J. E. Griffith

Poughkeepsie, New York
December 13, 1955

PROJECT STRETCH
FILE MEMO #14

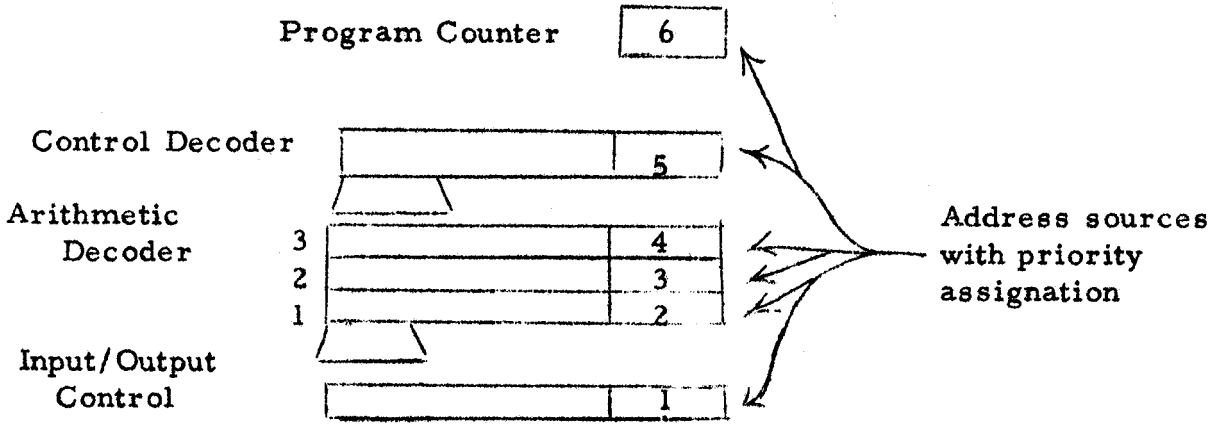
COMPANY CONFIDENTIAL

SUBJECT: Logical Equations for ANS Decoder
By: G. M. Amdahl

The following will describe a system which determines the sequence of memory references called for by the address sources within the system. This system is a simple version, in that the interchange of memory references between STORE and LOAD type instructions is not included and that address comparison for equality is also ignored. (Both of these ignored features are treated separately and are required for maximum speed.)

In this memo, the I/O system is assumed to have its separate priority system, and consequently acts here as a single unit.

The system described is based on the following block diagram machine (see memo on "Logical Rules for Sequencing Memory References" by G. M. Amdahl, E. M. Boehm and J. E. Griffith of 11/24/55).



These address sources are assumed to be controlling an arithmetic unit, a control arithmetic unit, a program sequence unit, an input/output unit(s), and a number of memory units of two categories, fast and slow. These units are all autonomous and asynchronous, possessing their own clock. They may be started on a task by the control section and after their operation is initiated, they will complete the task on their own. In the process of carrying out the task, these units will generate a number of signals useful for controlling sequencing of memory references. Below is a list of such signals by unit:

Arithmetic Unit Signals

AR — The time left before completion of operation is short enough to start a READ in slow memory.

AW — The time left before completion of operation is short enough to start a WRITE in slow memory or a READ or WRITE in fast memory.

AT — The arithmetic unit is ready to transfer a word between one of its registers and a memory register.

Control Arithmetic Unit Signals

CAT — The control arithmetic unit is now ready to transfer a word between one of its registers and a memory register.

Input/Output Unit Signals

I/O T — The input/output unit is now ready to transfer a word between one of its registers and a memory register.

Program Unit Signals

PT — The control decoder is now ready to receive a word from a memory register.

Memory Unit Signals

MA_j — Memory unit j is available for capturing a memory cycle.

MT_j — Memory unit j is ready to transfer a word between its memory register and any other machine register.

Each of the address sources must also generate signals (maintained in triggers) to be used in the sequencing of memory references. These signals are listed below:

R — A memory READ cycle is the kind of cycle called for (implied for program counter).

W	A memory WRITE cycle is the kind of cycled called for (not applicable for program counter).
C	A memory cycle is yet to be captured.
D	A word is yet to be transferred.
S	A "short" arithmetic operation is called for in this instruction (applicable only to arithmetic decoders).
#j	Memory box j is called for (one such signal is required for each memory box).

The logical equations used to control the sequencing at each of the address sources is now listed:

I/O Memory References (I/O)

$\#j_{I/O} \cdot MA_j \cdot C_{I/O} = Q_{I/O_j}$ (definition)

$Q_{I/O_j} \supset$ Send address from I/O to memory j and turn off $C_{I/O}$.

$Q_{I/O_j} \cdot R_{I/O} \supset$ Start READ cycle in memory j.

$Q_{I/O_j} \cdot W_{I/O} \supset$ Start WRITE cycle in memory j.

