March 12, 1958

Project 7000

File Memo

Subject: Second Report on Results of SIGMA Timing Simulator Program

1. Introduction

To evaluate the performance of an asynchronous computer such as SIGMA one must get down to the detailed interaction of the components under typical operating conditions.

The first report on this subject (Project 7000 File Memo dated 2/6/58) describes how we have attempted to make quantitative measurements of the performance of the SIGMA computer using a Timing Simulator code written for the 704.

This report lists the studies which have been done since the last report. Most of them have been directed toward evaluating the effect of the recent proposed redesign of the indexing Arithmetic Unit and its interrelationship with the more realistic Arithmetic Unit Times now being quoted.

The effort of the past month has been in the direction of obtaining results as soon as possible, not in making the simulator a more precise mirror of the SIGMA machine. The results must, therefore be considered as approximate in details although the large trends should be essentially correct.

2. Test Problems Used

The main test problems used continue to be the mesh calculation and the Monte Carlo Calculation described in the February 6th memo. The Mesh calculation was used for the main IAU-AU studies since it uses a more or less "normal" distribution of indexing and arithmetic operations. (Reference: File Memo Dated March 5, 1958).

In addition to these, a few runs were also made on three problems which have been used by others in inter-comparing IBM machines with the LARC, the TRANSAC, etc. They are:

(1) The Westinghouse Reactor problem
The calculation of the inner loop of the numerical solution of a neutron diffusion equation. It is very heavy on arithmetic, very little logic.

(2) Ziller's Transac Test problem
The evaluation of a polynominal using computed indices.

(3) Matrix Inversion
The inner loop of a matrix inversion routine. Arithmetic and logic are approximately of equal importance. The shortness of the loops makes effective multiplexing difficult.

3. Designs Studied

The chief differences between the "standard design" described in the February 6th memo, and the designs being studied in this report are in the indexing Arithmetic Unit, the arithmetic unit times and the inclusion of index core memory. For convenience the other items which were not changed are also included in the following list:

The "January" and "February" Designs:

4. Results and Conclusions:

A list of the parameter studies run since the February 6th memo are given in Appendix I.

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

(a) SIGMA performance for various problems

Table I lists the speed of SIGMA on the five problems which have been tried to date. One striking feature is the range of speeds which appear — from 40. to 86. for the improved times. This points up the difficulty of giving a single speed performance figure. It also indicates that SIGMA is not just a "speeded-up 704", but a machine with considerably different organization.

SIGMA shows the biggest improvement over 704 in the problems which are largely floating arithmetic- - Mesh, Westinghouse, and Transac Test. It shows less improvement for the problems involving logic and indexing- - Monte Carlo and Matrix Inversion. (See graph 1)

(b) The effectiveness of the February Improvements in the IAU

All the problems showed an improvement as a result of the February improvements in the Indexing Arithmetic Unit. Those heavy on indexing naturally showed the most. (See table 1)

The variation of speed vs. IAU times for various Arithmetic times are shown in graphs 2 and 3. The important point to notice is that although the changes in AU and IAU are each worth about 10% in speed separately, taken together they make a 30% improvement. Graph 4 shows the AU efficiencies as a function of the AU and IAU times.

(c) The effect of Instruction Memory Speed

As was found in the previous runs, the Monte Carlo problem with its frequent branching is more sensitive to the instruction memory speed than the Mesh Calculation. However, with the present speeds, as contrasted to the higher "standard" speeds used before, the performance is only about half as sensitive to the change in memory speed as it was. (10% decrease instead of 22%). The positive effect of having more instruction memory does not appear in these figures.

(d) <u>The effect of the number of Instruction Memory Boxes</u>

Graph 5 shows rather conclusively that there appears to be no gain beyond two instruction memory boxes for these arithmetic speeds.

(e) <u>The effect of the number of Main (Data) Memory Boxes</u>

Graph 6 shows that the performance has become less sensitive to the number of Main Memories than was true using the "standard" speeds. There is still a pronounced loss if one mixes data and instructions, however.

(f) <u>The effect of changing divide speed only</u>

Because of interest expressed in the importance of divide speed alone, several runs were made with different divide times assumed. The results are that divide reduces the speed about the same as the change in the 6-6-3-1 average arithmetic time would predict. For example, changing the divide from 2.7 us to 9.0 us changes the average AU time from 1.09 us to 1.48 us which from Graph 3 implies a speed of 61., whereas the actual run gave 62.

(g) <u>The effect of number of levels of look-ahead</u>

Graph 7 shows the performance vs. number of levels of look-ahead for 4 Main memories and 1 main memory. The speed continued to rise past 4 levels but the gains become relatively small.

(h) <u>The effect of Index Core Memory times</u>

The use of a small core array for index register has been included in the Simulator since the previous runs. Graph 8 shows the effect of various assumed cycle times on the Mesh Calculation for three sets of Arithmetic speeds. Here again, the performance is less sensitive to core cycle times when the arithmetic speeds are low than when they are high.

The 0.8 us cores themselves seem to cause about a 11% reduction in performance from that of 0.3 us transistor registers at the February speeds. However, they also have the insidious effect of discouraging other improvements which might be possible now or later but which would be masked by the core cycle times if we put index cores in SIGMA.

(i) <u>The effect of varying the 2.0 us memory read out time</u>

Graph 9 shows that the performance is decreased by later read-out times, but that is not a strongly varying function for small changes.

(j) <u>The effect of simultaneous Input-Output upon computing speed</u>

A series of runs was made varying the average I/O word rate while the regular program was running. The Simulator assumes that a high speed disk is storing words in consecutive memory locations taking priority over other memory references. The effect on the Mesh Calculation was surprisingly small. The Mesh Calculation is favorable case, since the index registers are used in it to hold all intermediate results and these are not disturbed by the disk. More cases should be examined before making further generalization.

5. <u>Summary</u>

The improvements proposed since the January estimate are certainly very worth while. However, the performance is still about a factor of two below that expected in the Los Alamos contract.

The SIGMA system becomes percentage wise less sensitive to all variations when its speed is low than when it is high. We must be careful not to let this apparent insensitivity encourage us to drop 5% here and 5% there as being unimportant. The SIGMA system is very non-linear and these losses can add up to considerably more than 10%. Conversely we do not want to freeze one part of the machine at a low level which does not matter now but may block the effects of future gains elsewhere in the system.

Harwood G. Kolsky

Harwood G. Kolsky
Product Planning Representative
Project 7000

John Cocke

John Cocke
Staff Engineer
Project 7000

HGK:JC:jcv

Table 1: Summary of Main Effects on Computer Speed.

