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

MAj   Memory unit j is available for capturing a memory cycle.

MTj   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).
A memory WRITE cycle is the kind of cycled called for (not applicable for program counter).

A memory cycle is yet to be captured.

A word is yet to be transferred.

A "short" arithmetic operation is called for in this instruction (applicable only to arithmetic decoders).

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)**

\[ #_{I/O} \cdot M_{I/O} = Q_{I/O} \quad \text{(definition)} \]

- Send address from I/O to memory j and turn off \( C_{I/O} \).

- Start READ cycle in memory j.

- Start WRITE cycle in memory j.

\[ #_{I/O} \cdot M_{I/O} = \overline{Q_{I/O}} \quad \text{(definition)} \]

\[ \sum_{j} P_{I/O} \quad \text{Turn off } R_{I/O}, W_{I/O}, #_{I/O} \text{ and } D_{I/O} \quad \text{(logical summation).} \]

- Transfer word from memory register j to I/O register.

- Transfer word from I/O register to memory register j.

**Arithmetic Decoder 1 (AD1)**

\[ #_{AD1} \cdot M_{AD1} = Q_{AD1} \quad \text{(definition)} \]

- Send address to memory j and turn off \( C_{AD1} \).
Q_{AD1j} \cdot \overline{Q_{I/Oj}} \cdot R_{AD1} \Rightarrow \text{Start READ cycle in memory } j.

Q_{AD1j} \cdot \overline{Q_{I/Oj}} \cdot W_{AD1} \Rightarrow \text{Start WRITE cycle in memory } j.

\#j_{AD1} \cdot D_{AD1} \cdot M_{Tj} \cdot AT = P_{AD1j} \quad \text{(definition)}

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

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

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

\text{Arithmetic Decoder 2 (AD2)}

\#j_{AD2} \cdot M_{Aj} \cdot C_{AD2} \cdot AR \cdot S_{AD1} = Q_{AD2j} \quad \text{(definition)}

\text{(here } j \text{ index runs only through slow memories)}

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

\text{Arithmetic Decoder 3 (AD3)}

\#j_{AD3} \cdot M_{Aj} \cdot C_{AD3} \cdot AR \cdot S_{AD1} \cdot S_{AD2} = Q_{AD3j} \quad \text{(definition)}

\text{(here } j \text{ index runs only through slow memories)}

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

\text{Control Decoder (CD)}

\#j_{CD} \cdot M_{Aj} \cdot C_{CD} = Q_{CDj} \quad \text{(definition)}

Q_{CDj} \cdot \overline{Q_{AD3j}} \cdot \overline{Q_{AD2j}} \cdot \overline{Q_{AD1j}} \cdot \overline{Q_{I/Oj}} \Rightarrow \text{Send address to memory } j \text{ 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} \Rightarrow \text{Start READ cycle in memory } j.
PCDj*RCD
-2
Transfer word from memory j to control arithmetic register.

PCDj*WCD
-2
Transfer word from control arithmetic register to memory j.

Program Counter (PC)

#jPC*MAj*CPC = QPCj

QPCj*PCDj*QAD3j*QAD2j*QAD1j*QI/Oj
-2
Send address to memory j, turn off CCD and start memory READ cycle.

#jPC*DP*CjMPT = PCDj

PCDj
-2
Transfer word from memory j to control decoder.

G. M. Amdahl