J. Cocke

SIGMA COMPUTER MEMO #3

SUBJECT: Evaluation of Sigma System Performance.

BY: R. J. Bahnsen and J. F. Dirac

DATE: January 14, 1958

In designing a Lookahead System to be used with the Sigma Arithmetic Unit, it is necessary to determine the number of buffer registers (or lookahead levels) that are required to keep the system running effectively. The following analysis assumes the operation of the Lookahead to be essentially that described in Sigma Memo #2.

Two different cases, case 1 - the arithmetic unit is faster than the indexing unit and case 2 - the indexing unit is faster than the arithmetic unit, are considered. The optimum condition where the average arithmetic execution time equals the average indexing time occurrs at the boundary of the two cases.

Definition of Terms

1. N - The number of lookahead buffer registers.
2. T_I - Indexing time - includes - the decoding of the instruction, selection of the proper index register and procurement of the index value, index addition to obtain the effective address, and loading the operation and address into the Lookahead buffer with a comparison indication, if any.
3. T_A - Arithmetic Unit time.
4. T_{Amax} - Time for the longest arithmetic operation.
5. $\overline{T}_{Amin}$ - Minimum average time for a series of N - arithmetic operations. (Any symbol with a bar over it indicates an average time.)
6. T_{DD} - Time required for data requested from memory to be available for processing. This time includes memory request time, memory access time and checking time.

SIGMA COMPUTER
MEMO #3

January 14, 1958

7. T_{DI} - Time required for instructions requested from memory to be available for processing. This time includes the same factors as T_{DD} plus address comparison time.

Case 1

Average Arithmetic Execution Time less than or equal to Indexing Time.

$$\overline{T}_A \leq \overline{T}_I$$

In this case the Indexing Unit limits the speed of the system. For the system to operate as fast as possible, the Indexing Unit should operate continuously. (It should never have to wait for the Arithmetic Unit to finish an operation.) To meet this condition it is required that enough buffers be provided in the Lookahead so that during the longest arithmetic time and the time for acquiring the data for this operation there be buffers available for the reception of indexed instructions. This is illustrated in figure 1.

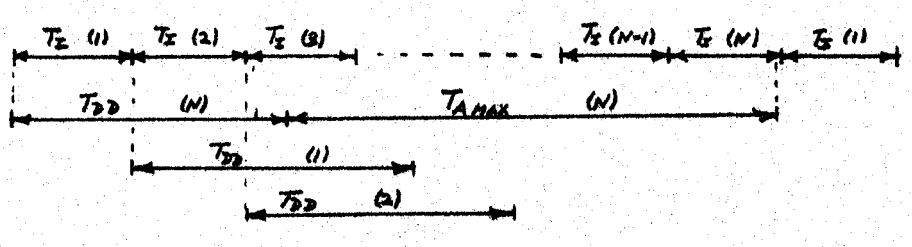

Fig. 1

In the figure, it is assumed that the arithmetic unit is immediately behind the indexing unit at the time that the longest arithmetic operation occurs. This implies that the long operation will occur infrequently enough so that the arithmetic unit will catch up to the indexing unit. From figure 1 we can say that the condition for keeping the Indexing unit running continuously is:

(1) $$T_{DD} + T_{A_{MAX}} \leq N \overline{T}_I$$

If the worst case arithmetic time is ignored and an average arithmetic time is used, we get:

(2) $$T_{DD} + \overline{T}_A \leq N \overline{T}_I$$

SIGMA COMPUTER
MEMO #3

January 15, 1958

Solving for N in equation (1) and (2) we have:

(3) $$N \geq \frac{T_{DD} + T_{A_{MAX}}}{\overline{T_I}}$$

(4) $$N \geq \frac{T_{DD} + \overline{T_A}}{\overline{T_I}}$$

A practical design figure for N should be somewhere between the limits given by eqs. (3) & (4). It can be seen that the increase in N due to consideration of the worst case arithmetic unit time is:

