Programmed Multiplication on the IBM 407*

ROGER L. BOYELL

The Ramo-Wooldridge Corporation, Los Angeles, California

Introduction

Although the method of progressive digiting has been used for accumulating sums of products[1] on an accounting machine, straight multiplication of two arbitrary factors (probably punched in the same card) is usually relegated to a calculating punch (IBM 602-A, 604, or 607). In high speed computer installations not including a calculating punch in the equipment complement, it may prove economical to use the versatile IBM 407 Accounting Machine for simple computations which take negligible computer time compared to card reading and punching. Normally the 407 has no ability to multiply. It can be wired, however, to perform accumulating and logic operations.

Advantage is taken of the multiple line reading and multiple program cycle features of the 407 to accumulate partial products for each multiplier digit and then crossfoot to form the final product. The multiplier digits are examined and no machine cycles are taken beyond those required for the greatest value digit in any card. Using this technique and summary punching the product reduces the effective speed of the 407 by about a factor of three, compared with conventional reading and summary punching of each card.

Detail Description

Two counters are required for each multiplier digit in the field. One subtracts the value of the digit and then adds unity on following cycles. The other accumulates a partial product for that digit. When the first counter reaches zero, the accumulation in the second is terminated. When all multiplier digit counters have reached zero, the partial products are crossfooted, and the full product is printed.

Figure 1 diagrams the wiring to multiply two 4-digit factors on a card, printing both factors and the product on the same line. Each card used in this operation has the multiplier in columns 7–10, the multiplicand in columns 17–20, and a Y-punch in column 79. The Y-punch initiates multiple line reading (MLR) and an intermediate special program. Coselectors 9, 11, 16 are picked on each MLR cycle after the first. Coselectors 9, 16 cause counters 3A, 3B, 3C, 3D each to subtract a digit of the multiplier on the first MLR normal cycle and then to add unity from the card count emitter on each MLR (normal or repeat) cycle following. Coselector 11 (section 5) eliminates printing except on the first MLR and the last program cycles. Coselector 11 (section 1–4) channels the multiplicand to the type wheels on the first MLR cycle and to

---

* Received May, 1957.

[1] An explanation is given in the Appendix.

