February 6, 1958

Memorandum to: Mr. Eric Bloch

Subject: SIGMA Timing Simulation Studies

Enclosed is a File Memo giving a summary of the results to date of John Cocke's and my SIGMA timing Simulator studies. I hope they will shed some light on the questions you asked in your letter of January.

We are continuing our studies and hope to have some results concerning the use of special core memories for index registers in the near future.

Harwood G. Kolsky

Harwood G. Kolsky

HGK/jcv

cc: Mr. R. E. Merwin
Mr. L. E. Kanter
Mr. J. E. Pomerene
Mr. J. F. Dirac
Mr. R. J. Bahnsen
Mr. W. Wolensky
Mr. H. K. Wild

SAVG

Kolsky

Memo 1

PROJECT 7000 February 6, 1958

FILE MEMO

SUBJECT: First Report on results of SIGMA Timing Simulator Program

1. Introduction

The concept of an asynchronous computer as envisioned in the Project 7000 designs makes the evaluation of the true performance of the system a very difficult problem. Although one can do a considerable amount (and a considerable amount has been done by others) using averages and statistics gleaned from other computers, any actual system is so complicated in its asynchronous interconnections that these simple methods soon become suspect. In the final stages of design it becomes increasingly important to evaluate the performance quantatively because a big design effort in one area of the computer system can be spoiled by a relatively small deficiency in another. Some examples of this are contained in the results reported herein.

We are certainly the last to claim that the Simulator as presently coded or the test problems being used represent any absolute standard of perfection. We do hope, and have some reason to believe, that the results are reasonable and typical. Our real hope is that they also prove to be useful in influencing the designs.

2. Test Problems Used

To obtain quantitative results on the performance of the SIGMA system it is necessary to pick an actual problem coded in SIGMA language and run it on the simulator, then compare it against the time required to do the same problem on other computers coded in their respective languages. For this purpose we choose two test problems referred to as (a) the Mesh Calculation and (b) the Monte Carlo Calculation.

a. The Mesh Calculation is part of an actual hydrodynamics problem. It represents a problem typical of those done at Los Alamos. Its characteristics are: (1) It contains a large percentage of floating point arithmetic, (2) It has relatively few branches or decisions, (3) Most main memory references are indexed, (4) At least half the arithmetic is done with data located in the high-speed registers.

b. The Monte Carlo Calculation is a part of an actual Monte Carlo neutron diffusion problem. It represents a problem typical of calculations which are largely logical decisions, such as commercial problems, assembly programs, FORTRAN type programs, etc. Its characteristics are: (1) It contains little arithmetic of either floating point or integer type. (2) There are very many branches both of index type and data-controlled type. (3) Most main memory references are not indexed.

Both the above test problems contain about 250 instructions. Care was taken in coding the problems to stagger the data memory references so that they would neither always clash nor always avoid clashing.

The hope is that these two test calculations are representative of two large classss of problems which will be done on SIGMA. They should not be considered as extreme cases since it is not hard to find individual examples which will run a lot faster or slower than these.

3. The "Standard Design" of the Simulator.

Since the Simulator requires between 30 and 40 input constants as well as several logical specifications for a given run, we found it important to avoid varying too many parameters at once. For this reason, we picked a reference combination of constants called the "Standard Design", and made most of the studies as single parameter variations from it. Our original intention was to make the Standard Design the same as the machine specified in the Los Alamos brochure and contract, however, in trying to include some current design thinking and in working out the details of the Simulator, we find that we have departed somewhat from the original descriptions. The two main areas of difference are the bus times and the index arithmetic unit times, both of which are much slower than those indicated on the original hand-drawn timing charts included in the brochure.

Describing the Standard Design exactly would require a complete description of the simulator code, however some of the more important numbers assumed are:

a. Machine components:

1.	Levels of look-ahead	4
2.	Number of Instruction Memories	2
3.	Number of Main (data) Memories	4

b. Computer Speeds

1.	Indexing Time*	0.6 usec
2.	Arithmetic Unit Times	
	Fl Add	0.6 usec
	Fl Mpy	1.2 usec
	Fl Div	1.8 usec
	Fetch	0.2 usec

*This is total time to index one order, includes instruction decoding, index addition, and storing modified address.

c. Memory Speeds

1. Fast (Instr.) Memory Times

Read out time	0.4 usec
End signal time	0.4 usec
Memory cycle time	0.6 usec

2. Main (Data) Memory Times

Read out time	0.8 usec
End signal time	1.7 usec
Memory cycle time	2.0 usec

3. High Speed Register Times 0.1 usec

d. Bus Speeds

1. Buses to and from Memories 0.2 usec slot (either read or write) available every 0.3 usec

2. Decode and switching time in central control unit 0.2 usec

Note: A separate bus system to the Fast Memories and the Main Memories is assumed.

