out word location
2) Read-in word location
3) Operations code
4) Modifiers

The general form of the instruction word is written as follows:

$$A \; a_1 \; a_2 \; B \; b_1 \; b_2 \; M_1 \; M_2 \; C_1 \; C_2$$

where A is an alphanumeric character specifying the block location of the the RO word, a_1 and a_2 are the tens and units digits, respectively, of the units position of the RO word in the specified block, B is an alphanumeric character specifying the block location of RI word, b_1 and b_2 are the tens and units digits, respectively, of the units position of the RI word in the specified block. M_1 and M_2 are alphanumeric character, known as modifiers, and have many uses, particularly in performing the functions of pilot selectors, digit selectors, branching, indexing, etc. C_1 and C_2 are alphanumeric characters forming the OPERATION part of the instruction. The use of a 2-position alphabetic code for the operation permits the use of a mneumonic code which is a great aid towards achieving an easily programmed machine.

A typical instruction word might appear as follows: A42C93_ _AD and is interpreted as:

ADD the word whose units digit is located in Block A, position 42 to the contents of the word whose units digit is located in Block C, position 93.

The programming will be explained in greater detail later.

PRINTER

A new printer is proposed for this machine, since no printer was found which met the requirements.

These requirements are:

1) Front Printer.
2) Mechanically Simple.
3) A minimum of electronics required for control.
4) Will accept a "line-image" serially from storage.
5) 300 LPM Speed, minimum.
6) A fair portion of the print cycle can be reserved for carriage operation.

In order to achieve (1), (2) and (3), it was felt that a mechanism such as the 407 wheel mechanism, controlled by an electrostatic clutch would be an "optimum" mechanism. However, some doubt was expressed concerning the ability of the electrostatic clutch to drive the inertia of the 407 wheel and hanger mechanism. As an alternative then, the mechanism shown in figure 1 was evolved. This mechanism gives the same action as the 407 wheel (the selected type approaches the print line without vertical motion) and is a lower-inertia mechanism.

Referring to figure 1, the Constant Speed Belt Drive Shaft drives the 120 type belts around the belt idlers, past the print line. Each belt contains 48 characters (including one blank position) and is approximately .080" wide. The belts are made of some flexible, non-stretching material, such as nylon, spring steel, or perhaps a combination. The type is affixed to the outer surface of the belt, and drive teeth are affixed to the inner surface.

The belt idler bearing is carried on the Print Arm, and in its normal (restored) position permits a certain amount of slack in the belt.

For 300 LPM operation, the machine cycle is 200 ms in length. A 80 cycle-point print cycle was chosen (somewhat arbitrarily) which permits 48 points for actual printing (120 ms) and 32 points for carriage motion (80 ms).

This choice results in a cycle-point time of 2.50 ms. An increase in machine speed and/or an increase in the number of cycle-points per cycle is dependent upon the ability of the electrostatic clutch to drive the inertial loads at higher speeds, and also upon the ability to increase the character rate of the print buffer.

A character code drum (not shown) is geared to the Constant Speed Belt Drive shaft and is used to indicate to the Print Decision circuits the character on the belts that is being presented to the print line in any given cycle-point. (Since this printer is essentially a drum printer, the logic of such a system is well known and will not be discussed in detail here.)

When a decision to print is reached for a given print position, the corresponding electrostatic clutch is impulsed, causing the Print Arm to rotate, carrying the belt idler toward the print line.

Ideally, the center of the belt idler moves toward the print line at a velocity equal to the linear velocity of the belt, resulting in Zero rotational velocity of the belt idler, and therefore the selected type moves horizontally toward the print line with no vertical component. During this action, most (but not all) of the slack is removed from the belt.

At the termination of the printing action, the return spring causes the Print arm to restore to its normal position.

In order to operate the printer as a serial printer rather than a parallel printer, the constant Speed Belt Drive Shaft is given a "twist". The amount of twist is determined by the length of the printer cycle point and the character rate.

CARD FEED

One of the main objectives of this program is to have the card feed mechanism an integral part of the machine such as is the case with the Type 402-3 and Type 407.

This seems to be a very reasonable goal since the Card Feed mechanism and particularly the Print Mechanism Occupy very little space compared to the mechanisms proposed for WWAM.