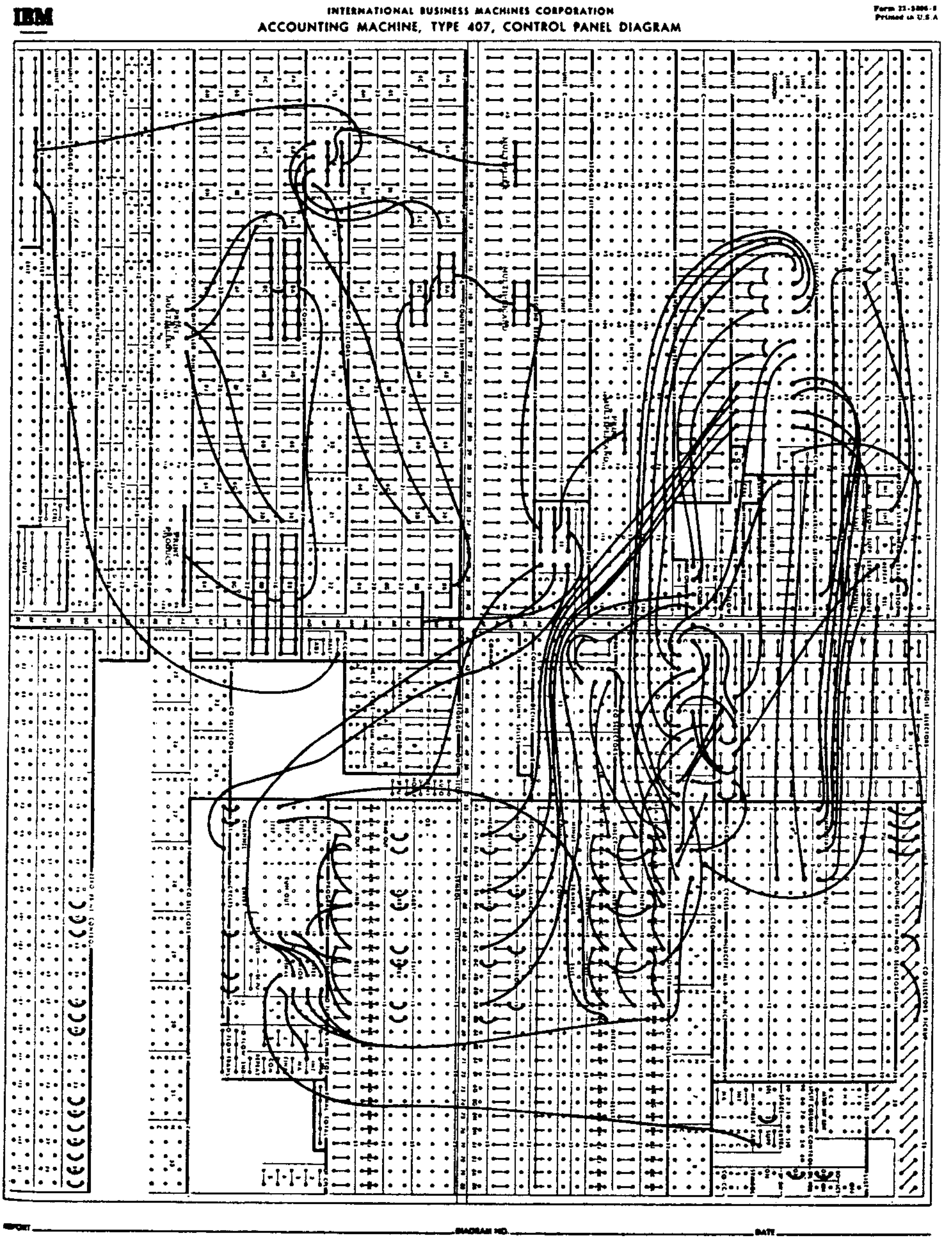


FIG. 1. Programmed multiplication
(4 digits by 4 digits)

the partial product counter that is also used as the final product counter (8D) on succeeding MLR cycles. This coselector also isolates the final product counter entry hubs on program cycles (when the counter will be adding through the exit hubs).

The four partial product counters 8A, 8B, 8C, 8D accumulate simultaneously, but each accumulates only as long as the respective multiplier digit counter 3A, 3B, 3C, or 3D is negative. When, while adding unity on each cycle, counter 3A reaches zero, counter 8A is stopped from adding the multiplicand. When all four multiplier digit counters reach zero, MLR repeat is stopped, and the program cycles roll the partial products all into counter 8D for printing. Control for this sequence involves progressive selector 2 (sections 24–27) and pilot selectors 1–4, which are coupled to coselectors 1–4. The MLR wiring causes two MLR normal cycles before zero to nine[2] MLR repeat cycles, as explained below.

Progressive selector 2 is picked on the second MLR normal cycle by internal wiring and then is picked on each MLR repeat cycle through pilot selector 6. (The repeat couple hub cannot be wired directly to the progressive selector couple hub as MLR normal cycle 2 will cause a back circuit. Pilot selector 6 remains transferred through all MLR repeat cycles, and progressive selector couple 1 causes picking of progressive selector 2 as well as coselectors 9, 11, 16 noted above). Thus on all MLR cycles after the first, the negative balance off hubs of the four multiplier digit counters are channeled to pilot selectors 1, 2, 3, 4 immediate pick hubs. The transferred state of one of these pilot selectors reflects a zero balance state in the counter which controls it.

Now at the end of any cycle at which counter 3A stands at zero, its negative balance off hub will emit a short impulse. Pilot selector 1 (and thus coselector 1) will transfer immediately and hold through all succeeding MLR cycles. Coselectors 2, 3, 4, operate similarly. By use of a set of bus hubs, progressive selector 2 (section 22) feeds MLR cycles impulses (after the first) to coselectors 1, 2, 3, 4 (section 1). By use of another set of bus hubs, when any of these coselectors is normal (zero balance not yet reached in the controlling multiplier digit counter), the MLR repeat hub is impulsed, thus causing another MLR cycle and allowing the multiplicand to add into the counters 8A, 8B, 8C, 8D.

Coselectors 1, 2, 3, 4 (section 2) are series-wired from progressive selector 2 (section 21) such that when all four coselectors are transferred (zero balance reached in all controlling multiplier digit counters) on any MLR cycle after the first, the MLR release hub is impulsed, and no unnecessary MLR cycles are taken.