In addition to the above there is usually a 0.1 us delay between the completion of any function and the beginning of the next one by the unit. Also a 0.1 usec delay is inserted whenever there is a transfer of information from one register to another (for example from the IAU to the Look-ahead registers).

4. Results to Date

A list of the parameter studies done to date are listed in Appendix 1. The following summary data is available for each of the runs listed:

(1) Total time of run
(2) Percentage time busy for AU, IAU, each bus, each memory box, and each level of look-ahead.
(3) Percentage time arithmetic unit is waiting for an instruction or for a data reference.
(4) Percentage time there are conflicting memory or bus references.
(5) Average depth of look-ahead for run.

In addition to this summary data, we also have detailed timing charts for a few of the runs. This information is available for anyone who would like to study the detailed listings.

Appendix 2 consists of graphs of some of the runs showing the variation of computer speed vs various parameters. In each case the speed is in terms of a 704 version of the same problem,

Table 1 extracts a few of the key runs which demonstrate the most important results concerning machine speed which we have obtained to date.

Table 2 gives some more qualitative information concerning the "balance" of the SIGMA System.

TABLE 1

Summary of Main Effects on Computer Speed Studied with Sigma Timing Simulator

(Speeds are in terms of Number times 704 Speed for same problem.)

	Description of Run	Mesh Calc. Speed	Mesh Calc. % change	Monte Carlo Speed	Monte Carlo % change
1.	"Standard" Design (see Sect. 3)	100.	0	45.	0
2.	Arithmetic Speed Variation				
	(a) 2 x AU times (1.28 usec)	73.	-27%	43.	- 4.%
	(b) 3 x IAU times (1.4 usec)	67.	-33%	26.	-42.%
	(c) both 2 x AU & 3 x IAU	60.	-40%	24.	-46.%
3.	Fast (Instr.) Memory Speed				
	(a) 2.0 us Mem replacing FM	98.	- 2%	35.	-22.%
4.	Combining Instr. & Data Memories				
	(a) Instr. & Data in 4 boxes MM	82.	-16%	32.	-29.%
	(b) Instr. & Data in 6 boxes MM	86.	-14%	33.	-27.%
5.	Levels of Look-Ahead				
	(a) 2 levels Look-Ahead	89.	-11%	38.	-15.%
	(b) 6 levels Look-Ahead	106.	+ 6%	46.	+ 3.%
6.	Effect of Instr. Buffer above IAU			25.	-45%
	(a) No buffer	64.	-36%	~~44.~~	~~-1.%~~
	(b) No buffer & 1.4 usec IAU	44.	-56%	25.	-45.%
	(c) No buffer & 1.4 IAU & 1.28 AU	44.	-56%	25.	-45.%
	(d) No buffer & 2.0 us FM	55.	-45%		

TABLE 2

Some Qualitative Comparisons of the "Balance" of Different SIGMA Configurations

SYSTEM	DESCRIPTION
1. "Standard" Design	Fairly well balanced
2. #1 with 2.0 us Instr. Mem.	Still balanced for Mesh Calc. but Instr-fetch limited for Monte Carlo.
3. #1 without IAU buffer	Instruction-fetch limited.
4. #1 with slower IAU times	Index arithmetic unit limited.
5. #1 with slower AU times	Arithmetic unit limited for Mesh Calc. Monte Carlo still balanced.
6. #1 with less look-ahead	Main Memory time limited.
7. #1 with fewer boxes of MM.	Main Memory time limited.

5. Conclusions

(a) Arithmetic Speed

As may be seen from studying the graphs or the examples given in Table 1, the SIGMA performance is as sensitive or perhaps more sensitive to the speed of the indexing arithmetic unit as it is the main arithmetic unit. The curves on graph 1, speed vs. arithmetic times, show a "saturation" effect where the computer performance is independent of AU speed below some critical value. Clearly it makes no sense to strain on AU speeds if the IAU is not improved first. The curves on graph 2, speed vs. Index Arithmetic Unit time, show the same effect.