Unless otherwise stated runs were made with 4 Data Memories 2. 0 us, 2 Instruction Memories, 0. 6 us, 1 Index Core memory, 0. 8 us and 4 levels of look-ahead.

		Description of Run	SPEED		
			Jan. Est.	Feb. Imp.	% change
1.		Effects of IAU time change only			
	(a)	Mesh Calc (using AU= 1. 09 us)	62.	73.	+18. %
	(b)	Monte Carlo	29.	40.	+36. %
	(c)	Westinghouse Reactor Calc.	83.	86.	+ 4. %
	(d)	Transac Test Problem	64.	73.	+13. %
	(e)	Matrix Inversion	35.	44.	+26. %
		Average	55.	63.	+15. %
2.		Effect of IAU and AU changes separately. (Mesh Calc.)		Speed	% change
	(a)	Jan. Est. Times: I = 1. 45 us, AU = 1. 40 us		56.	0
	(b)	I = 0. 9 us, AU = 1. 40 us		62.	+10. %
	(c)	I = 1. 45 us, AU = 1. 09 us		62.	+10. %
	(d)	Feb. Imp Times: I = 0. 9 us, AU = 1. 09 us		73.	30. %
			SPEED		
3.		Effect of Instruction Memory Speed	0. 6 us FM	2. 0 us FM	% change
	(a)	Mesh Calc (with I=0. 9 us, AU=1. 40)	73.	71. 5	- 2. %
	(b)	Monte Carlo	40.	36.	-10. %
4.		Effect of changing Divide Speed only (IAU=0. 9 us		Speed	% change
	(a)	Mesh calc. with 1. 0 us Divide		75.	+ 3. %
	(b)	Mesh calc. with 9. 0 us Divide		60.	-18. %
5.		Levels of Look-ahead (IAU=0. 9, AU=1. 09)		Speed	% change
	(a)	Mesh Calc with 3 levels		69.	-5. 6%
	(b)	Mesh Calc with 5 levels		74.	+2. 3%
6.		Effect of Varying No. of Instruction Memories		Speed	% change
	(a)	Mesh calc with 1 0. 6 us FM		73.	0%
	(b)	Mesh Calc with 1 2. 0 us FM		64.	-12. %
7.		Effect of Varying X-Core cycle times		Speed	% change
	(a)	Mesh calc with 0. 4 us cores		81.	+10. %
	(b)	Mesh calc with 0. 2 us cores		83.	+14. %
8.		Effect of Varying 2. 0 us Data mem. read out time		Speed	% change
			0. 8 us RO	1. 2 us RO	
	(a)	Mesh calc with 2. 0 us FM	72.	71.	- 1. %
	(b)	Monte Carlo with 2. 0 us FM	36.	35.	- 3. %
9.		Effect of I/O memory interference		Speed	% change
	(a)	Mesh calc with I/O storing every 8. 0 us on average		71.	- 3. %
	(b)	Mesh calc with I/O storing every 2. 0 us		66.	-10. %

3/12/58

APPENDIX I

SIGMA Timing Simulator Runs Made February 4 to March 5, 1958

For Mesh Calculation

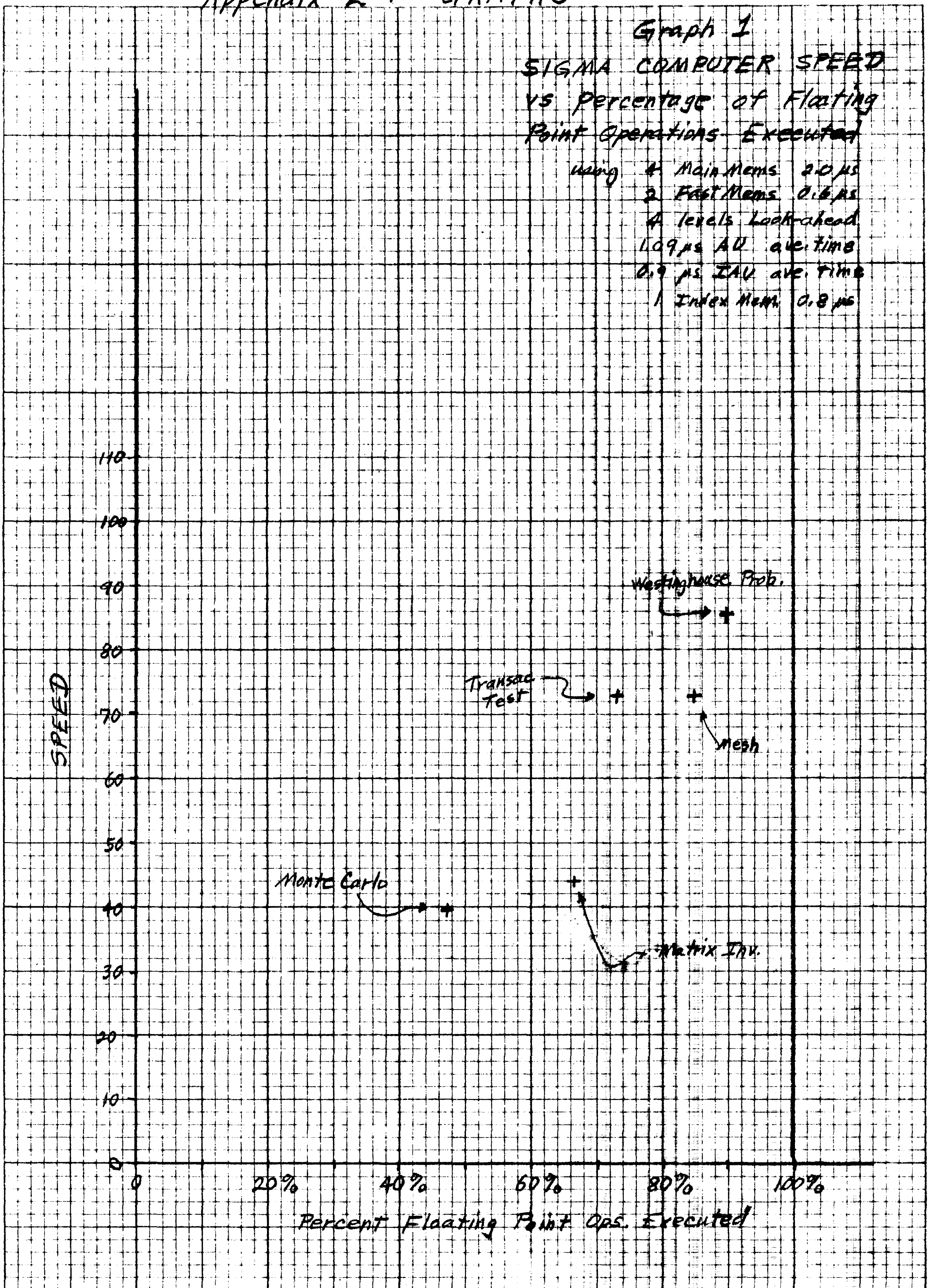
1. Varying X-Core times: 0.1, 0.2, 0.4, 0.6, 0.8 usec. with AU = 0.64; IAU = 0.6
2. #1 with AU = 1.40; IAU = 1.75
3. AU = 1.40; IAU = 1.75, X cores and 2.0 us memory
4. #3 with no IAU buffer
5. AU = 1.09, IAU = 1.75, Transistor X-register
6. Varying IAU times for 1.09 usec. AU, X-cores, 2.0 us FM IAU = 0.8, 0.9, 1.0, 1.15, 1.25, 1.35, 1.45, 1.55, 1.75
7. Varying AU times and IAU times, X-cores, 0.6 us FM AU = 0.29, 0.51, 0.79, 1.09, 1.35, 1.63 for IAU = 0.8, 0.9, 1.0, 1.1, 1.2, 1.4
8. Varying divide time only: 1, 3, 5, 7, 9 usec.
9. Varying No. Look-Aheads for February times: 1, 2, 3, 4, 5, 6, 7, 8
10. Varying I/O time: 16, 12, 08, 4, 2 usec. rate
11. Varying X Core times for February times: 2, 4, 6, 8 usec. cycle
12. January and February IAU for 1.09 us AU time
13. Varying No. Instruction Memories for 0.6 us and 2.0 us FM: 1, 2, 3, 4
14. Varying No. Data memories for 0.6 us and 2.0 us FM:1, 2, 3, 4, 5, 6, 7, 8
15. Data and Instruction both in MM: 1, 2, 3, 4, 5, 6, 7, 8
16. With 0.6 us MM, Data and Instruction both in MM: 1, 2, 3, 4, 5, 6, 7, 8
17. Varying No. Data Memories and no. levels look-ahead No. MM's : 1, 2, 3, 4, 5, 6, 7, 8, for No. Look Aheads: 1, 2, 3, 4, 5, 6, 7, 8
18. Varying MM read-out time: 0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.4, 1.6, 1.8, 2.0 usec.