This first proposal envisions only a single card feed with one drum stacker. More elaborate input systems will be considered in detail after intial cost and rental figures have been obtained for this first proposal.

Proceeding on this basis, then, the card feed is to be a "standard" card feed, positively-clutched, running at 300 CPM, employing parallel feeding past two 80-column brush stations. The number of cycle-points in the card feed cycle should be a sub-multiple of the number of printer cycle-points in order to permit simple time-sharing of the In/Out Buffer. For an 80-point printer, a 20-point feed is desirable. Another possible choice is a 72-point printer and a 24-point feed. A more complete discussion of time-sharing of the In/Out Buffer appears later.

PUNCH

Recent developments indicate that it is feasible to obtain either a 150 CPM or 300 CPM punch mechanism and mount it on the Accounting Machine base. If this is true, then it is conceivable that the Accounting Machine can have two card feeds on the Printer base, one of which can be used as a Summary Punch or Reader Punch. Such a concept will undoubtably generate a good deal of controversy.

The apparent advantages are that synchronism can be maintained between input, output and computing, with subsequent savings in packaging, control circuitry, and buffering.

In order to completely eliminate buffering on the smallest machine with summary punching, a 300 CPM, 20-cycle point punch mechanism is desirable.

SYSTEM LOGIC

The following discussion refers to the system diagram dated 7-29-57.

The major elements of the system are the Core Memory, card feed input, punch and printer output, a memory addressing system, arithmetic, print-idet controls, the "operation" register and the so-called "cycle-control."

There are three types of cycles which occur during compute time;

A) Program cycle. This is the first cycle which occurs at the beginning of a program step and consists of ten digit times, during which a program word is read out of storage.

The first digit time of the program cycle is called C_2 time, the second is called C_1 time, the third is called M_2 time, and the fourth is called M_1 time. These four characters of the program word are read into the operations register.

The next three characters (b_2, b_1, B) are the Read-in Word address, and are gated to set the Read-in Rings.

The last three characters (a_2, a_1, A) are the Read-out Word address and are gated to set the Read-out Rings.

The generation of the gating signals is under control of the units program ring and the addressing of the memory is under control of both the units and tens program ring and the program group ring. The drive to the units program ring is under control of the "cycle-control" circuits.

B) Read-out Cycle. Following the program cycle, the "cycle-control" circuits cause the Read-out rings to address memory for one digit time, and an alternate digit times thereafter, until a "word-mark" is sensed.

C) Read-in cycle. Following the first digit time of the Read-out cycle, the "cycle-control" circuits cause the Read-in rings to address memory for one digit time, and an alternate digit times thereafter until a Read-in "word-mark" is sensed. During this process, the read-in and read-out rings are advanced alternately.

Various modifications of this basic routine may occur, and will be described at the appropriate time.

One of the simplest operations is the "word-transfer" and a typical program word for performing this operation may appear as:
A43 C92 _ _TR

During the program cycle, the instruction TR is stored in the operations register at C_2 and C_1 times. There are no modifiers in this program word, so nothing enters the operation register at M_2 and M_1 times. At b_2 time, the "2" is read from storage and causes the units RI ring to be set to "2". At b_1 time, the "9" is read from storage and causes the tens RI ring to be set to "9", and at B time, the "C" is read from storage and enters the RI Group circuits, so that that Block C will be addressed during RI digit times.

Similar operations occur at a_2, a, and A times with the RO rings and RO Group circuits.

At the end of the program cycle, the "cycle-control" circuits cause the RO rings to address memory so that the character in position 43 of block A will be read out and stored in the Read-Out register (designated by Δ) and also regenerated thru the regeneration switch back into the same location in memory.

During the next digit time, the "cycle-control" circuits cause the RI ring to read out the character in position 92 of block C (setting that position to "O" for all bits.) During the "Write" portion of this digit time. the output of the RO register is written into position 92 of block C, effecting the transfer of one digit.

During this time, the RO ring advanced, so that during the next digit time, position 42 of block A is read out, and the RI ring advances. At the fourth digit time, then, position 91 of Block C will be addressed and the new character written there.

This process continues until a "word-mark" is detected, and when this happens, the "cycle-control" circuits cause the tens program ring to advance, thereby addressing the next program word.

