DOCUMENT RESUME

ED 124 152 IR 003 540

AUTHOR Tucker, Paul Thomas
TITLE Auxiliary Mass Storage System; (A New Control Structure for Serially Organized Memories in an Interactive Computer System).
INSTITUTION Illinois Univ., Urbana. Computer-Based Education Lab.
SPONS AGENCY Advanced Research Projects Agency (DOD), Washington, D.C.; National Science Foundation, Washington, D.C.
REPORT NO CERL-R-X-45
PUB DATE Oct 75
CONTRACT US Army/DAHC-15-73-C-0077
GRANT USNSF-C-723
NOTE 127p.; Not available in hard copy from ERIC
AVAILABLE FROM PLATO publications, Computer-based Education Research Lab., 252 Engineering Research Laboratory, University of Illinois, Urbana, Illinois 61801 ($4.80, prepayment required)

EDRS PRICE MF-$0.83 Plus Postage. HC Not Available from EDRS.
DESCRIPTORS *Computer Assisted Instruction; *Computer Programs; *Computer Storage Devices; Engineering Technology; Higher Education; Information Storage; Program Descriptions
IDENTIFIERS PLATO; Programmed Logic for Automatic Teaching Operations

ABSTRACT

Interactive computers are providing more services to greater numbers of users. However, before such systems can be economically applied to public education, the cost of high performance memory must be reduced. By using massive amounts of serial memory in conjunction with a highly sophisticated memory controller, a system was constructed which performs as well as random access memory, but at less cost. To demonstrate the control structure, and to determine the applicability of the memory system to PLATO computer-based educational system, a memory controller was installed and operated, a small memory system was introduced, and software was written to slave the memory controller system to the operation of the PLATO system. (EMH)

CERL REPORT X-45

AUXILIARY MASS STORAGE SYSTEM

(A New Control Structure for Serially Organized Memories in an Interactive Computer System)

by

Paul Thomas Tucker

Computer-based Education Research Laboratory
University of Illinois
Urbana, Illinois
October 1975

This manuscript was prepared with partial support from the National Science Foundation (USNSF C-723) and the Advanced Research Projects Agency (US Army/ DAHC 15-73-C-0077).

ACKNOWLEDGEMENTS

Sincere and grateful appreciation is extended to my advisor Professor Donald L. Bitzer for his continuous support, inspiration, and interest. In addition, special appreciation is extended to Bruce Sherwood for his willingness to give much of his time and efforts in assisting in the preparation of this report. In addition, without the assistance of numerous other individuals, this entire project could not have been completed. They include David Andersen, Larry Crotzer, Donald Hartman, Leonard Hedges, Fred Holy, James Parry, Robert Rader, and Jack Stifle. Special appreciation is given to David Andersen, Larry Crotzer, and Leonard Hedges. David Andersen was instrumental in the development of the necessary software for the memory drivers, Larry Crotzer was very helpful in assisting in interpreting the Control Data documentation and assisting in trouble-shooting the ECS interface, and Leonard Hedges was instrumental in the assembly, modification, and testing of the hardware. Roy Lipschutz and Wayne Wilson completed the necessary drawings. Linda Gardner assisted in the typing and assembly of materials. Pauline Jordan was very helpful in editing the manuscript. Elaine Avner prepared the final draft for publication and Sheila Knisley typed the final manuscript.

In 1971, Donald Bitzer began to formulate a serial memory structure and James Parry formulated a serial central memory structure. Discussions between the two individuals and later, this author and others, resulted in the basic foundation for the first phase of the AMS memory system.

ABSTRACT

Interactive computer systems are today becoming very widely accepted and are supplying ever increasing services to larger and larger numbers of users. However, one of the major remaining problems to be overcome before such systems can be economically applied to such fields as public education is the need for massive amounts of high performance yet very inexpensive memory. This report will present one approach to the solution of this problem by demonstrating the feasibility of using massive amounts of serial memory which will eventually be less expensive than random access memory. The use of a highly sophisticated memory controller will cause this serial memory to perform at a similar level to that of random access memory in this application.

This memory control approach will demonstrate the capability of a serial memory having characteristic latency times of hundreds of μsecs (microseconds) to appear to the controlling computer to have average access times of tens of μsecs.

To demonstrate this control structure, and to determine the applicability of this memory system to the PLATO computer-based education system, a memory controller was built and operated, a small memory system was built and operated, and software was written to slave the memory controller system to operation of the PLATO system.

TABLE OF CONTENTS

Chapter Page

CHAPTER 1

INTRODUCTION

The purpose of this project was to investigate a new approach to the problem of providing massive amounts of low-cost computer memory in an interactive computer environment which normally requires large amounts of high-speed, high-cost random access memory. This alternative approach uses serial memory with a very high transfer rate in place of fast random access memory. The name of this memory is the Auxiliary Mass Storage (AMS) system. Although several types of memory devices could be used for this experiment, commercially available semiconductor serial shift-register memory is utilized. Even though access to specific location in a serial memory can require significant amounts of time, the use of inter-leaving and parallel memory control paths can provide a significant reduction of the access times and permit very high transfer rates.

The interactive system used in this memory experiment is the PLATO IV computer-based education [1,2] system located on the campus of the University of Illinois, Urbana, in the Computer-based Education Research Laboratory.

The PLATO system must process a response in 0.1 second for each keypress even though each keypress is treated as an individual job for the Central Processing Unit (CPU). This performance requirement dictates that all of the necessary data and program material be swapped from high performance memory instead of from disks or drums [3].