The curve for a given AU speed "peals off" from the saturated group at approximately the speed where the two units start to balance.

(b) The half microsecond memory

The Mesh calculation by itself showed a very low sensitivity to the speed of the instruction memory. However, the Monte Carlo calc. with its many branches is held up on instruction fetches at each branch if the instruction memory is made slower. (See graph 3) Note that if there is no instruction buffer above the IAU, the Mesh Calc. is degraded to near the level of the Monte Carlo Speed and then also becomes sensitive to the FM speed variation.

(c) Combining Instruction and Data Memories

Except when there is a branch instruction the instruction fetches are coming from consecutive memory locations, usually at two instructions per word. Thus, there is a regularity in information flow here which is not present in random data references. If data and instructions share the same memories, the evidence given in graph 4 indicates that the data fetches begin to block the smooth instruction flow and result in decreased performance, even when the total number of boxes is the same.

(d) Number of Levels of Look-Ahead

Graph 5 shows the effect on performance of the number of levels of look-ahead registers. Graph 5 also shows the same curves with slower AU and IAU speeds. These indicate that a large number of levels does the Monte Carlo problem less good, than the Mesh problems because of the constant branching. It also indicates that the look-ahead cannot pull up the performance of a slow machine single-handed.

(e) Instruction Buffer above the Indexing Arithmetic Unit

This register must be considered as part of the general IAU performance question. However, since the question of the presence or absence of such a register arose recently we ran off a few studies to show its effect. Table 1 and graph 3 show how drastic the effect was on the Mesh Calc. The Monte Carlo problem suffered only slightly. The explanation lies in the fact that the original Mesh Calculator is not instruction fetch limited whereas the Monte Carlo was. Removing the buffer register makes the Mesh Calculator also instruction fetch limited. The effect of the buffer is that of one level of look-ahead for the IAU.

In conclusion, the Simulator has emphasized the complex nature of an asynchronous machine. The whole computer system presents as analogy to a resonant circuit. If the frequency is far from resonance the circulating current is effected only slightly by separate element resistances. When one starts to "tune up" the curcuit, however, the resistance of an individual circuit element may prove a great limitation on the current. Our goal is to "tune up" the SIGMA system as a whole so that the effort for improvements will be directed where most needed.