An arithmetic operation such as "add" would require an instruction such as the following:

E72 M38 _ _ AD

During the program cycle, the operations performed are similar to that previously described for the "TRANSFER" operation.

At the end of the program cycle, the "cycle-control" circuits cause the RO rings to address memory so that the character in position 72 of block E will be read out and stored in the RO register and also regenerated thru the regeneration switch, back into the same location in memory, and the sign bit will be stored in the "Sign Control" circuits.

During the next digit time, the "cycle-control" circuits cause the RI rings to read out the character in position 38 of block M to one input of the adder and the sign of that character to the sign control circuits.

At this same time, the output of the RO register is presented to the other unput of the adder via the TRUE-COMPLIMENT circuit which has been set in accordance with the signs of both RO and RI words.

The output of the adder passes thru the Translator to the regeneration switch and enters memory during "Write" time of the Read-in cycle.

This process continues until a "word-mark" is sensed. Normally, the length of the Read-in word will be equal to the length of the Read-out word or else exceed it. If the RO word mark is sensed first, no further RO cycles occur for the remainder of the program step, and the RI cycles continue so as to propogate all carries in the RI field.

On any case, the program step terminates upon detection of the RI word mark, unless the sign of the RI word was changed during the arithmetic process.

When such is the case, the tens program ring is prevented from advancing, so that the program word is read out again so as to set the Read-out and Read-in rings to the proper addresses, although the RI address is the only one of importance.

Following the program cycle, a series of RI cycles occur in which the RI word passes thru the adder and is recomplimented until the RI word mark is again sensed, causing the program step to end, the tens program ring to advance, and the next program word to be read and executed.

On Reset Add and Reset Subtract operations, the operation is similar to a simple Transfer, except that the entire RI word (which may exceed the RO word in length) is set to blank on RI cycles and the RO word transferred to it.

Absolute Add and Absolute Subtract simply ignore the sign of the RO and RI words.

Other aspects of the programming features of this machine will be covered later.

Because no Pilot Selectors, digit selectors, column splits, etc., are provided, the card image must be stored in memory in the Hollerith code so that reference can be made to any punching position of any column of the card during the calculate portion of the machine cycle.

For this reason, all input areas of memory consist of 12 bits per position. All input information may be transferred to working storage thru an encoder immediately after card reading without destroying the information in the input area.

The input area of memory is loaded by a method similar to that employed in WWAM. A row-bit plane of 80 cores each are provided for each reading station and these are located directly above the corresponding input area of memory.

At each cycle point of the feed cycle, the sensing of a bit causes the corresponding row bit core to be set to "1". Between CB times, the input area of storage and the corresponding row-bit planes are scanned by the RO rings, causing the row-bit cores to be sensed, set to zero and the data from the second reading row bit plane regenerated into a corresponding bit plane of the input area of storage.

Each "1" bit sensed at the First reading row bit plane (which is used for checking only) advances a Modulo 16 counter as the row-bit plane is scanned. After card reading is completed, the 15's complement of the counter value is transferred to a second modulo 16 counter.

The card advances to Second Reading, setting the cores of the Second Reading row-bit plane. As this plane is scanned and data is entered into the input area of memory, the second modulo 16 counter is advanced by each "1" bit sensed.

At the completion of Card Reading, this counter is checked for a zero content to provide the read check.

Since no word-marks are provided for the input areas, special precautions are necessary when programming a RO word from the input area during compute time. Either a block transfer must be made of the input area to working storage, or else when individual words are addressed, the RI word must be of the same length as the RO word, since the word marks can appear only in the areas of storage other than the input areas.

In order to accomplish the functions of pilot selectors, coselectors, digit selectors, etc., a "Digit Test" instruction is provided so that any punching position of any column can be interrogated.

To illustrate a way in which a test could be made and used, suppose that two compute routines are written, one which starts with program word 12 and another which start with program word 18, and further that if a "7" is punched in column 72, the second routine is to be used, otherwise the first routine is to be used.

Also, suppose the program word 11 is to determine which routine is to be used, Program word 11 could appear as:

(11) I72 P18-7DT

The RO word is position 72 of the input block (The word is only one digit in length because of the operation DT) and the word is to be tested for the presence of a 7, written in position M_2 of the instruction.

If no 7 is present, the next program step is 12 (the next in succession). If a seven is present, then program step 18 will be called out on the next program cycle.