An elaborate memory hierarchy is necessary for the PLATO system to function properly, starting with a high-speed central memory of only 65k words, continuing to a large mass swapping memory of two million words for the storage of working data, and finally to massive disk memory for the storage of library materials [3]. The mass swapping memory, Control Data Extended Core Storage (ECS) [4], represents by far the largest fraction of

of the total cost of the computer hardware and is the point in the memory structure toward which the work detailed in this paper is directed.

Other computer systems use other forms of mass storage where PLATO uses ECS but these forms are unsatisfactory for a highly interactive system such as PLATO. A memory medium suitable for this position in the memory hierarchy requires an average access time in the tens of μsecs and a transfer rate in the hundreds of millions of bits per second. In contrast, magnetic disk media offer average access times of tens of thousands of μsecs and average transfer rates of only ten million bits per second.

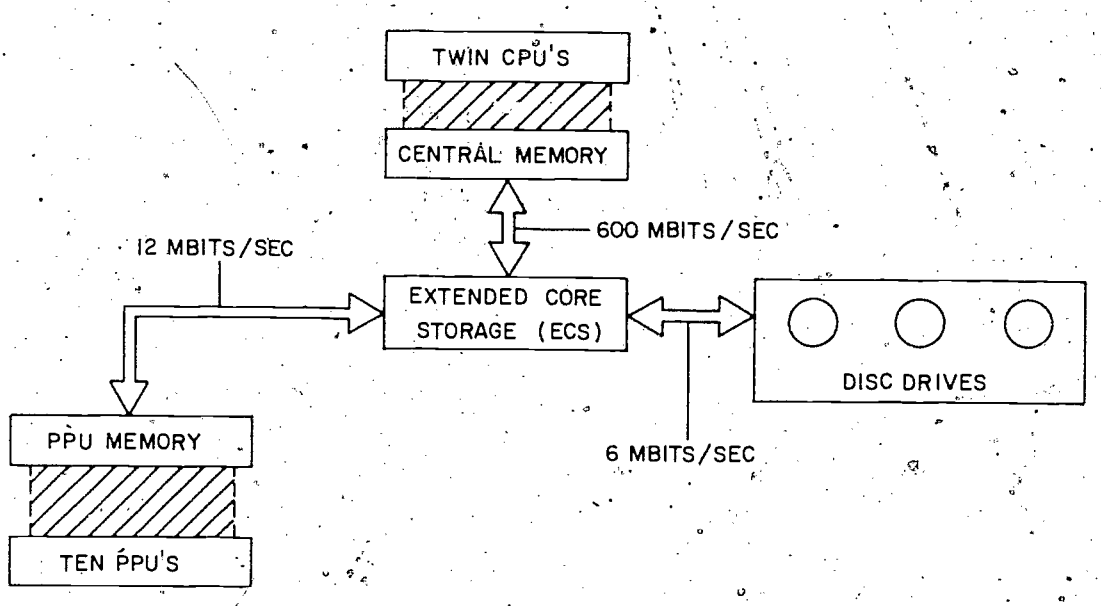
Consider an interactive system servicing 1000 simultaneous users, and assume that each user presses keys at an average rate of 0.5 key per second, with each key requiring the swapping of an average of 2000 60-bit words of information (120,000 bits) between Central Memory (CM) and swapping memory. Operating under a memory system offering 50-μsecs access and 600-megabit-per-second transfers, 500 accesses or 25,000 μsecs and 500 transfers or 100,000 μsecs for a total of 125,000 μsecs per second (one-eighth second per second) would be required just to service the data access requirements. Operating under disk parameters, 5 million μsecs would be spent for access and 6 million μsecs for transfer. This is a total of 11 million μsecs per second (eleven seconds/second!).

It is not necessary that the access time to data be zero to maximize CPU efficiency, because in a timesharing environment the CPU can execute a job for one user while the data for the next are being fetched. A typical job only runs a few msecs (milliseconds) in the CPU, whereas disk transfers plus access take several tens of msecs, so overlapping of processing and access to disk is not possible. But if a transfer takes less than a few msecs, accesses can be totally overlapped, provided the control structure is appropriate.

The basic memory module of the new memory uses a serial shift-register device which provides a 1.25 megabit-per-second transfer rate. Using 480 parallel registers, a transfer rate of 10 million 60-bit words per second is achieved, with an average access time of 400 μsecs. Through the use of memory parallelism and overlapping of memory accesses, the effective access time is reduced to about 40 μsecs and the typical performance of this new memory system meets the PLATO requirements for this level of memory.