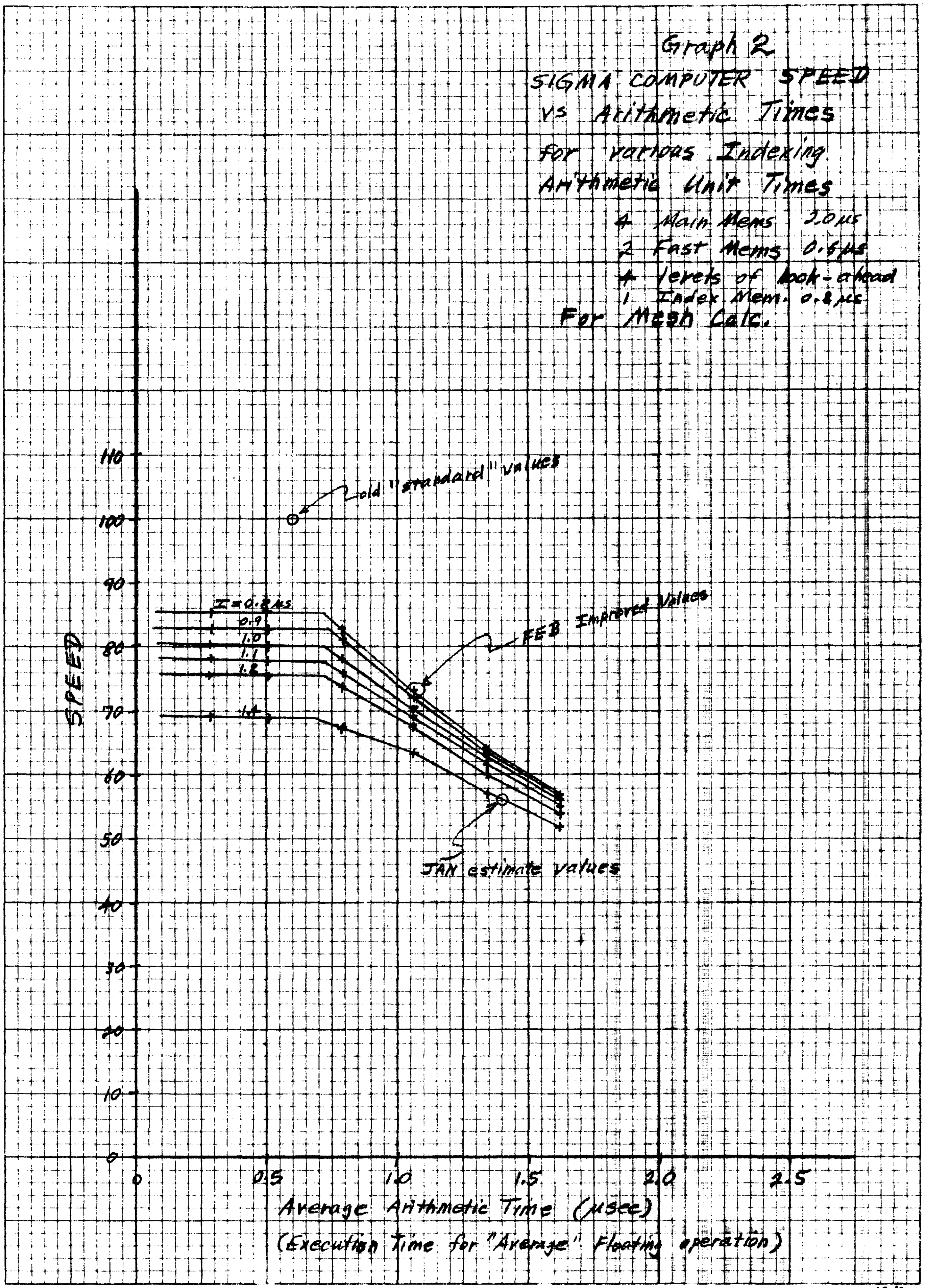
For Monte Carlo Calculation

1. Varying X-Core times: 0.1, 0.2, 0.4, 0.6, 0.8 usec. with AU = 1.40 and IAU = 1.75
2. AU = 1.40, IAU = 1.75 with Transistor X-registers
3. AU = 0.64, IAU = 0.6 with LA = 4, 8 Transistor X-registers
4. AU = 1.09, IAU = 1.75 with LA = 4, 8 Transistor X-registers
5. Varying IAU times for 1.09 usec. AU, X-Cores, 2.0 us FM IAU = 0.8, 0.9, 1.0, 1.15, 1.25, 1.35, 1.45, 1.55, 1.75
6. Varying AU times for 0.9 usec. IAU time, X-Cores, 2.0 us FM AU = 0.29, 0.51, 1.09, 1.35, 1.63
7. January and February IAU for 1.09 us AU
8. Varying No. Instruction Mems. for 0.6 us FM and 2.0 us FM: 1, 2, 3, 4
9. Data and Instruction both in Main Memory
10. Varying MM read out time: 0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.4, 1.6, 1.8, 2.0 usec.