John Cocke

Harwood G. Kolsky

John Cocke H. G. Kolsky

HGK:JC:jcv

APPENDIX 1

SIGMA Timing Simulator Runs Made to Date February 4, 1958

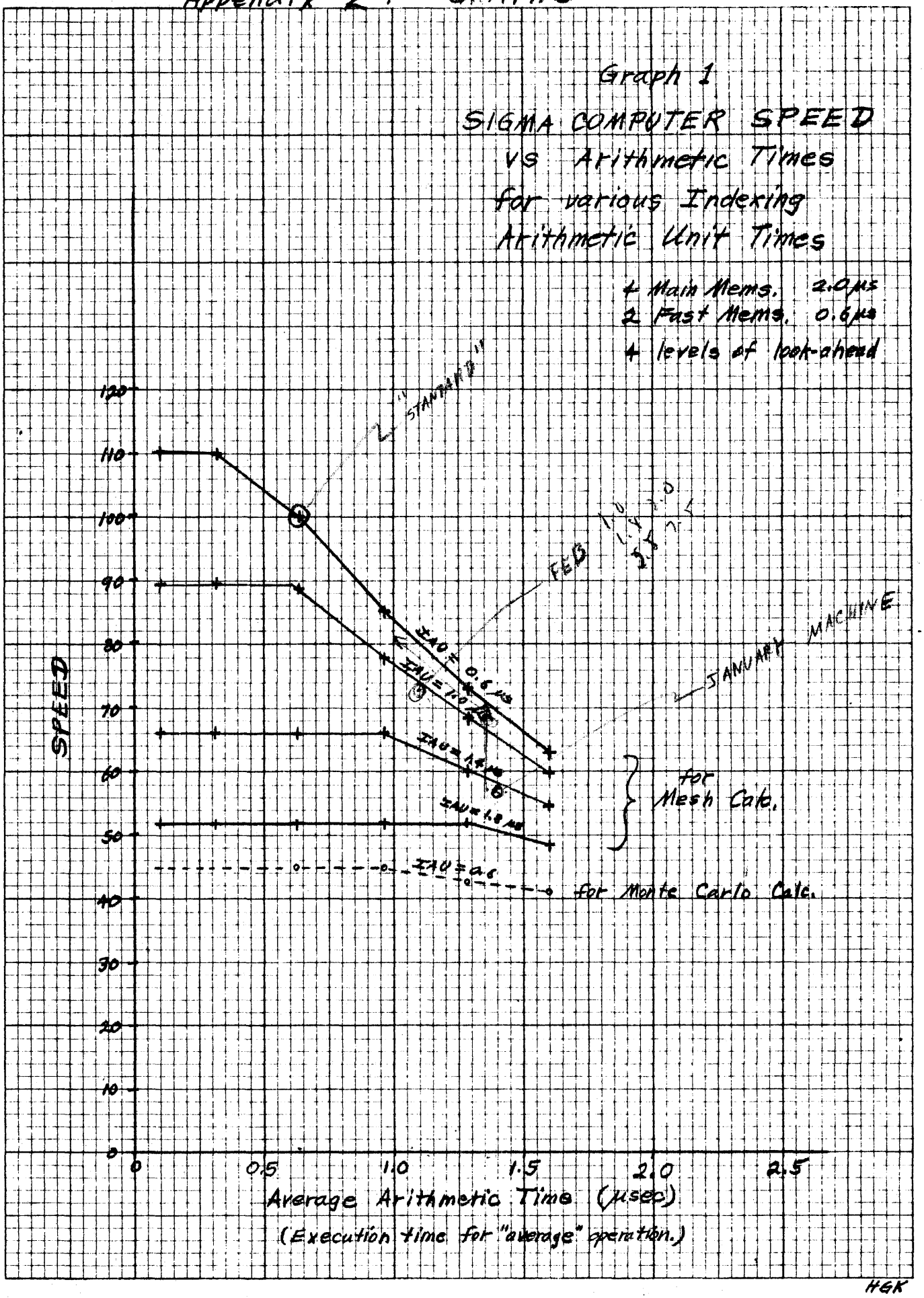
For Mesh Calculation

1. Varying AU speeds:Ratios 2, 2.5, 1.5, 0.5, 0.1
2. Varying No. LA: 1, 2, 3, 4, 5, 6, 7, 8
3. Varying No. FM: 1, 2, 3, 4
4. Varying No. MM: 1, 2, 3, 4, 5, 6, 7, 8
5. Varying FM Speeds: 0.3, 0.5, 0.6, 0.7, 0.8, 2.0 usec
6. Varying MM Speed: 2.0, 0.6 usec
7. Varying IAU Times: Ratios 4, 3, 2, 1.5
8. Varying AU Times with IAU Ratio 3: AU Ratio 1.0, 2.5, 2.0, 1.5, 0.5, 0.1
9. Putting whole program MM (indices, instr., and data)
10. Putting Data and Instr. in MM, Varying No. MM's: 8, 7, 6, 5, 4, 3, 2
11. Varying AU times with IAU ratio 2: AU ratio 2.5, 2.0, 1.5, 0.5, 0.1
12. Varying AU times with IAU ratio 4: AU ratio 2.5, 2.0, 1.5
13. Varying No. LA with AU ratio 2 & IAU ratio 3: 8, 7, 6, 5, 4, 3, 2, 1
14. Varying Mem. Decode time: 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8 usec
15. Without IAU Buffer: Varying AU times: Ratios 2.5, 2.0, 1.5, 0.5, 0.1
16. Without IAU Buffer: Varying FM Speed: 0.3, 0.6, 0.8, 1.4, 2.0, 2.5 usec
17. Without IAU Buffer: With AU ratio 2.0, IAU ratio 3.0
18. Without IAU Buffer: Varying No. LA: 8, 6, 4, 3, 2, 1
19. Without IAU Buffer: Varying No. MM's: 8, 7, 6, 5, 4, 3, 2, 1
20. Without IAU Buffer: Varying IAU Times: ratios 1.5, 2.0, 3.0, 4.0