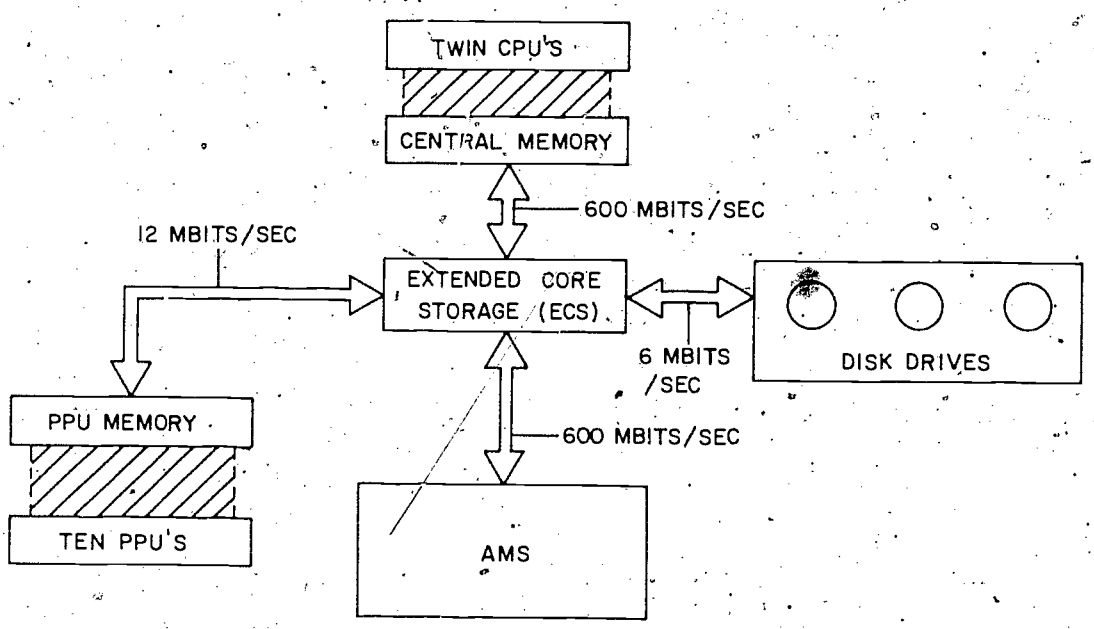
The concepts involved in the organization of the AMS memory system are directly applicable to several forms of serial memory on the technological horizon, including Charge Coupled Devices (CCD) [5,6], Magnetic Bubble Devices [7], and Electron Beam Devices [8,9]. The P-channel, Metal Oxide Semiconductor (MOS) technology was chosen for the experiment principally because of its immediate availability. The other serial technologies mentioned above promise much reduced cost and greatly increased density and performance over the current experimental version of AMS.

In the architecture of the PLATO memory hierarchy a fundamental aspect is transfer speed. The speed of a transfer and the interaction of that transfer with the operation of the other parts of the system determine the appropriate sizes of data swaps from one level of memory to another. From an economic standpoint it is least expensive to locate the bulk of all data storage in the lowest cost medium that is available, but from a system performance standpoint it is most effective to locate the bulk of the data in Central Memory, which provides the highest performance. A compromise between these two extremes is implemented in the PLATO system. Figure 1.1 is a schematic block diagram of the existing memory structure of the PLATO system without utilizing an AMS element, and Figure 1.2 shows how AMS can be included as a fundamental system component. In the diagram CPU means Central Processing



PLATO MEMORY WITHOUT AMS

Figure 1.1



PLATO MEMORY WITH AMS

Figure 1.2

Unit, and PPU means Peripheral Processing Unit. A PPU is a minicomputer which is primarily used to handle input and output.

It is important from a system standpoint that the time spent for data transfers between central memory and ECS be minimized, because during this time both of the CPUs are halted to allow all of the central memory bandwidth to be dedicated to the transfer. This procedure degrades CPU performance. Transfers between ECS and other media (disk, PPUs, or AMS) do not cause the CPU to stop, but they can cause a slow-down of ECS-CM transfers should the two types of transfers exist simultaneously. Disk and PPU transfers do not offer significant conflicts because of their low activity, but AMS transfers can potentially cause extreme degradation of the ECS-CM transfers. For this reason, a priority system was installed in the ECS and AMS controllers which forces the AMS to assume a lower priority to CM. This system still allows large amounts of data to be transferred between AMS and ECS but avoids significant system degradation.

The result of this investigation is the development of a control structure which will allow extremely low-cost serial memory to be usable and which will provide large amounts of memory with overall performance comparable to high-cost random access memory. The work has only incidentally been concerned with the actual serial memory utilized to demonstrate this control structure, as the technologies suitable for massive serial memories have not reached the state of development necessary for their implementation. The controller is, however, readily adaptable to these new serial memory media as they become available.

CHAPTER 2

A CONTROL STRUCTURE FOR SERIAL SWAPPING MEMORIES

2.1 Introduction

Computer systems attempting to service large numbers of simultaneous users are of two basic types. One type is primarily intended to provide batch job services with the addition of small amounts of interactiveness for the purpose of editing files and submitting jobs. This is the most common form of multi-user computer service. The second type is intended to provide a high degree of interactivity with its users and thereby to effectively converse with the users. This second type of system may not have the high peak performance characteristics available to an individual user that the first type offers but instead it offers all of the users a high level of interaction on a continuous basis.

In the normal timesharing environment, a user will utilize and demand interactivity while he is aditing or modifying a datafile, but he will retire to a batch job status when actual compilation and execution of the job is required. In this mode, there are two types of jobs required for the computer to adequately service any user: editing functions, such as line insertions and deletions, and batch process functions during actual execution of the batch job. The editing functions are very simple and require very small amounts of computer activity. The batch job process function requires rolling the job into CM and only occasionally rolling it out for access to the computer by other users. The important aspect of this type of service is that very little swapping of files from some external storage medium is needed for each job.

On the other hand, the more interactive form of computer system, an individual "job" in response to a keypress will be fairly small on the average,

but there are a large number of such jobs per unit time. When a CPU job is executed, the CPU needs access to all of the appropriate files. In order to insure that the CPU does not sit idle, those files must be readily available. As an example, consider a computer system used to teach genetic biology. Here the user is a student and interacts with the computer by answering questions posed by the computer and asking questions of the computer. Every time that a question is answered by the student the computer has to judge that answer in an intelligent fashion, and every time that a question is asked by the student, the computer is called upon to make an intelligent response. In the biology example, the computer might need to make more than simple matching judgments; answers should also be checked for misspelled versions of the correct answer as well. Moreover, if the student asks for a new generation of a sample insect colony, the computer would have to compute the proper characteristic ratios according to known genetic formulas. In this case a keypress would require a significant amount of CPU processing with access to the appropriate programs and to a significant amount of data. In the PLATO system, statistics indicate that fully half of the keys pressed by users are of the type that require significant CPU activity; the rest merely require key echoing.