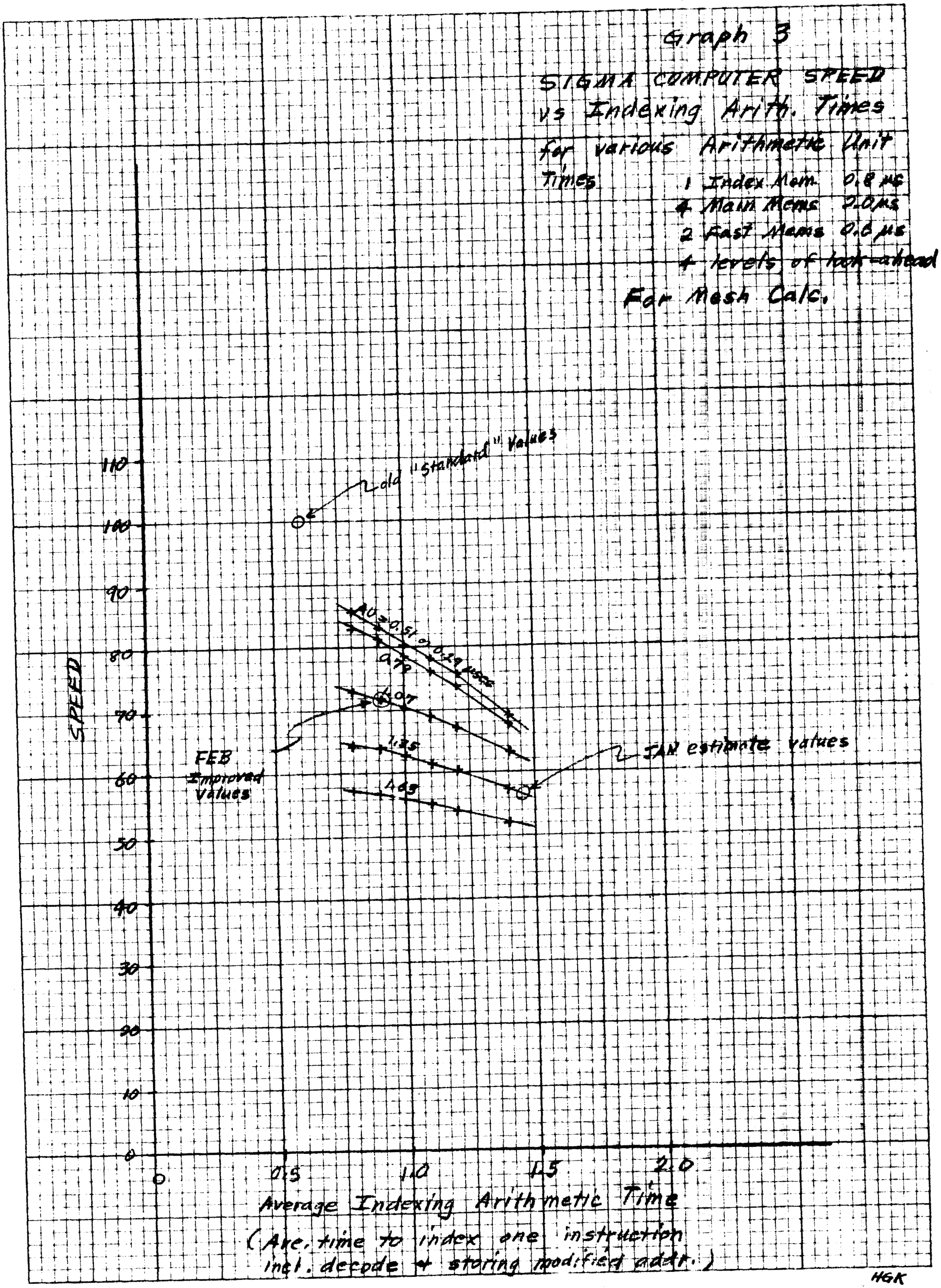
$\left.\begin{array}{l}\text{For Transac Test Problem}\\ \text{For Matrix Inversion}\\ \text{For Westinghous Reactor Problem}\end{array}\right\}$ each run with January and February IAU for 1.09 us AU

Appendix 2 : GRAPHS




Graph 2
SIGMA COMPUTER SPEED
vs Arithmetic Times
for various Indexing
Arithmetic Unit Times
4 Main Mems 2.0 μs
2 Fast Mems 0.6 μs
4 levels of look-ahead
1 Index Mem. 0.8 μs
For Mesh Calc.
old "standard" values
FEB Improved Values
JAN estimate values
I = 0.8 μs
0.9
1.0
1.1
1.2
1.4
SPEED
110
100
90
80
70
60
50
40
30
20
10
0
0
0.5
1.0
1.5
2.0
2.5
Average Arithmetic Time (μsec)
(Execution Time for "Average" Floating operation)

HGK



Graph 3
SIGMA COMPUTER SPEED
vs Indexing Arith. Times
for various Arithmetic Unit Times
1 Index Mem. 0.8 μs
4 Main Mems 2.0 μs
2 Fast Mems 0.6 μs
4 levels of look-ahead
For Mesh Calc.
old "standard" values
AU = 0.51 or 0.49 μsecs
0.79
1.07
1.35
1.65
FEB Improved Values
JAN estimate values
SPEED
110
100
90
80
70
60
50
40
30
20
10
0
0
0.5
1.0
1.5
2.0
Average Indexing Arithmetic Time
(Ave. time to index one instruction incl. decode + storing modified addr.)