Another way that a digit test could control the program sequence is the following:

Assume that the problem is the same as stated above, except that it is desired to make the digit test on Program step 5, execute additional steps, and then branch on step 11 to step 43 or 18. In this case, step 5 could appear as (5) I72 P11 _ 7DS

This instruction will cause position I 72 to be tested for the presence of a 7, and if present, will cause program word 11 to be read out on this RI cycle and a character (say a minus sign) to be written in position M_2 of Program word 11. This operation can be thought of as analogous to the setting of a selecter. Then, as the program proceeds, and program word 11 is eventually read out, it would appear as:

(11) P43 P18 _ BR

with the minus sign appearing in position $\overline{M}_2$ if a 7 was present in column 72 of the card. The operation BR causes position M_2 to be interrogated for the sign and if present, the next step will be 18, otherwise it will be step 43.

Branching on a Balance Test is also possible. The instruction

(14) A53 B6274 AT

may be interpreted as meaning the following:

AT-add and balance test
A53-the RO word
B62-the RI word
and in position M_2,
a O means branch on zero,
1 means branch on non-zero
2 means branch on plus
3 means branch on minus
4 means branch on zero or plus
5 means branch on zero or minus
and in position M_1, the 7 would mean to branch by advancing the program seven words.
A minus sign in position M_1 would cause the program to go back seven steps.

If the Test is unsuccessful, then the program advances to the next step in succession (in this case, step 15).

The Balance test is, of course, made on the sign of the RI word after the Arithmetic has been performed.

It appears that a form of indexing is possible.

An instruction such as A 72 B 99 IIAd would be interpreted as follows:

AD- Add
A 72- the first RO word
B99- the first RI word and the I's in positions M_1 and M_2 as meaning " to index", that is, add the successive RO words to the successive RI words, for the remainder of the blocks.

If an I appears only in position M_1, it refers to the RO word, and the effect would be to add successive RO words to a single RI word location.

If an I appears only in the M_2 position, it refers to the RI word, and would cause the RO word to be added into all successive RI words of one block.

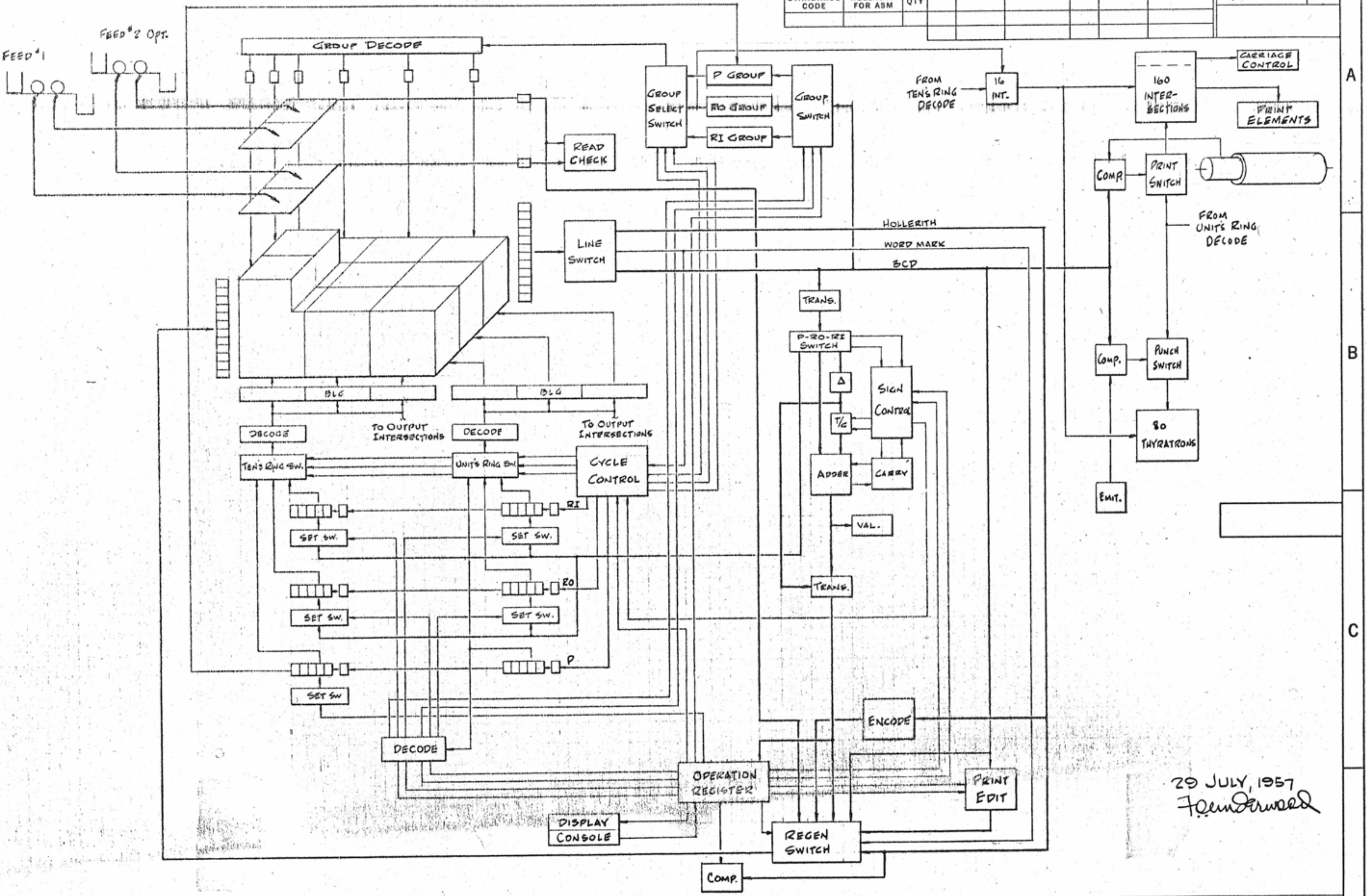
It has been previously mentioned that an 80- point printer cycle is proposed for this machine. Since card reading occurs for 12/20 cycle and printing occurs for 48/80 cycle (.6), nearly 80 ms are available for computing with an unbuffered, timeshared memory, provided printing and card reading occur during the same portion of the machine cycle.