In the PLATO system the amount of time required by the CPU to service keypresses is about evenly distributed from 0 msecs to 20 msecs, which is the limit allowed any particular user during one timeslice. The mean is 10 msecs. Under these constraints the maximum time that can be allowed for access to all of the data base that might be required to service a user is 10 msecs, because if more were required, the total access time plus transfer time would exceed the CPU process time, and the CPU would sit idle for the difference. To prevent queuing problems, it is wise to make the access plus transfer time considerably shorter than the average process time.

It is apparent that disk-based data storage is unsatisfactory for a highly interactive computer system. In disk systems, the rotational access alone averages 8-10 msecs, and transfers are slow (typically 5 megabits per second). It is also apparent that random access data storage is far more than adequate because typical access times for such units are 2-5 μsecs and data transfer rates of 600 megabits per second are available. A very large gap exists, however, between these two available media both in performance and price. The memory system that this thesis describes offers one suitable medium to occupy this gap.

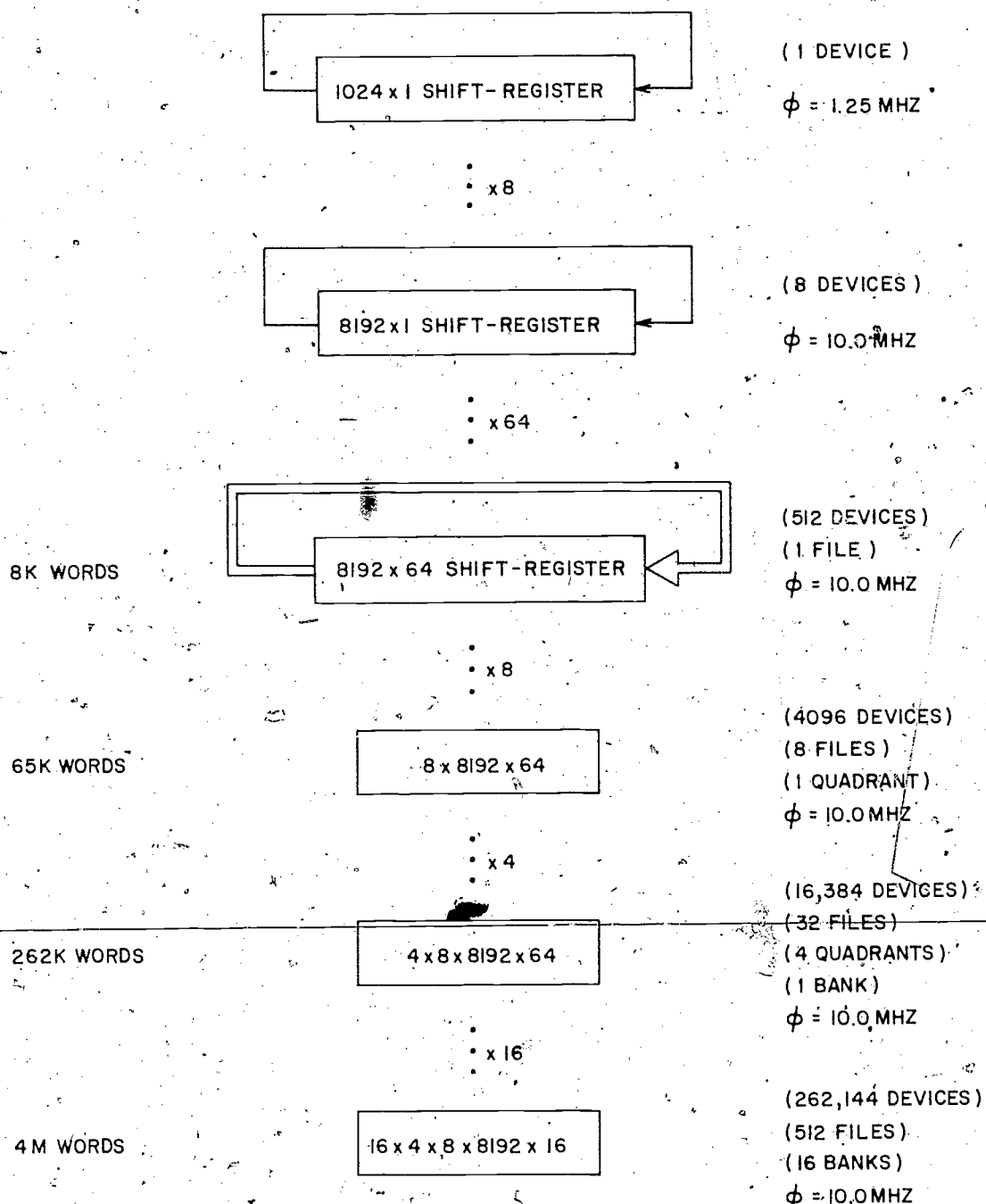
2.2 An Overview of the AMS Memory

The AMS memory system was specifically designed to satisfy the stringent performance requirements of the PLATO computer network and to provide a swapping memory very low in cost. This memory system consists basically of standard semiconductor serial shift-register devices, configured to allow a fairly sophisticated controller to manipulate them according to commands issued by the CPU. The shift-registers provide an average access time of 400 μsecs and a transfer rate of 600 megabits per second, achieved through parallelism. The use of eight independent subcontrollers causes the effective access time to approach 50 μsecs.

There are two fundamental elements in the AMS memory: the actual memory section and the memory controller.