$\#j \cdot D_{I/O} \cdot MT_j \cdot I/OT = P_{I/O_j}$ (definition)

$\sum_j P_{I/O_j} \supset$ Turn off $R_{I/O}$, $W_{I/O}$, $\#j_{I/O}$ and $D_{I/O}$ (logical summation).

$P_{I/O_j} \cdot R_{I/O} \supset$ Transfer word from memory register j to I/O register.

$P_{I/O_j} \cdot W_{I/O} \supset$ Transfer word from I/O register to memory register j.

Arithmetic Decoder 1 (AD1)

$\#j_{AD1} \cdot MA_j \cdot C_{AD1} \cdot (AW + AR \cdot R_{AD1}) = Q_{AD1_j}$ (definition)

$Q_{AD1_j} \cdot \overline{Q}_{I/O_j} \supset$ Send address to memory j and turn off C_{AD1}.

$Q_{AD1j} \cdot \overline{Q}_{I/Oj} \cdot R_{AD1} \supset$ Start READ cycle in memory j.

$Q_{AD1j} \cdot \overline{Q}_{I/Oj} \cdot W_{AD1} \supset$ Start WRITE cycle in memory j.

$\#j_{AD1} \cdot D_{AD1} \cdot MT_j \cdot AT = P_{AD1j}$ (definition)

$\sum_j P_{AD1j} \supset$ Turn off R_{AD1}, W_{AD1}, $\#j_{AD1}$, and D_{AD1}.

$P_{AD1j} \cdot R_{AD1} \supset$ Transfer word from memory j to arithmetic register.

$P_{AD1j} \cdot W_{AD1} \supset$ Transfer word from arithmetic register to memory j.

Arithmetic Decoder 2 (AD2)

$\#j_{AD2} \cdot MA_j \cdot C_{AD2} \cdot AR \cdot S_{AD1} = Q_{AD2j}$ (definition)

(here j index runs only through slow memories)

$Q_{AD2j} \cdot \overline{Q_{AD1j}} \cdot \overline{Q_{I/Oj}} \cdot R_{AD2} \supset$ Send address to memory j, turn off C_{AD2} and start memory READ cycle.

Arithmetic Decoder 3 (AD3)

$\#j_{AD3} \cdot MA_j \cdot C_{AD3} \cdot AR \cdot S_{AD1} \cdot S_{AD2} = Q_{AD3j}$ (definition)

(here j index runs only through slow memories)

$Q_{AD3j} \cdot \overline{Q}_{AD2j} \cdot \overline{Q}_{AD1j} \cdot \overline{Q}_{I/Oj} \cdot R_{AD3} \supset$ Send address to memory j, turn off C_{AD3} and start memory READ cycle.

Control Decoder (CD)

$\#j_{CD} \cdot MA_j \cdot C_{CD} = Q_{CDj}$ (definition)

$Q_{CDj} \cdot \overline{Q}_{AD3j} \cdot \overline{Q}_{AD2j} \cdot \overline{Q}_{AD1j} \cdot \overline{Q}_{I/Oj} \supset$ Send address to memory j and turn off C_{CD}.

$Q_{CDj} \cdot \overline{Q}_{AD3j} \cdot \overline{Q}_{AD2j} \cdot \overline{Q}_{AD1j} \cdot \overline{Q}_{I/Oj} \cdot R_{CD} \supset$ Start READ cycle in memory j.

$Q_{CDj} \cdot \bar{Q}_{AD3j} \cdot \bar{Q}_{AD2j} \cdot \bar{Q}_{AD1j} \cdot \bar{Q}_{I/Oj} \cdot W_{CD} \supset$ Start WRITE cycle in memory j.

$\#j_{CD} \cdot D_{CD} \cdot MT_j \cdot CAT = P_{CDj}$ (definition)

$\sum_j P_{CDj} \supset$ Turn off R_{CD}, W_{CD}, $\#jCD$ and D_{CD}.

$P_{CDj} \cdot R_{CD} \supset$ Transfer word from memory j to control arithmetic register.

$P_{CDj} \cdot W_{CD} \supset$ Transfer word from control arithmetic register to memory j.

Program Counter (PC)

$\#j_{PC} \cdot MA_j \cdot C_{PC} = Q_{PCj}$ (definition)

$Q_{PCj} \cdot \bar{Q}_{CDj} \cdot \bar{Q}_{AD3j} \cdot \bar{Q}_{AD2j} \cdot \bar{Q}_{AD1j} \cdot \bar{Q}_{I/Oj} \supset$ Send address to memory j, turn off C_{CD} and start memory READ cycle.

$\#j_{PC} \cdot D_{PC} \cdot MT_j \cdot PT = P_{CDj}$ (definition)

$\sum_j P_{CDj} \supset$ Turn off $\#j_{PC}$ and D_{PC}.

$P_{CDj} \supset$ Transfer word from memory j to control decoder.

G. M. Amdahl