(5) $$\Delta N = \frac{T_{A_{MAX}} - \overline{T_A}}{\overline{T_I}}$$

Case 2

Average Indexing Time less than or equal to Average Arithmetic Time.

$$\overline{T_I} \leq \overline{T_A}$$

Since the Arithmetic Unit limits the speed of the system in this case, it is desirable to keep the Arithmetic Unit operating continuously. This means that enough buffers should be provided so that a series of short instructions will not cause the arithmetic unit to have to wait for instructions or data. Figure 2 represents the condition where the indexing unit was as far ahead of the arithmetic unit as the depth of the lookahead would allow when the series of short instructions occurred. It is assumed that the next instruction after these N - short instructions has an arithmetic time which is longer than average. From figure 2 it can be seen that the following condition should be met to

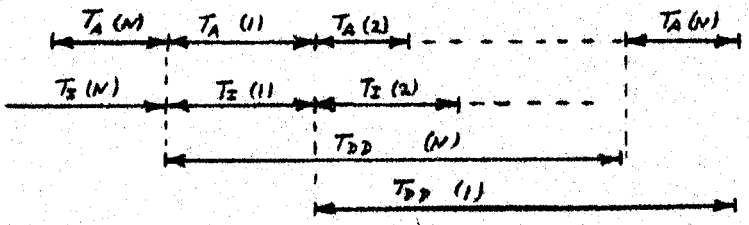

Fig. 2

keep the arithmetic unit running steadily.

(6) $$T_{DD} \leq (N-1)\,\overline{T_{A_{MIN}}}$$

An average minimum arithmetic time is used since it is unlikely that a series

SIGMA COMPUTER
MEMO #3

January 15, 1958

of (N -) instructions would all be of the minimum length (such as reset add or load). Equation (6) solved for N explicitly gives:

$$(7) \qquad N \geq \frac{T_{DD}}{\overline{T}_{A\,MIN}} + 1$$

If as in Case (1) we consider only average times, we get the following for N:

$$(8) \qquad N \geq \frac{T_{DD}}{\overline{T}_{A}} + 1$$

Equation (7) can be written as

$$(9) \qquad (N-1) \geq \left(\frac{T_{DD}}{\overline{T}_{A}}\right)\left(\frac{\overline{T}_{A}}{\overline{T}_{A\,MIN}}\right)$$

From this it can be seen that the increase in N necessary to take into account the possible deviation from the average arithmetic speed is:

$$(10) \qquad \Delta N = \frac{T_{DD}}{\overline{T}_{A}}\left(\frac{\overline{T}_{A}}{\overline{T}_{A\,MIN}} - 1\right)$$

The Optimum or Balanced System

$$\overline{T}_{A} = \overline{T}_{I} = T$$

The most desirable system is one in which the average arithmetic time is equal to the indexing time. If this is true, either equation (4) from Case (1) or equation (8) from Case (2) will indicate the number of lookahead buffers needed to take care of average times. If $\overline{T}_A$ and $\overline{T}_I$ are replaced by T in these equations, we get:

$$(11) \qquad N \geq \frac{T_{DD}}{T} + 1$$

This is the requirement for buffering out memory reference time and checking time.

Overlapping the Procurement of Instructions with Indexing

If a memory reference for instructions is overlapped with indexing, the following can be stated. There is no advantage in reducing the indexing time to less than the time required to get the instruction from memory and check it. If the

indexing time is less than this, the indexing unit will have to wait before it can operate on the next instruction. In the interest of economy we can say:

$$T_I \geq \frac{T_{DI}}{2}$$

T_{DI} is divided by 2 since half word instructions are used for Sigma.