2.2.1 The AMS Memory Storage

Figure 2.1 illustrates the organization of the AMS storage elements. The basic element utilized is a 1024 × 1 serial shift-register chosen to operate at 1.25 MHz. Groups of eight of these devices have been arranged in such a fashion that a group appears as if it were a single 8192 × 1 register operating at a maximum rate of 10 MHz. Furthermore, 64 groups have been



AMS MEMORY ORGANIZATION

Figure 2.1

paralleled to form a file unit which represents the basic memory module. The basic module, called a file, is an 8192 × 64 serial register capable of reading or writing 10 million 64-bit words per second. At these rates, and since the file is serial, the average access time to an individual storage location varies from 0 to 819.2 μsecs. The average access time is therefore 419.6 μsecs.

Any number of files can be installed to assemble a complete memory: groups of eight were chosen as the minimum size. This grouping is called a quadrant. For convenience of power supply and chassis wiring, groups of four quadrants are arranged as banks. Banks can be added until the total memory size desired is achieved. The AMS system was designed for a maximum of 16 banks.

A file is 8192 words, a quadrant is 65k words, a bank is 262k words, and a fully completed AMS system would be four megawords. A word is 64 bits.

2.2.2 The AMS Memory Controller

The heart of the AMS controller is an array of eight independent controllers. These units are called subcontrollers and have the task of manipulating individual files for the purpose of data transfer into and out of the AMS memory. In addition to the subcontrollers, there are several other units including an idle controller which maintains control over those files not under the control of a subcontroller, and a data channel through which all data and transfer parameters are communicated.

In the PLATO system, the AMS memory is connected to one of the ports of a high-speed random access swapping memory, and that memory in turn is connected to a high-speed port of the CPU's central memory. The high-speed random access memory utilized is a product of Control Data Corporation and

is called Extended Core Storage (ECS). This memory is characterized by a 3.2 μsecs access time, 600 megabit transfer rate, and four access ports. Since the CPU is the origin of all of the AMS control parameters and is the unit that interprets the status information provided by the AMS controller, a 128-word buffer communication area is established in the random access swapping memory accessible by both the CPU and AMS. The CPU plants transfer parameters called jobs in the communication area. Groups of eight of these jobs are combined and called batches. A job contains the exact parameters for one transfer (file number, starting address within a file, starting transfer address in the random access memory, and transfer length). A batch containing up to eight jobs is intended to contain a list of all the jobs that would be required by the CPU to service a request by an individual user: program files, datafiles, and status information. The AMS controller treats the group of transfers within a batch as a unit and supplies the CPU with status information relative to the entire batch as well as relating to the individual transfers. In this manner, the CPU is relieved of the burden of figuring out whether all of the files needed for a particular user have been transferred or not.

In practice, two memory access operations are required to service a user: first, acquire all of the files that are needed to perform the task requested, and second, after processing, return to storage those files which were modified in the process. A suitable disposition of the task of rewriting some files (datafiles and status) is to attach those jobs to batches relating to subsequent users.

Under normal operating conditions, the time required for the AMS controller to execute the first batch posted is less than the time required for

the CPU to compute and post the remaining batches. For this reason, the CPU will not be required to wait while even the first batch is completed.

The basic files within the AMS memory have the following access time characteristics:

maximum access time: 819.2 μsecs

average access time: 409.6 μsecs.

However, since the AMS controller has several subcontrollers with which to manipulate files independently, the effective access times encountered in use are greatly reduced. In practice, the average access times, measured as the time unavailable for data transfer while waiting for a file to be properly positioned, is reduced to tens of μsecs. This significant reduction in access times is primarily the result of 1) the ability to independently manipulate all of the AMS files, 2) the availability of eight file manipulators, and 3) the nature of the batch posting mechanism.

2.3 AMS Batch Execution Process

Presented here is a description of the sequence executed by the AMS controller in the course of responding to a set of batches posted by the CPU.

At the beginning of a CPU execution timeslice, the CPU takes the assemblage of keys that have been inputted from various terminals and determines what files of data will be required for each key to be processed. The CPU then posts the proper transfer information in the form of one batch for each user, even if partial batches are the result. In the mean time, the AMS controller has been interrogating the communication area. As soon as the CPU has posted the first batch, the AMS controller begins to execute that batch. The AMS controller samples the CPU-AMS communication area only

once each 30 μsecs in order to avoid jamming the ECS data channel with unnecessary transfers.

Once the AMS controller has recognized that a batch has been posted by the CPU, it proceeds to assign those jobs that are contained in that batch to the eight subcontrollers. If a batch is partially filled, NOP (no-operation) jobs are assigned to the unused subcontrollers. As soon as all of the jobs associated with a batch have been assigned, the AMS controller begins to interrogate the next location of the communication area for the next batch. As a batch is posted in this location, and as individual subcontrollers become completed (those with NOPs become completed immediately), parts of the next batch are assigned to the idle subcontrollers. In the meantime, the AMS controller posts status information in the control area where the first batch was picked up, overwriting the original batch parameters. In this fashion, the AMS controller attempts to keep as many of the subcontrollers active as possible. As those subcontrollers that were issued valid jobs become ready for transfer, they request access to the data channel. When granted that channel, they transfer their assigned data. After transferring their data, the subcontrollers return the files to their original position and detach themselves. At this point, they are ready to accept another job. Any unassigned jobs in the second batch are assigned to the subcontrollers. At the point that all of the jobs in the original batch have been completed, the AMS controller writes status information to the control area one last time, indicating that the entire batch was completed, and proceeds to pick up any remaining jobs in the second batch. When all of the jobs in the second batch have been accepted by an AMS subcontroller, the AMS controller proceeds to the next batch. This sequence of processing one batch at a time and starting the next batch in

those subcontrollers that are idle continues until all of the batches that the CPU has posted have been processed. Chapter 5 discusses in detail the resultant access and transfer characteristics.