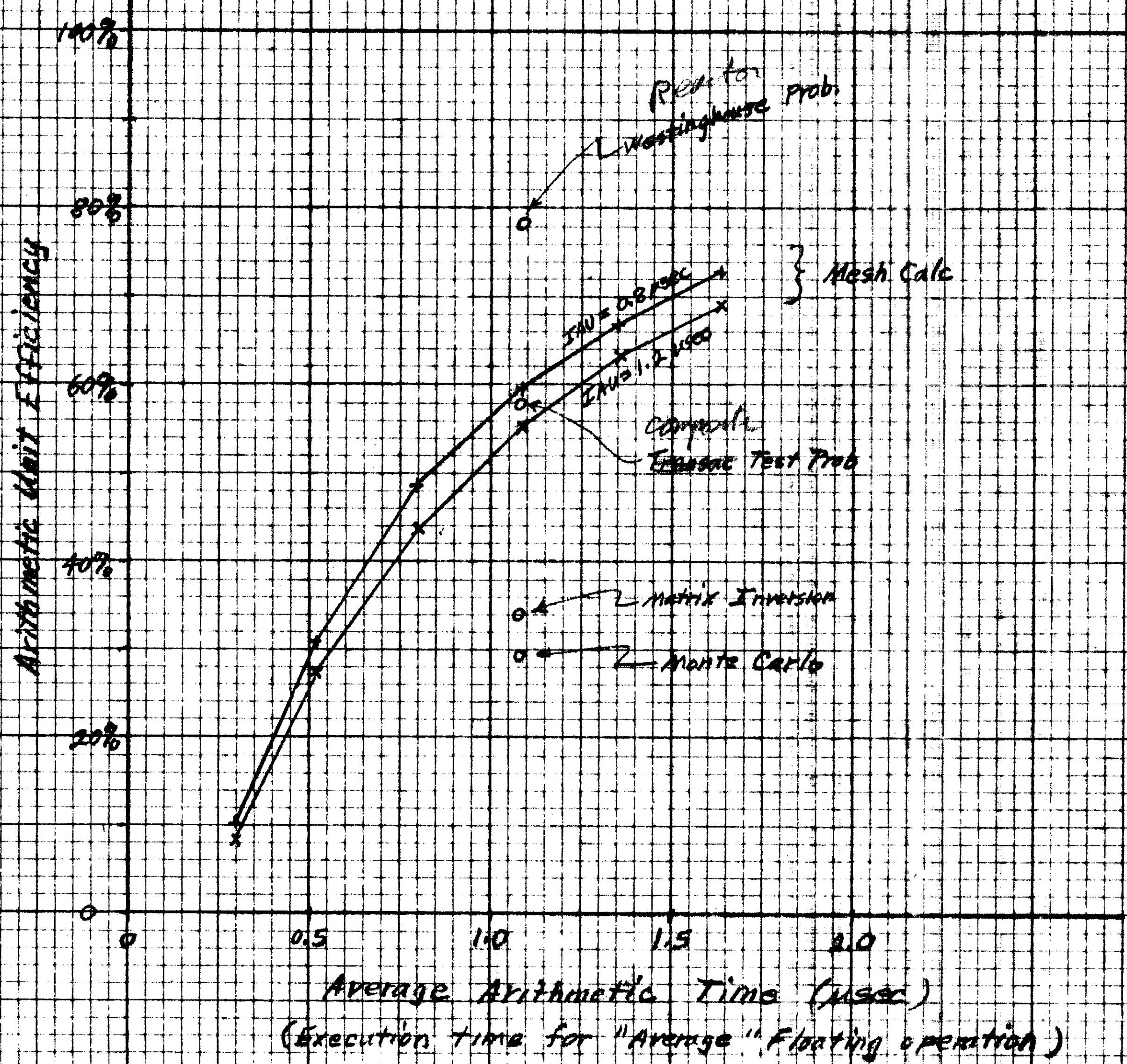
HGK

Graph 4

SIGMA Arithmetic Unit Efficiency vs Ave. Arithmetic Time for various cases.

1 Index Mem. 0.8 μs
4 Main Mems 2.0 μs
2 Fast Mems 0.6 μs
4 levels of look-ahead

$$\%\ \text{Efficiency} = \frac{\text{Time A.U. is operating}}{\text{Total Time for Problem}} \times 100$$