Finally, coselectors 1, 2, 3, 4 (section 3) channel MLR cycles impulses (after the first) to each multiplicand counter 8A, 8B, 8C, 8D as long as the coselector remains normal (the particular multiplier digit counter not having reached a zero balance) in order that the multiplicand be added into only the appropriate counter.

When the MLR release hub is impulsed, MLR cycles cease, and the program cycles begin. In the example shown four program cycles are taken—one to roll each of three partial products into the fourth and one to print the result.

A cycle could be saved by rolling 8A into 8C and 8B into 8D on the same cycle and then rolling 8C into 8D on the next, but proper interconnection of

---

[2] Equivalent to the value of the greatest multiplier digit.

	MACHINE CYCLE	MLR NORMAL		MLR REPEAT CYCLES (ZERO TO NINE)					PROGRAM CYCLES			
		1	2	1	2	3	4	5	1	2	3	4
MACHINE COMPONENT	MLR REPEAT		IMP.	IMP.	IMP.	IMP.	IMP.					
	MLR RELEASE							IMP.				
	PILOT SEL. 6	N	N	T	T	T	T	T	N	N	N	N
	PROG. SEL. 1	T	N	T	T	T	T	T	N	N	N	N
	PROG. SEL. 2	N	T	T	T	T	T	T	N	N	N	N
	COSEL. 9, 11, 16	N	T	T	T	T	T	T	N	N	N	N
	PROG. STOP											IMP.
	NON-PRINT		IMP.	IMP.	IMP.	IMP.	IMP.	IMP.	IMP.	IMP.	IMP.	
THOUSANDS	CTR. 3A	−3	+1	+1	+1 *	+1	+1	+1				DIRECT RESET
	CTR. 8A		$+10^3$ M	$+10^3$ M	$+10^3$ M				READOUT RESET			
	COSEL. 1	N	N'	N	N	T	T	T	N	N	N	N
HUNDREDS	CTR. 3B	−2	+1	+1 *	+1	+1	+1	+1				DIRECT RESET
	CTR. 8B		$+10^2$ M	$+10^2$ M						READOUT RESET		
	COSEL. 2	N	N	N	T	T	T	T	N	N	N	N
TENS	CTR. 3C	−0 *	+1	+1	+1	+1	+1	+1				DIRECT RESET
	CTR. 8C										READOUT RESET	
	COSEL. 3	N	T	T	T	T	T	T	N	N	N	N
UNITS	CTR. 3D	−5	+1	+1	+1	+1	+1 *	+1				DIRECT RESET
	CTR. 8D		+M	+M	+M	+M	+M		+ (8A)	+ (8B)	+ (8C)	READOUT RESET
	COSEL. 4	N	N	N	N	N	N	T	N	N	N	N

FACTORS PRINTED FROM CARD

COSELECTORS 1, 2, 3, 4 TRANSFERRED

PRODUCT PRINTED FROM COUNTER

FIG. 2. Machine operation for Multiplier 3205

Legend: IMP.—Impulsed hub
N—Selector normal
T—Selector transferred
M—Value of multiplicand
*—Zero balance reached

the entries and exits would require using several additional selectors that may be needed for other operations. Also, crossfooting in this manner would partially eliminate the type wheel echo check. (Each partial product counter readout in the example has the echo check applied. The type wheels rotate to position on all program cycles, even though the hammers fire only on the last.)

Figure 2 charts the activity of important counters and selectors for each machine cycle when forming the product of a multiplicand M and a representative multiplier 3205. Taking each small group of items (marked off in figure 2) and following the activity from cycle to cycle, one can visualize the operation of the machine for comparison with portions of the wiring of figure 1. For example, on the fifth MLR repeat cycle, coselectors 1, 2, 3, 4 are all transferred. Tracing the impulses through sections 1 and 2 of these coselectors, one finds that the MLR repeat hub is no longer impulsed, but the MLR release hub is, thus terminating multiple line reading.

With conventional summary punch wiring, the product for each card may be punched from counter 8D on the last program cycle. In fact, the features available even in the 407 model A-1 allow endless modifications to this "basic multiply panel" to perform other operations besides straight multiplication.

Timing Considerations

The time required to form and print the product includes two MLR normal cycles, zero to nine MLR repeat cycles, and several program cycles to crossfoot the final product, all at 150 cycles per minute. It is desired to find the mean number of variable cycles taken, assuming random multiplier digits.