In order to permit such operation, the printer sub-cycles determine the scanning rate of memory. Scanning of the Card Feed (actually, the row-bit planes and the input area of memory) occurs every fourth printer subcycle during Input-Output time, so that the 120 positions of the print area plus the 80 positions of the input area must be scanned sequentially during the 250 milliseconds of the printer subcycle. This sets the "Digit-rate" of the machine at 83 1/3 KC or 12 us per digit. This, then, is the required memory cycle time.

When card feed clutch decision time is considered, approximately 40-50 us of compute time is available, which appears to be more than adequate when one considers that only 12 us per digit is required in this machine, where 20 us per digit is required in WWAM.

F. O. Underwood
Department 631

STANDARDS CODE	RELEASED FOR ASM	QTY	SYM	DATE	CHANGE NO	DEVELOPMENT NO	Q/M
						4740	



29 JULY, 1957
F.O. Underwood

MATERIAL SPECIFICATION	NO
CASE DEPTH	
HARDNESS	
SURFACE TREATMENT	SPEC NO
TECH SERVICES APPRO	DATE

TOLERANCE UNLESS OTHERWISE NOTED	
DECIMALS	
FRACTIONS	
ANGLES	
CORNERS AND/OR EDGES BROKEN	OUTSIDE
	INSIDE

ALIGNMENT WITHIN		NOTE I
CONC TO	WITHIN	T I R NOTE II
FLAT WITHIN		NOTE III
PARALLEL TO	WITHIN	NOTE IV
STRAIGHT WITHIN		NOTE V
SQUARE TO	WITHIN	NOTE VI