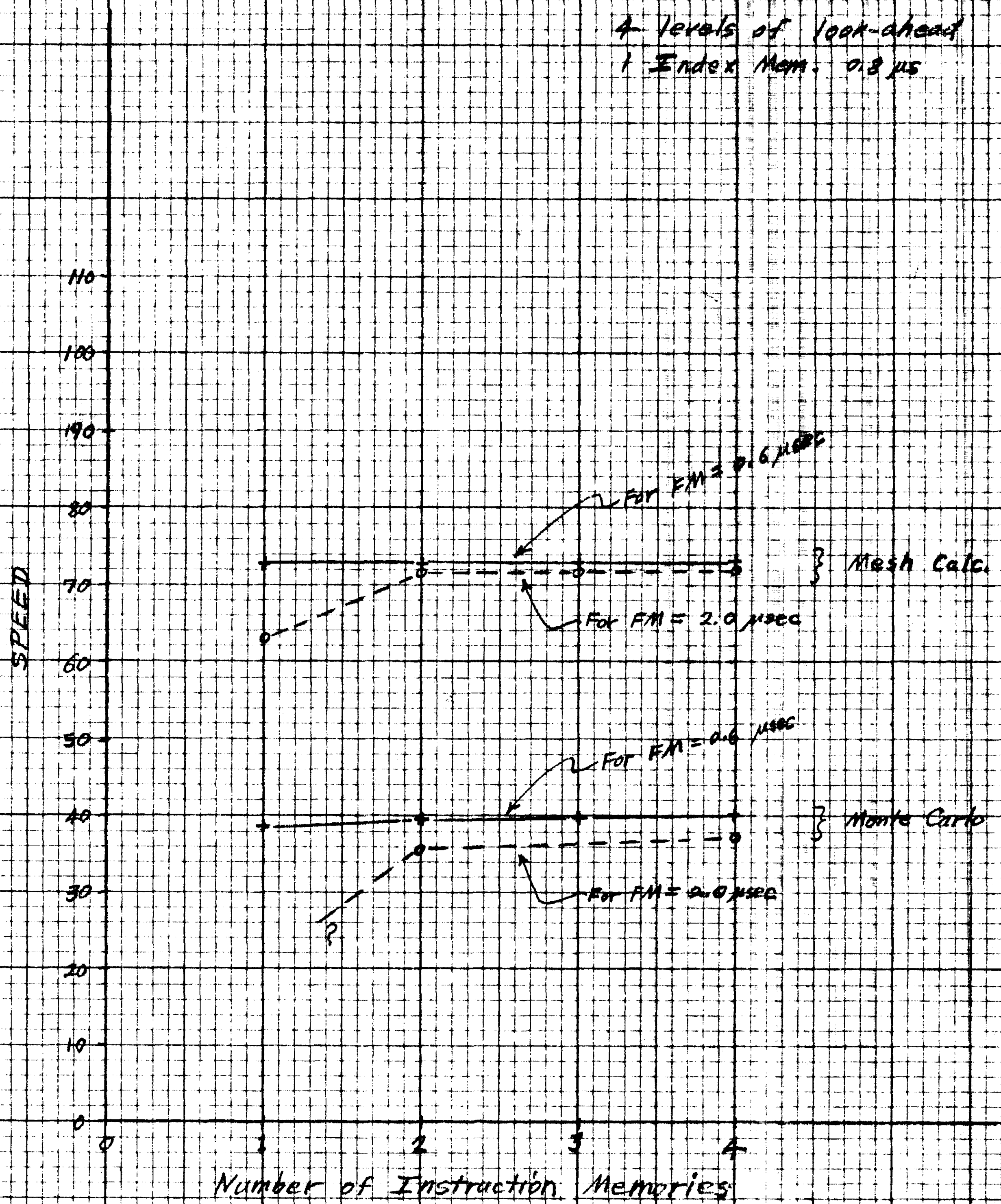


HGK

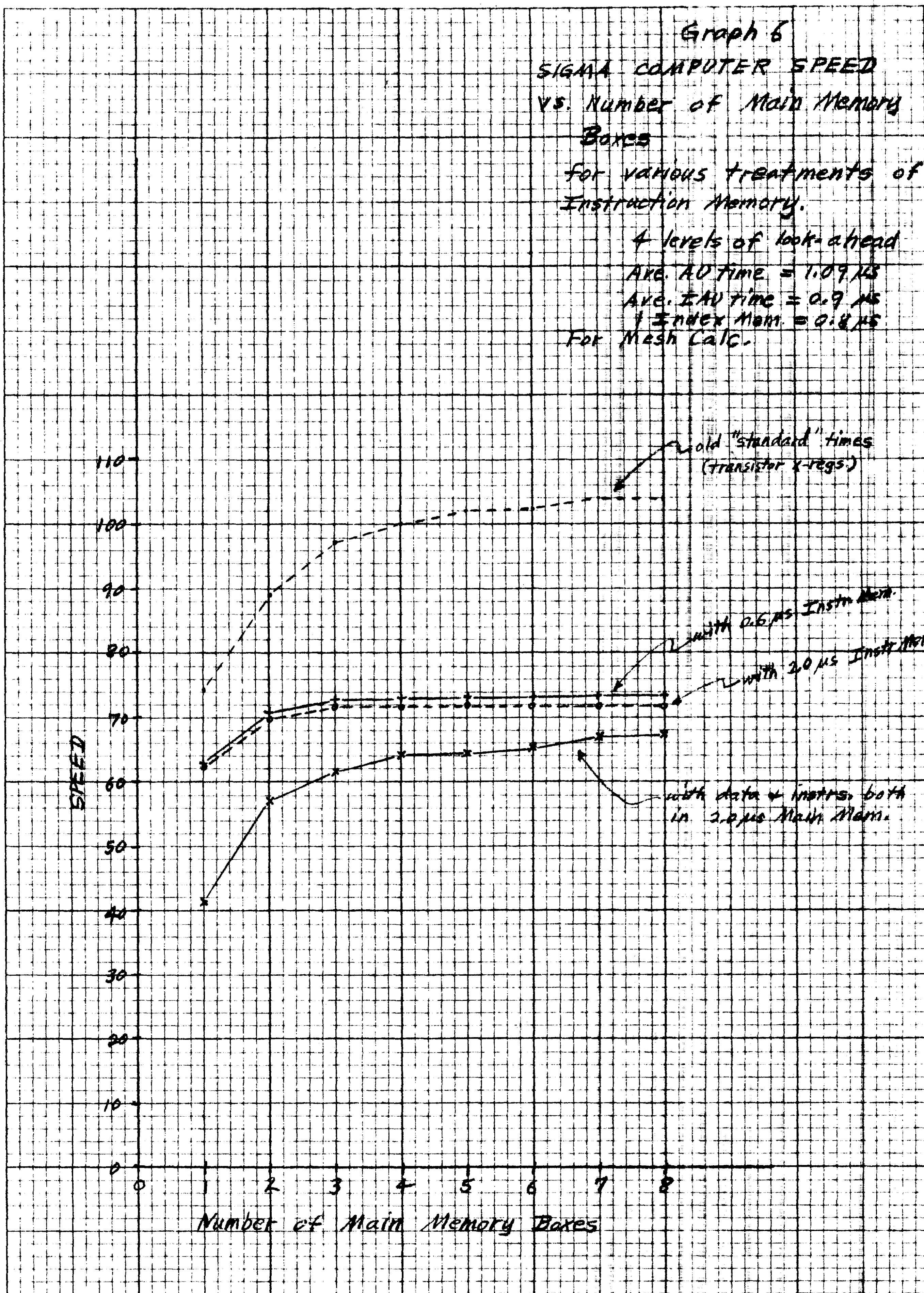
Graph 5

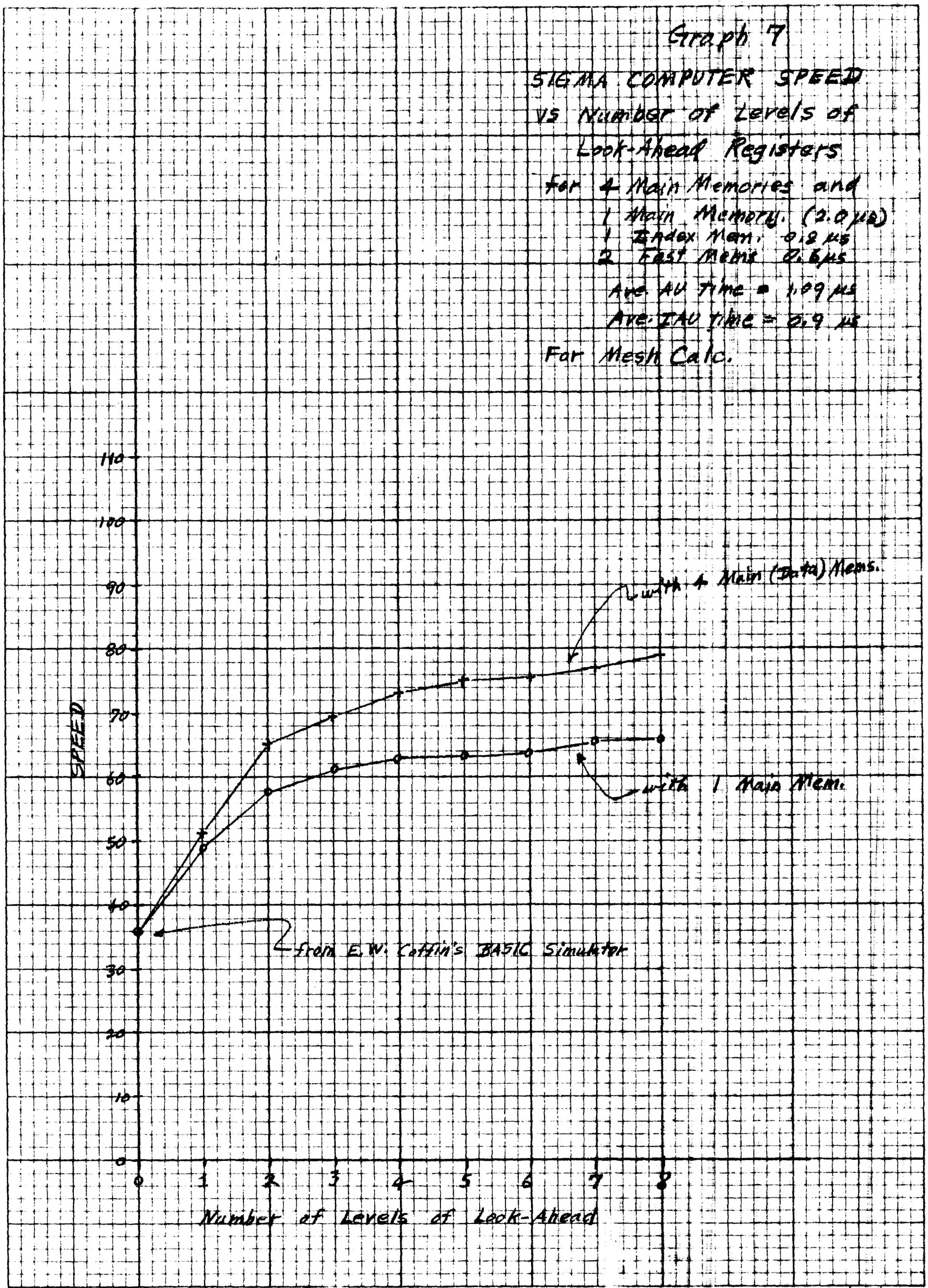
SIGMA COMPUTER SPEED
VS Number of Instruction Memories

4 Main memories 2.0 μs
Ave. A.U. time = 1.09 μs
Ave. IAU Time = 0.9 μs
4 levels of look-ahead
1 Index Mem. 0.8 μs

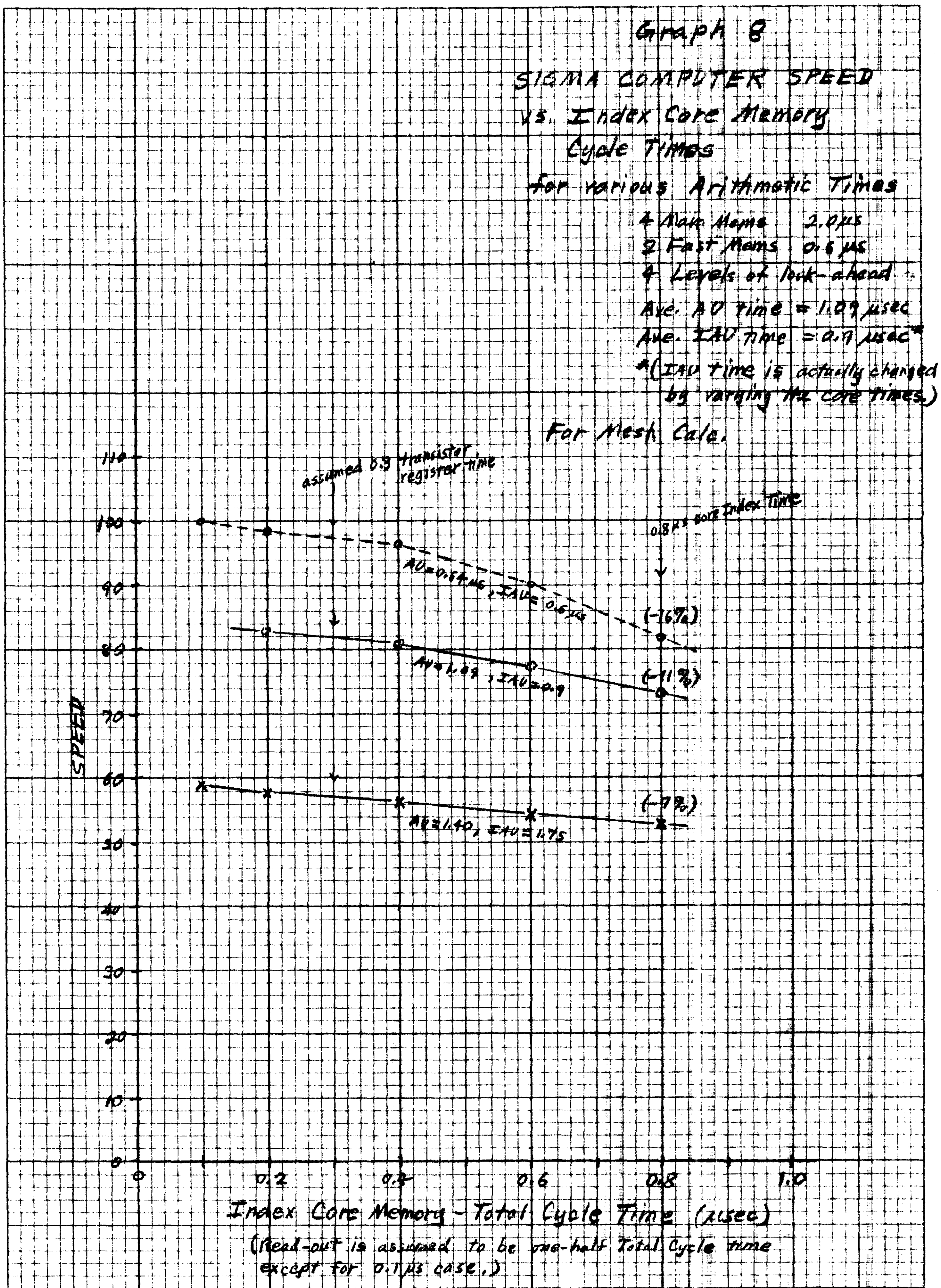


HGK


Graph 6
SIGMA COMPUTER SPEED
vs. Number of Main Memory Boxes
for various treatments of Instruction Memory.
4 levels of look-ahead
Ave. AU time = 1.09 μs
Ave. IAU time = 0.9 μs
+ Index Mem. = 0.4 μs
For Mesh Calc.
old "standard" times (transistor X-regs.)
with 0.6 μs Instr. Mem.
with 2.0 μs Instr. Mem.
with data + instrs. both in 2.0 μs Main Mem.
SPEED
110
100
90
80
70
60
50
40
30
20
10
0
0
1
2
3
4
5
6
7
8
Number of Main Memory Boxes
HGK