2.4 Historical Development of the AMS Control Structure

The AMS control structure was the result of a long sequence of interaction between the hardware and software groups of the PLATO system. There were three distinct design stages before the design was finalized.

The result of the first phase was a controller with eight subcontrollers, as in the final version, but all of the job parameters and status information were communicated via Peripheral Processor Unit (PPU) channels. After some evaluation and discussion with the software staff, it became apparent that this method of communication would be too clumsy for the CPU to handle and too slow to take advantage of the high performance characteristics of the memory system. In this first design the file structure was somewhat different: even though the minimum file size was 8192 words, groups of eight files were slaved to each other and were not totally independent. This resulted in a reduced efficiency of the eight subcontrollers as compared to the final, totally independent design.

In the second design of the AMS memory system, it was decided to communicate all job parameters and status through the same data channel through which data was to be transferred. In this version, however, the control area consisted of just 16 words, eight used for job parameters and eight used for status. Each job was treated individually, and the CPU was given the task of keeping track of which job was related to which user's request. In this version, however, the files were made totally independent of one another, so that just one 8k word file needed to be activated at a

time. The problem with this design was that the CPU overhead was too high, and the CPU could not post several batches ahead to overlap transfers and processing.

In the third and final version of the AMS structure, the large communication area was added and the batch orientation was implemented. As a result of the third design cycle, a serial memory control structure specifically designed to operate in the environment presented by the PLATO system was developed. Since there was an extensive amount of interaction between the software group that would use it and the hardware group that would build it, a very useful and powerful yet feasible structure was developed.

CHAPTER 3

POTENTIAL SERIAL MEMORY MEDIA

3.1 Introduction

There are three prime contenders for the high-performance serial mass-memory market. These are 1) Semiconductor Serial Memories (MOS and CCD), 2) Magnetic Bubble Memories, and 3) Electron Beam Memories. These three differ greatly in their technologies.

3.2 Charge Coupled Devices

The Semiconductor Serial Memory is the most advanced of the three serial media. Actual production devices have already been introduced into the semiconductor market, and larger, faster devices are promised shortly. The significantly advanced state of the semiconductor devices is due largely to the corresponding advanced state of the semiconductor random access memory, which has been getting very large attention and is expected to be the dominant device in the high-speed, high-performance memory area. The technologies developed for random access devices have been applied also to the serial memory field. Of the semiconductor devices available, CCD and MOS, it appears that CCD has the advantage due to inherent simplicity. However, MOS has advantages because it is more heavily used throughout the semiconductor industry. In fact, the first CCD devices which have been introduced are hybrids of the two technologies [10].

A CCD device is basically an inactive device (contains no active transistors at the memory-cell level). Instead, the storage of data is similar to the storage of data in a transmission delay line but the transmission line is electronically generated and is often compared to a fire bucket brigade. Charge (water) is placed in a location (bucket) at the beginning of the line and passed on the end of the line by electrodes (men). Along

the way, no water is added to the bucket and since some spills or leaks out, less arrives at the end of the line than started at the beginning. In the case of the CCD register, data is represented by the absence or presence of charge at the end of the line during one cycle and this information (charge or no charge) is amplified and fed back into the beginning of the line again so that the line and data storage are endless. The principal advantage CCD technology has over other forms of semiconductor serial memory is that there are no active devices at the cell level and thus the design of the actual cell is very simple. A severe disadvantage exhibited by the semiconductor bucket brigade that is not seen in the fire bucket brigade is the fact that charge leaks away even when there is no transfer, so that it is necessary that each data element arrive at the end of the line often enough that no data is lost. If the line is long (requiring a small amount of support circuitry), the data must move quickly from one end to the other and consequently use a larger amount of power; if the line is short, a large amount of support is required and the advantage of the simple cell is lost. An optimum compromise can be reached: a register which has to operate at a medium speed so that it can be made several hundred bits long yet not dissipate an excessive amount of power. This device will have a limited dynamic range over which it will operate. Since varying the speed of the register is one technique that is used in the AMS controller to optimize the access-time performance, and this technique is not applicable were a CCD-based system implemented, some modification of the controller would be necessary if CCD were used. In particular, an intermediate level of buffering would need to be added to act as a "rubber band" between the CCD memory and ECS.

The projected cost per bit at the device level of CCD memory in 1977 is around 0.01 cents per bit. Based on this number, the system cost would be

about 0.02 cents per bit [11].

3.3 Magnetic Bubble Memories

Magnetic bubble memories were first developed at Bell Laboratories [12] and are similar to CCD devices in being passive storage devices. An additional advantage of bubble memories is that data stored in the memory may be retained for indefinite periods of time without the application of power. To date, however, the development of these bubble memories has been slowed by the search for a suitable substrate material and production devices have been released. In addition, bubble devices which have been operated have been limited to fairly low operating speeds. Bubble memories have a long development cycle ahead of them.

The basic concept of a bubble memory is that a thin layer of magnetic material containing inherently strong magnetic dipoles is exposed to a strong continuous magnetic force. This material separates into domains of oppositely polarized areas, each with its dipoles aligned at 0° or 180° to the applied field. The separation of the domains within the magnetic layer is stable. If electrodes are used on the surface of the material to apply additional localized fields, the domains can be defined and moved about the surface. In particular, electrodes can be designed to cause a row of these domains to move in a line; thus, this row becomes a serial shift-register. One mil (1/1000 of an inch) diameter bubbles have been generated and manipulated and this size of cell would result in 10^6 bits per square inch of material or 10^9 bits per cubic inch. The constant field can be applied by using a permanent magnet, and the application of this field is the only requirement for data retention in the memory. A variable field can be applied to the entire unit if it is desired that all of the data in a

particular unit move in the same manner, as would be the case were it used as a serial shift-register [12].