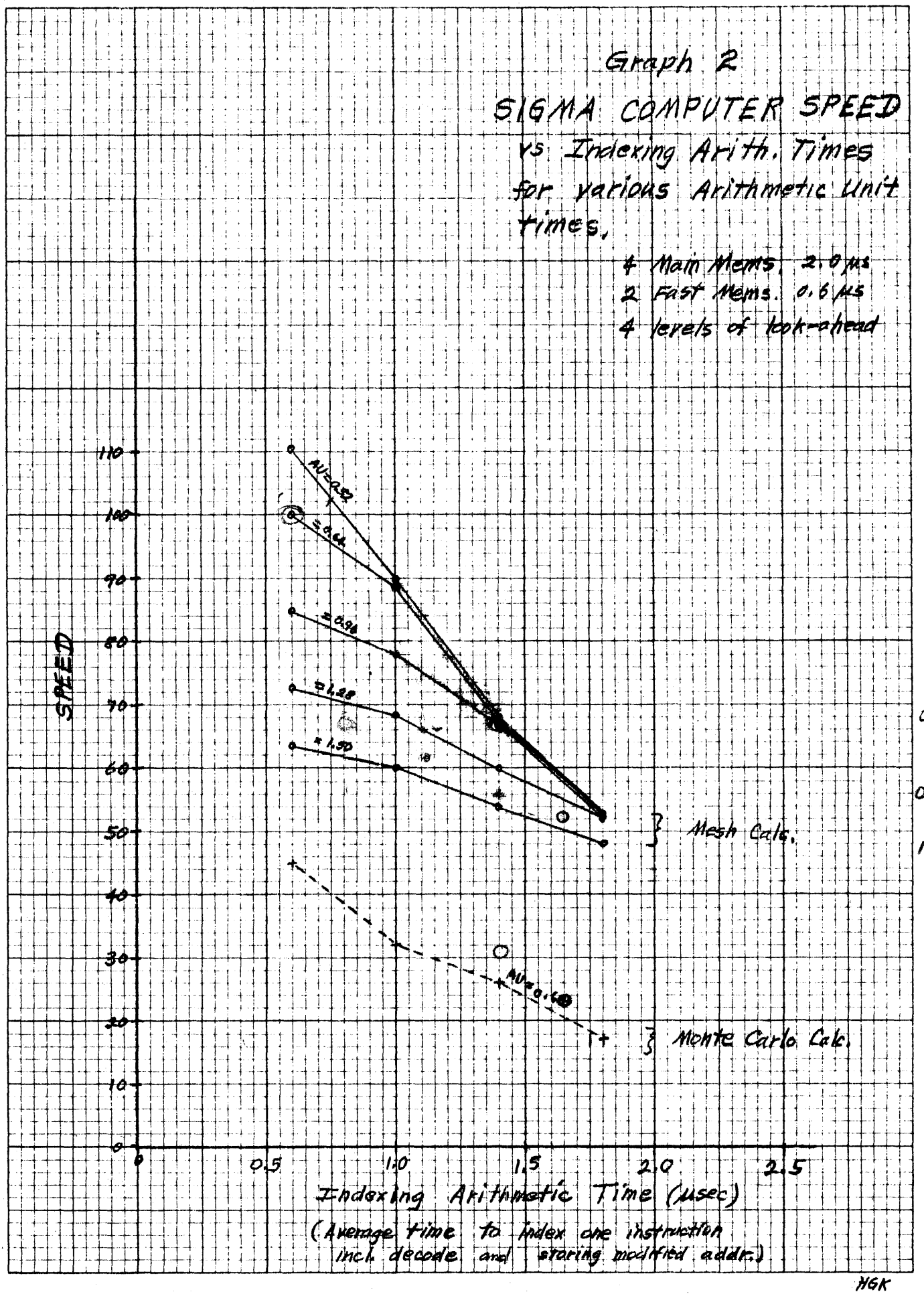
For Monte Carlo Calculation

1. Varying FM Speeds: 0.3, 0.4, 0.6, 2.0 usec
2. Varying AU Speeds: Ratios 2.5, 2.0, 1.5
3. Varying Memory decode time: 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8 usec
4. Varying No. FM's: 2, 3, 4
5. Varying No. MM's: 8, 7, 6, 5, 4, 3, 2, 1
6. Varying IAU Times:Ratios 2, 3, 4
7. Varying No. LA's: 1, 2, 3, 4, 5, 6, 7, 8
8. With AU ratio 2, IAU ratio 3, Varying No. LA's: 1, 2, 3, 4, 5, 6, 7, 8
9. Putting Data and Instruction in MM, Varying No. MM's: 4, 6
10. Without IAU Buffer, Varying FM times: 0.3, 0.6
11. Without IAU Buffer, Varying IAU times: Ratio 1.5, 2.0, 3.0

Appendix 2: GRAPHS



HGK



EUGENE DIETZGEN CO.
MADE IN U. S. A.