Let the multiplier be represented as

$$a = a_n \times 10^n + a_{n-1} \times 10^{n-1} + \cdots + a_1 \times 10 + a_0$$

where n is the number of multiplier digits in the field, and define the greatest digit as

$$a_m = \max_{0 \leqq i \leqq n} a_i .$$

Then the probability that any digit is no greater than an arbitrary digit j is

$$p(a_i \leqq j) = \frac{j+1}{10}$$

and the probability that no digit is greater than j becomes

$$p(a_m \leqq j) = \left(\frac{j+1}{10}\right)^n .$$

The probability that the greatest digit is exactly j is

$$p(a_m = j) = \left(\frac{j+1}{10}\right)^n - \left(\frac{j}{10}\right)^n .$$

This is then the probability that exactly j MLR repeat cycles will be required, where $0 \leqq j \leqq 9$. The mean of this distribution is

$$\bar{j} = \frac{\sum_j \frac{j[(j+1)^n - j^n]}{10^n}}{\sum_j \frac{(j+1)^n - j^n}{10^n}} .$$

However, the denominator of this expression reduces to unity (as should the cumulative distribution function).

$$\sum_j \frac{(j+1)^n - j^n}{10^n} = \frac{1^n + (2^n - 1^n) + (3^n - 2^n) + \cdots + (10^n - 9^n)}{10^n} = 1.$$

Then

$$\bar{j} = 10^{-n} \sum_j j[(j+1)^n - j^n]$$

where $\bar{j}$ is the mean number of MLR repeat cycles, in terms of the number of multiplier digits n. Figure 3 is a smoothed graph of the probability density $p\,(a_m = j)$ for $1 \leqq n \leqq 8$, and figure 4 is a plot of the mean $\bar{j}$ against n.

In the sample problem with $n = 4$ and using that particular method of cross-footing, the mean number of cycles is composed of:

2.00	MLR normal
7.47	MLR repeat
4.00	program
13.47	machine cycles per card.

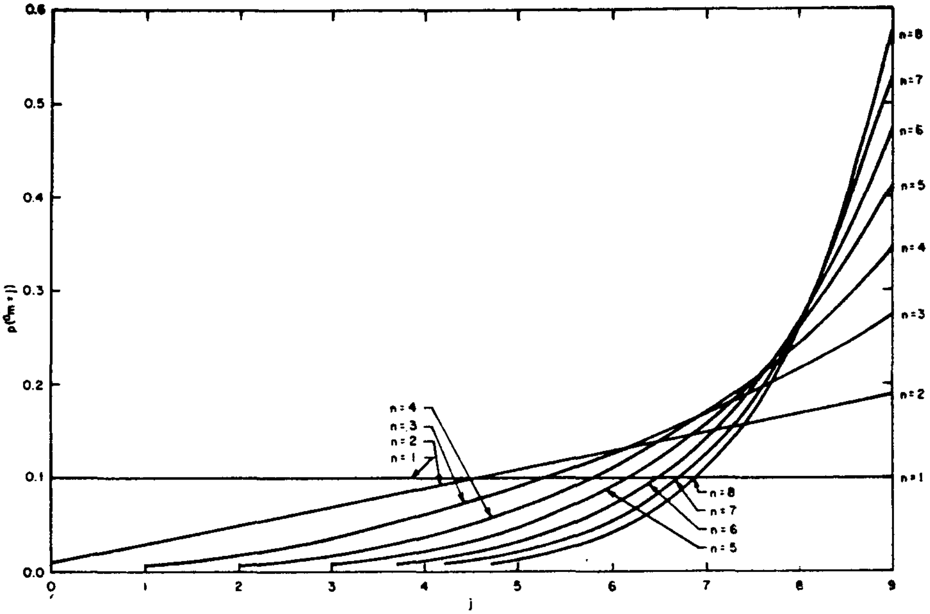
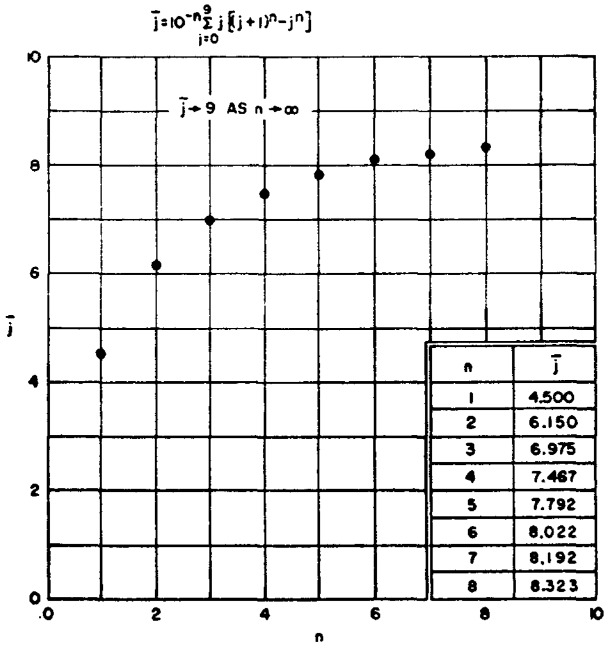