Due to speed limitations and involatility, bubble memories will likely be future substitutes for that level of memory now occupied by moving head disks, adding a massive increase in bit-space density and thus portability. This memory medium will probably not become an effective source of high-performance memory in an AMS-type system.

3.4 Electron Beam Memories

The first form of random access memories used in early computers consisted of storage tubes, called Williams tubes. These cathode-ray tubes used charge deposited by the electron beam as the storage medium and achieved rapid random access through positioning of the beam. The advent of ferrite core technology displaced the electron beam medium. Recently, however, in the search for a very high density, low-cost medium for mass memory, the electron beam technology has been re-investigated and significant results have been achieved. The techniques used in the latest electron beam work are somewhat different from the earlier versions, principally in the actual storage medium and the nature of the addressing. In the later units the storage medium is a multi-layered semiconductor material, and the storage mechanism is charge deposited inside the semiconductor rather than charge deposited on a surface. Data is written into the memory by the application of a bias voltage to the semiconductor material during the application of the electron beam, with the resultant diffusion and deposition of the electron-hole pairs created by the beam. The addressing in the early units was strictly random access, whereas the addressing in the later units was generally serial in nature because of a high-speed data flow requirement and the problems of positioning an electron beam to within the accuracies

required. Single tubes have been demonstrated in the laboratory containing up to 30 million bits in 100 cm^2. The positioning of a beam to this level of accuracy in short periods of time would be unrealizable; rather the beam is first deflected to a general area and then is redeflected by a second deflection system driven by an analog ramp generator. The result is that data input and output are in blocks and serial in nature [13].

Electron Beam Memories promise the availability of very large amounts of medium-speed memory that is non-volatile. This technique will be a major competitor of CCD devices in the market of inexpensive mass memory. One disadvantage of Electron Beam Memories is that the basic unit of memory is very large and, therefore, defines the minimum size memory system to be very large. If the basic tube is 30 million bits in size, and if the memory system requires that parallel tubes be used to keep the data transfer speed high, the number of words in the minimum system will be 30 million. Charge-coupled devices on the other hand will be available in smaller sizes, certainly less than 1 million bits per device, thus allowing smaller storage systems and allowing parallelism at a larger than word level to further increase the transfer speed.

CHAPTER 4

AN AUXILIARY MASS STORAGE SYSTEM

4.1 Introduction

Given a serial memory medium as a base, is it possible to configure it and its controller in such a fashion as to make it appear to the Central Processing System as if it were random access in nature or at least configure it in such a fashion as to allow the Processing System to use it without appreciable degradation? The answer is yes if appropriate constraints are placed on the memory, the computer system, and the nature of the computer service to be offered. The memory system considered is a serial semiconductor memory; the computer system considered is a Control Data Cyber 73 [14] computer with a large amount of Extended Core Storage (ECS); and the computer service offered is that offered by the PLATO computer-based education project. Under these conditions, this work will show that a controller can be designed to interface a serial memory to the PLATO system and provide memory service with very little degradation compared with a system using only ECS.

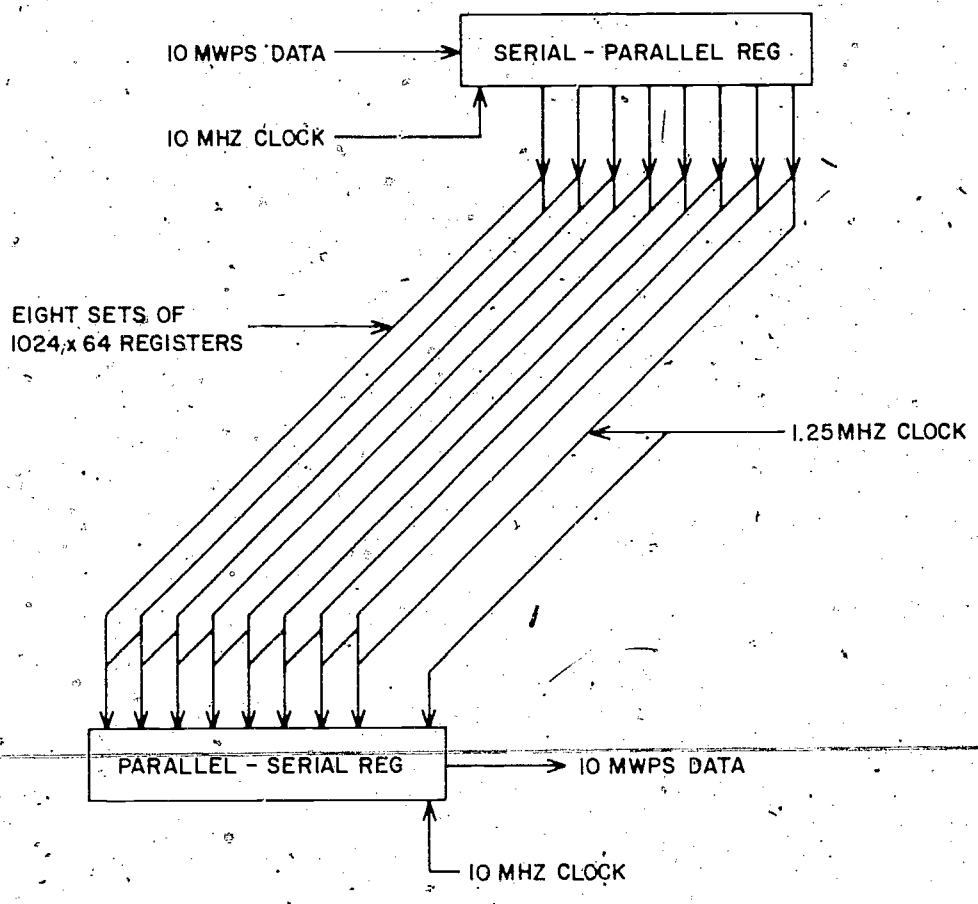
4.2 Configuration Options

In the design of the AMS system, two distinct types of design decisions were required: 1) basic architectural decisions and 2) specific implementation decisions. The architectural decisions have a large bearing on the performance of the working system and the ability of the system to satisfy user requirements. The implementation decisions determine the ease with which the unit can be implemented, the problems encountered in that implementation, and the reliability of the final system.