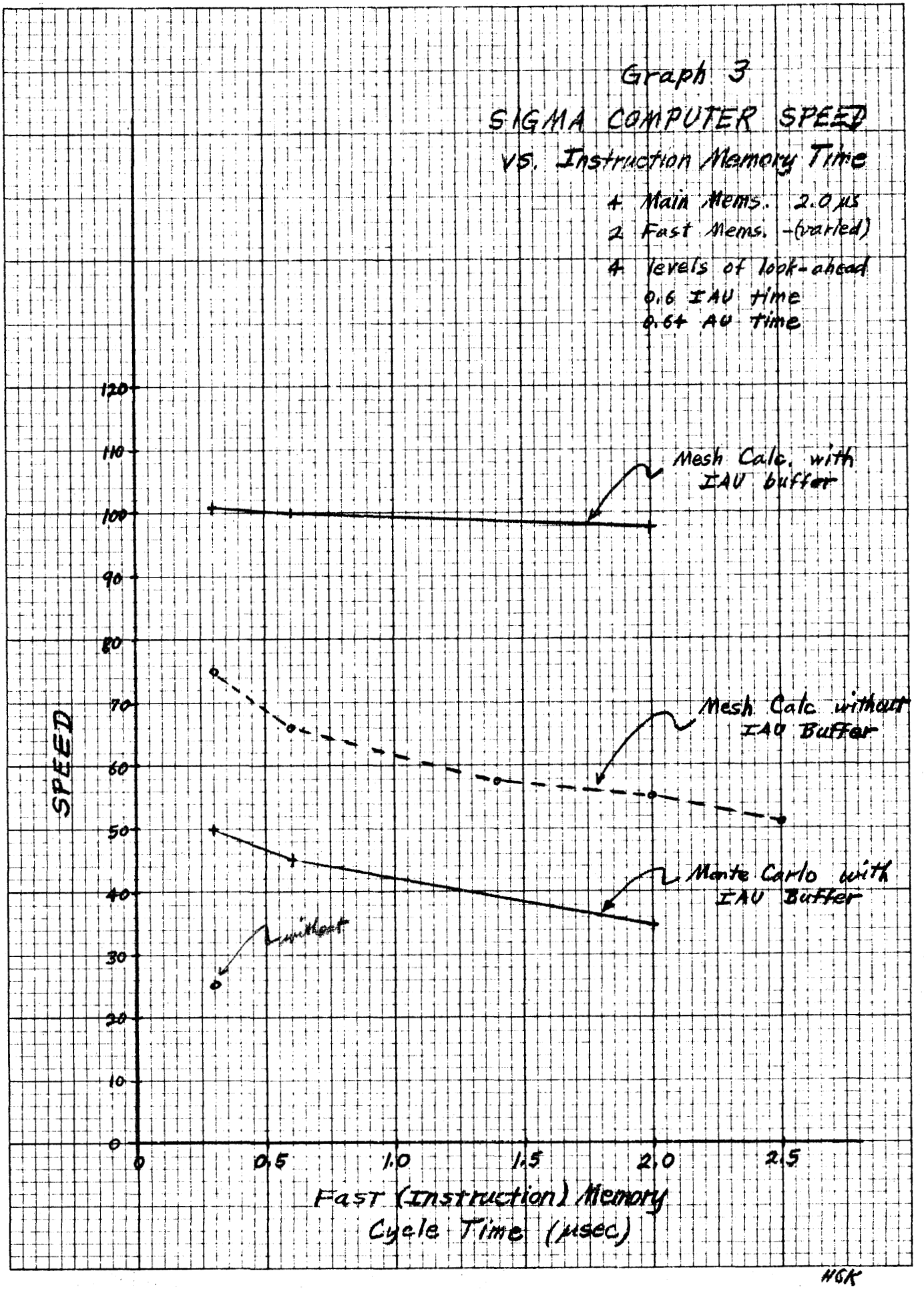
NO 340 ·10 DIETZGEN GRAPH PAPER
10 X 10 PER INCH

O "
times "

O = "
times

12 Feb

HGK



HGK