Examples

To illustrate the concepts outlined previously, consider the two examples indicated below. In both examples, it will be assumed that an optimum design of $\overline{TA} = \overline{TI}$ exists. Typical values for T_{DD} and TDI are figured as follows:

	T_{DD}		T_{DI}	
	2.0M	0.5M	2.0M	0.5M
Request time	0.20	0.20	0.60	0.60
Get Slot	0.25	0.25	0.25	0.25
50' Cable	0.075	0.075	0.075	0.075
Read Time	0.80	0.40	0.80	0.40
Gate Out	0.20	0.20	0.20	0.20
50' Cable	0.075	0.075	0.075	0.075
Set Register	0.05	0.05	0.05	0.05
Checking	0.60	0.60	0.6	0.6
	2.25	1.85	2.65	2.25

Example No. 1

(a) $\overline{T}_{DD} = 2.25$ microseconds, $\overline{T}_{DI} = 2.65$ microseconds
$\overline{T}_A$ is determined as follows:

(1) 6 operations at .5 each = 3.0 us
(2) 6 operations at 1.0 each = 6.0 us
(3) 3 operations at 2.5 each = 7.5 us
(4) 1 operation at 7.5 each = 7.5 us

$\overline{T}_A = 1.5$ us $= \frac{24.0 \text{ us}}{16 \text{ ops}}$

from (1) and (2) $\overline{T_A}$ minimum = .75 us
and $\overline{T_A}$ maximum = 7.5 us.
To maintain $\overline{T_A}$ then $\overline{T_I} \leq 1.5$ us. and
execution must be continueusly processing.

(b) Case 2 implies by equation (7)

$$N = \frac{2.25}{.75} + 1 = 4 \quad \text{levels}$$

(c) If $\overline{T_I} = \overline{T_A}$ exactly, then we must be able to continue indexing on a long execution or Case (1) implies by equation (3)

$$N \geq \frac{2.25 + 7.5}{1.5} = 6.5$$

$$N = 7 \quad \text{levels}$$

where

$$\Delta N = \frac{7.5 - 1.5}{1.5} = 4 \qquad (5)$$

is due to the long execution example.

(d) For purely average operations the depth of lookahead would be:

$$N \geq \frac{2.25 + 1.5}{1.5} = 2.5$$

$$N = 3 \quad \text{levels}$$

(e) Notice that $\overline{T_I}$ $\overline{\frac{T_{DI}}{2}}$ and

a simple buffer overlap scheme satisfies the rate of instruction procurement.

Example No. 2

(a) $\overline{T_{DD}}$ = 2.2 microseconds

SIGMA COMPUTER
MEMO #3 January 15, 1958

$\overline{T_{DI}}$ = 1.3 microseconds

$\overline{T_A}$ is determined as follows

(1) 6 operations at .2 each = 1.2 us
(2) 6 operations at .6 each = 3.6 us
(3) 3 operations at 1.2 each = 3.6 us
(4) 1 operation at 1.8 each = 1.8

$$\overline{T_A} = .63 \text{ us} = \frac{10.2}{16}$$

from (1) and (2) $\overline{T_A}$ minimum = .4 us
and $\overline{T_A}$ maximum = 1.8 us.
To maintain $\overline{T_A}$ then $\overline{T_I} = .63$ us.
and execution must be continuously processing.

(b) Case 2 implies by equation (7)

$$N \geq \frac{2.25}{.4} + 1 = 6.6$$

$$N = 7 \quad \text{levels}$$

(c) If $\overline{T_I} = \overline{T_A}$ exactly then we must be able to continue indexing on a long execution or case (1) implies by equation (3)

$$N \geq \frac{2.25 + 1.8}{.63} = 6.42$$

$$N = 7 \quad \text{levels}$$

The large number of levels is dictated by the now comparatively slow memory access time since:

(d) For purely average operations the depth of lookahead would be:

$$N \geq \frac{2.25 + .63}{.63} = 4.58$$

$$N = 5$$

(e) Notice that a much faster memory access is required so that $\overline{T_I} \geq \frac{\overline{T_{DI}}}{2}$ and a simple buffer overlap scheme will satisfy the rate of instruction procurement.

SIGMA COMPUTER
MEMO #3

January 15, 1958

R J Bahnsen

R. J. Bahnsen

J. F. Dirac

J. F. Dirac

RJB:JFD:bjn