INTERNATIONAL BUSINESS MACHINES CORP

NAME UNBUFFERED STORED PROGRAM ACCOUNTING MACHINE

DESIGN	FOU	7-29-57	MODEL	
DETAIL	FOU		SCALE	
CHECK	FOU		DRAW	
APPRO	FOU		CHECK	

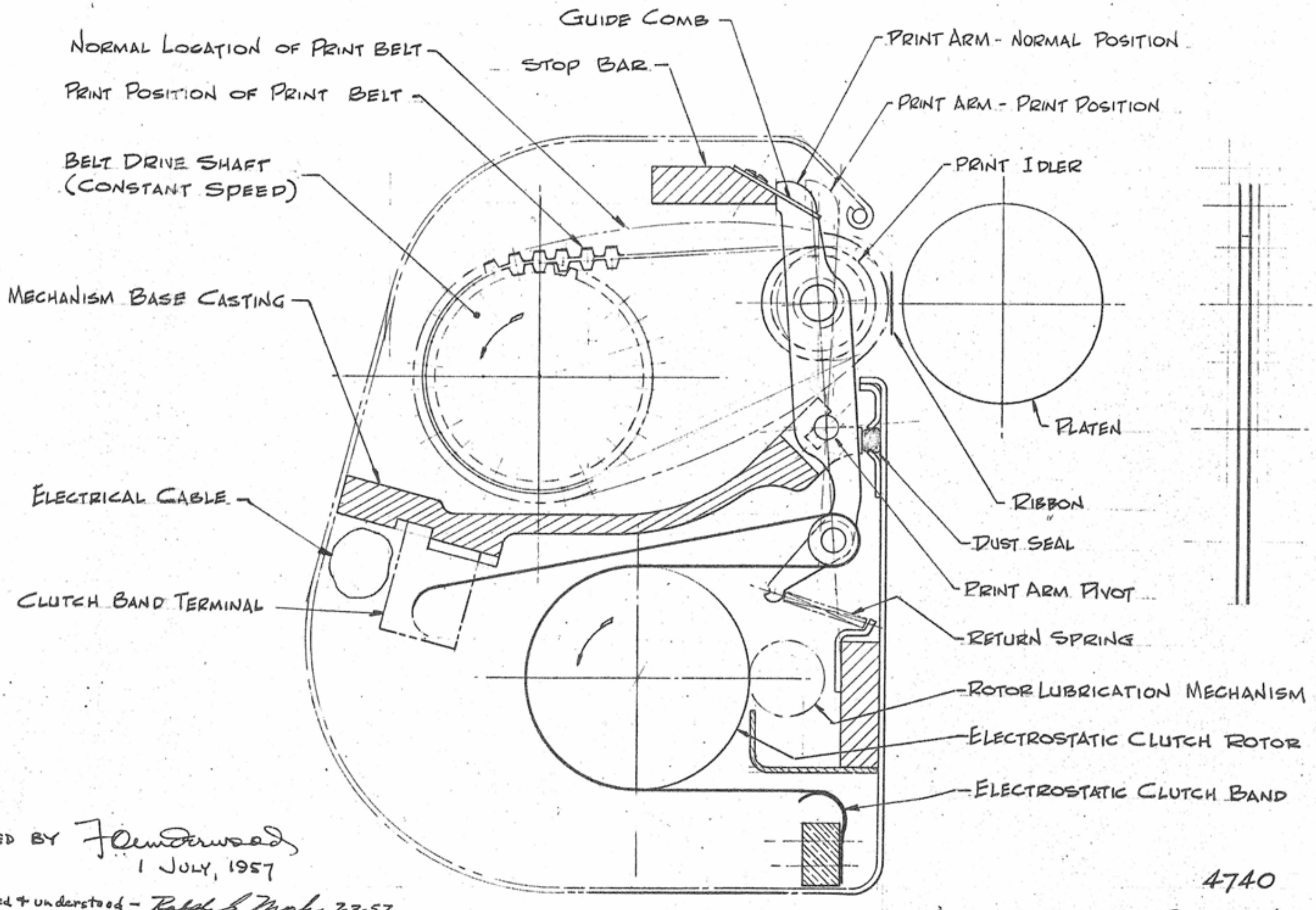


Fig. 1

4740
PRINT MECHANISM
1 JULY 1957
F Underwood

INVENTED BY F Underwood
1 JULY, 1957

Examined + understood - Ralph S. Mork 7-3-57
J J [illegible] 7-3-57

INTERNATIONAL BUSINESS MACHINES CORP.

DATE	CHANGE NO.	DATE	CHANGE NO.	DATE	CHANGE NO.
DRAW	CHECK	APPRO.	REDRAW	CHECK	
MACH.	MODEL	NAME			

STANDARDS CODE	RELEASED FOR ASSEM.	QTY.	NOTE X	PRINT TO ENG.	SPEC. NO.

DEFIANCE NO. 167H 12007

FORM 56-6593-0