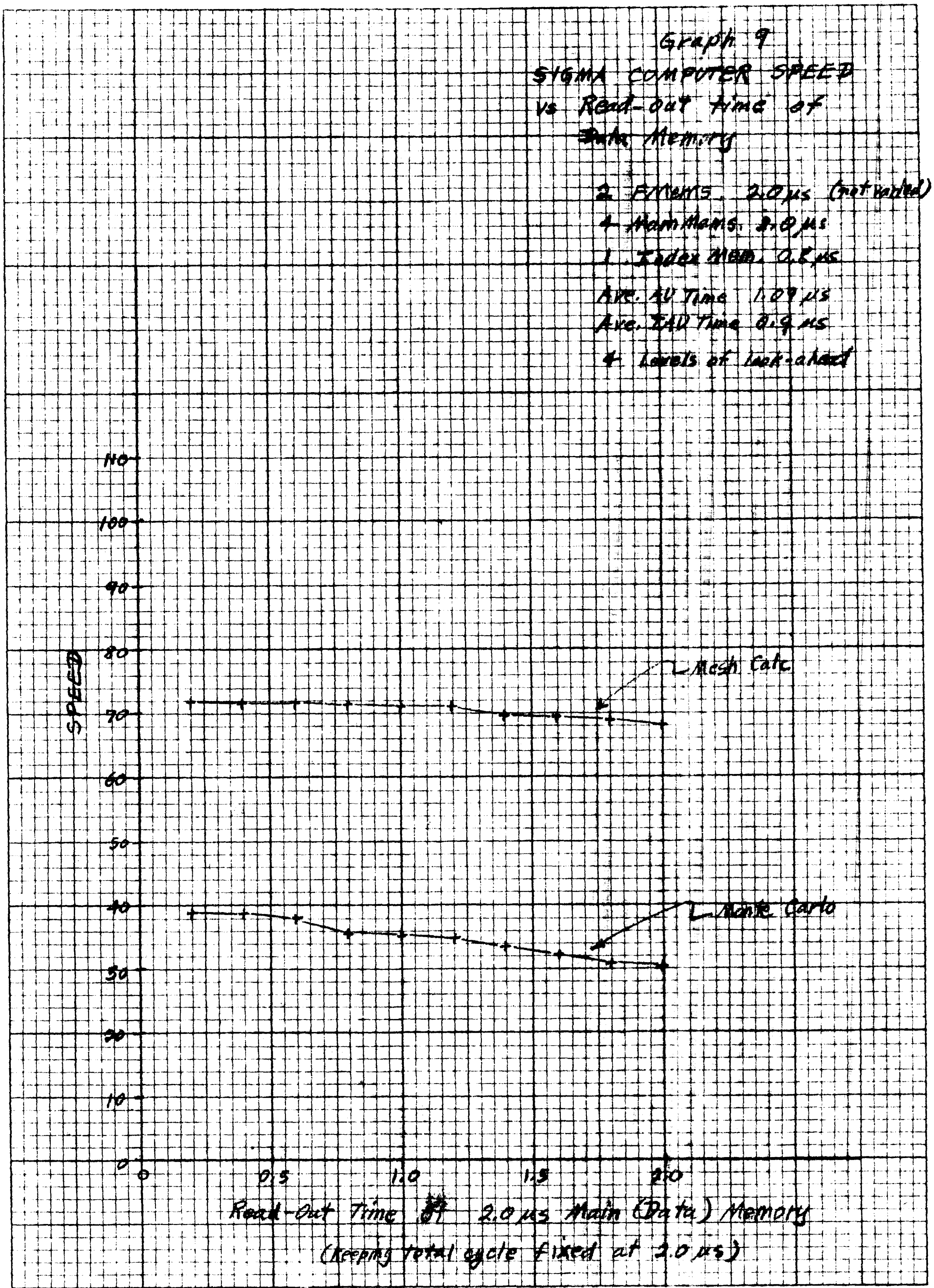

Graph 7
SIGMA COMPUTER SPEED
vs Number of Levels of
Look-Ahead Registers
for 4 Main Memories and
1 Main Memory. (2.0 μs)
1 Index Mem. 0.8 μs
2 Fast Mems 0.6 μs
Ave. AU Time = 1.09 μs
Ave. IAU Time = 0.9 μs
For Mesh Calc.
with 4 Main (Data) Mems.
with 1 Main Mem.
from E.W. Coffin's BASIC Simulator
SPEED
110
100
90
80
70
60
50
40
30
20
10
0
0
1
2
3
4
5
6
7
8
Number of Levels of Look-Ahead

HGK



Graph 8
SIGMA COMPUTER SPEED
vs. Index Core Memory
Cycle Times
for various Arithmetic Times
4 Main Mems 2.0 μs
2 Fast Mems 0.6 μs
4 Levels of look-ahead
Ave. AU time = 1.09 μsec
Ave. IAU time = 0.9 μsec*
*(IAU time is actually changed
by varying the core times.)
For Mesh Calc.
assumed 0.3 transistor register time
0.8 μs core Index Time
AU=0.54 μs, IAU=0.6 μs
(-16%)
AU=1.09, IAU=0.9
(-11%)
AU=1.40, IAU=1.75
(-7%)
SPEED
110
100
90
80
70
60
50
40
30
20
10
0
0
0.2
0.4
0.6
0.8
1.0
Index Core Memory - Total Cycle Time (μsec)
(Read-out is assumed to be one-half Total Cycle time
except for 0.1 μs case.)

HGK


Graph 9
SIGMA COMPUTER SPEED
vs Read-out time of
Data Memory
2 F Mems. 2.0 μs (not varied)
4 Main Mems. 2.0 μs
1 Index Mem. 0.5 μs
Ave. AU Time 1.09 μs
Ave. IAU Time 0.9 μs
4 Levels of look-ahead
SPEED
110
100
90
80
70
60
50
40
30
20
10
0
Mesh Calc
Monte Carlo
0
0.5
1.0
1.5
2.0
Read-Out Time of 2.0 μs Main (Data) Memory
(keeping total cycle fixed at 2.0 μs)

HGK