FIG. 3. Probability p that the greatest digit a_m in a randomly distributed number of n digits equals j



n	$\bar{j}$
1	4.500
2	6.150
3	6.975
4	7.467
5	7.792
6	8.022
7	8.192
8	8.323

FIG. 4. Average value $\bar{j}$ of the greatest digit in a randomly distributed number of n digits

Thus the effective speed is 11.14 cards per minute. As a check on the analysis two random-number decks of 111 cards each were run through the 407 with a panel wired as in figure 1. Each required just 10.0 minutes.

Application

This method of programmed multiplication on the 407 was of great utility in a computer installation[3] when the machine used weekly for a particular problem underwent extensive modification. Unfortunately this machine (the ORDVAC) was the only one available that had both multiplying instructions and the ability to handle alphabetic information.

The problem strongly resembled a payroll calculation. Hourly rate (from an employee master card) was adjusted by an amount dependent upon shift worked and multiplied by hours worked (from each detail card), the result being multiplied by 1.5 if the work was on overtime. A listing was required of all detail information by man name as well as the accumulated regular and overtime hours and pay for the week. A summary card was required for each detail item, duplicating the original card with the addition of the computed pay for those hours.

All this was performed in one pass of the cards through the 407, using a control panel ingeniously wired by Mr. Otto C. Juelich (now of North American Aviation, Columbus, Ohio) and based on the multiplying method described above.

A third MLR normal cycle was taken for adjustment of the rate for shift by subtracting one of three emitted quantities into the multiplier digit counters. If the overtime factor of 1.5 was applicable for any detail card (about a fifth of them) the product of rate by hours was crossfooted with five equivalent quantities shifted one digit to the right, requiring (only for overtime cards) four additional program cycles. With a 3-digit hourly rate as the multiplier and a summary card punched for each detail card the mean number of cycles was composed of:

3.00	MLR normal
6.98	MLR repeat
4.00	program
0.80	additional program for overtime cards
3.00	summary punch delay
17.78	machine cycles per card.

Thus the effective speed was 8.44 cards per minute. For the usual 500 cards for one week this required one hour of 407 time, which is not unreasonable. The ORDVAC, having no card input/output buffer and necessitating decimal-binary conversion and reconversion, could not reduce this time by even a factor of

---

[3] Computing Laboratory, Ballistic Research Laboratories, Aberdeen Proving Ground, Md.

three, in spite of the large disparity in cost. This is the basis for the thesis that "it may prove economical to use an IBM 407" for such computations.

APPENDIX

Progressive digiting consists of punching both factors in the same card, sorting in descending sequence by the multiplier field, merging with a descending sequence-number deck (having one card for each integer up to the greatest multiplier) and running the entire deck through an accounting machine. Two counters are used, one of which adds each multiplicand, and the other of which adds in the contents of the first for each sequence-number card. Thus the first counter contains the sum of all preceding multiplicands (higher value multipliers) and adds this amount into the second counter the proper number of times according to the multiplier value.

When the last sequence-number card has been read, the first counter contains the sum of the multiplicands, and the second counter contains the sum of the products, to be printed or summary punched. More than one multiplicand may be punched in each card, with two counters required for each field. Progressive digiting is useful in obtaining correlation coefficients, for which the individual products are not required. A slightly different method and an application are discussed in "Harmonic Analysis by the Use of Progressive Digiting", by H. R. J. Grosch (*Proceedings of the Research Forum, August 1946*, IBM, Endicott, New York).