4.2.1 Architectural Options

The basic technique adopted in order to cause the AMS to perform satisfactorily in the PLATO environment was to increase the transfer rate by parallelism and to cut the access time with multiple control units in the controller. In addition, since it is not useful to have a fast memory system if it is not possible to supply transfer parameters to it quickly, control communications were made very fast and efficient.

The ECS system that the AMS is designed to interface into operates at a speed of 10 megawords per second based on a 60-bit word. In addition, this transfer needs to be synchronized at the level of eight-word records due to the structure of ECS. Even if the average data transfer rate into ECS were only one megaword per second, this data would have to be transferred in sets of eight-word records at 10 megawords per second. Since it was necessary to support this high rate on a record basis, it was not difficult to design the AMS so that is supported a continuous 10-megaword rate. The basic 1024-bit shift-register is only capable of operating a conservative 1.25 megaword per second (assuming 64 parallel registers to form a word), and so 512 (8 × 64) shift-registers were assembled and operated in parallel to supply the desired 10 megawords per second. Figure 4.1 details this multiple register configuration used to generate 10 million 64-bit words per second with read/write capability. The assemblage of 512 individual registers shown in Figure 4.1 provides an 8192 word, 64-bit memory section: the section is called a file and is the minimum access element. Thirty-two of these files are assembled to form a 262K word bank and the total capacity of the AMS is 16 banks or four million words.



AN AMS 8192 WORD, 64-BIT FILE

Figure 4.1

Multiple access control paths were used to allow the AMS controller to speed up the effective access times of the data transfers. The number of these multiple paths was chosen to be eight for several reasons. In the PLATO system, when a keypress is received by the CPU, this keypress might require the access by the CPU to several different data areas in order to process that keypress: student banks, lesson material, a common block, and so on. Furthermore, all of the possibly required information must be available in ECS before the CPU can begin processing this keypress. It is, therefore, advantageous that the AMS controller be able at least to process simultaneously all of the requirements for one keypress. The number of data blocks was estimated to average three with a reasonable maximum of eight. The second factor considered in the decision of how many access control paths to implement was the fact that, as detailed below, the transfer parameters communicated through ECS require the use of eight-word records. For these reasons, eight separate transfer controllers or sub-controllers were implemented.

The fastest data communication channels available in the Cyber 70 series computers are the paths into and out of ECS. This path, which is the same channel used to communicate data between AMS and ECS, was also used to communicate the parameters of the transfers that the CPU would like to have the AMS controller perform. The CPU establishes a 128-word communication control area to which both AMS and the CPU have access. The CPU plants in this area the jobs, and AMS plants in this area the job status. The 128-word area is separated into 16 sets of eight words. The AMS controller operates principally on one set at a time and flags the completion of these sets to the CPU so that the CPU can operate on the data that the AMS has transferred. The use of 16 sets allows the CPU to assemble many

sets of jobs at one time and the AMS will sequentially execute the sets of jobs without CPU interaction. The eight jobs in a set correspond to the eight subcontrollers in AMS. In addition, a single PPU channel, capable of a maximum transfer rate of only 12 megabits per second, was utilized to communicate basic setup parameters to the AMS controller, such as whether to commence operations and where the CPU has established the AMS-CPU communication area. It is important that during the course of execution of jobs issued by the CPU no information need be transferred via this slow PPU channel, as this would greatly degrade the performance of the AMS controller.

4.2.2 Implementation Options

After the archictectural structure of the AMS controller was defined, there still remained the necessity to define the various implementation options still to be determined. Among these were choices such as the type of logic to use in the AMS controller, the method of construction of the controller, the physical configuration of the memory modules and control modules. The decisions were made based on the previous experience of the designer, on the mechanical constraints of the construction capabilities of the laboratory in which the work was to be done, and on the immediate availability of the integrated circuits.

4.2.2.1 Controller Implementation

The electronics laboratory of the Computer-based Education Research Laboratory is experienced in the construction of medium size electronic equipment. The equipment previously built has generally been that necessary to communicate with the Cyber-70 series computers via PPU channels operating at a maximum speed of 1 MHz. For this reason the structural techniques used in the AMS controller were similar to those previously employed, with some modifications to account for the increased operating speeds (10 MHz)

and complexity. In addition, at the time of the original design of the AMS controller, there were available two logic families designed for operation at speeds in excess of 10 MHz: Schottky Transistor-Transistor-Logic (TTL) [15], and Emitter-Coupled-Logic (ECL) [16]. The ECS is rated to operate much faster than the TTL but requires more power and does not have available the large assortment of complex functions that the TTL family has. The ECS logic is, however, truly a high-speed logic family, whereas the Schottky TTL is more a medium-speed family which has been speeded up through the internal use of Schottky transistor technology. All of the previous experience of both the designer and laboratory has been with the TTL family and this outweighed the advantages of ECL. The TTL family was used.

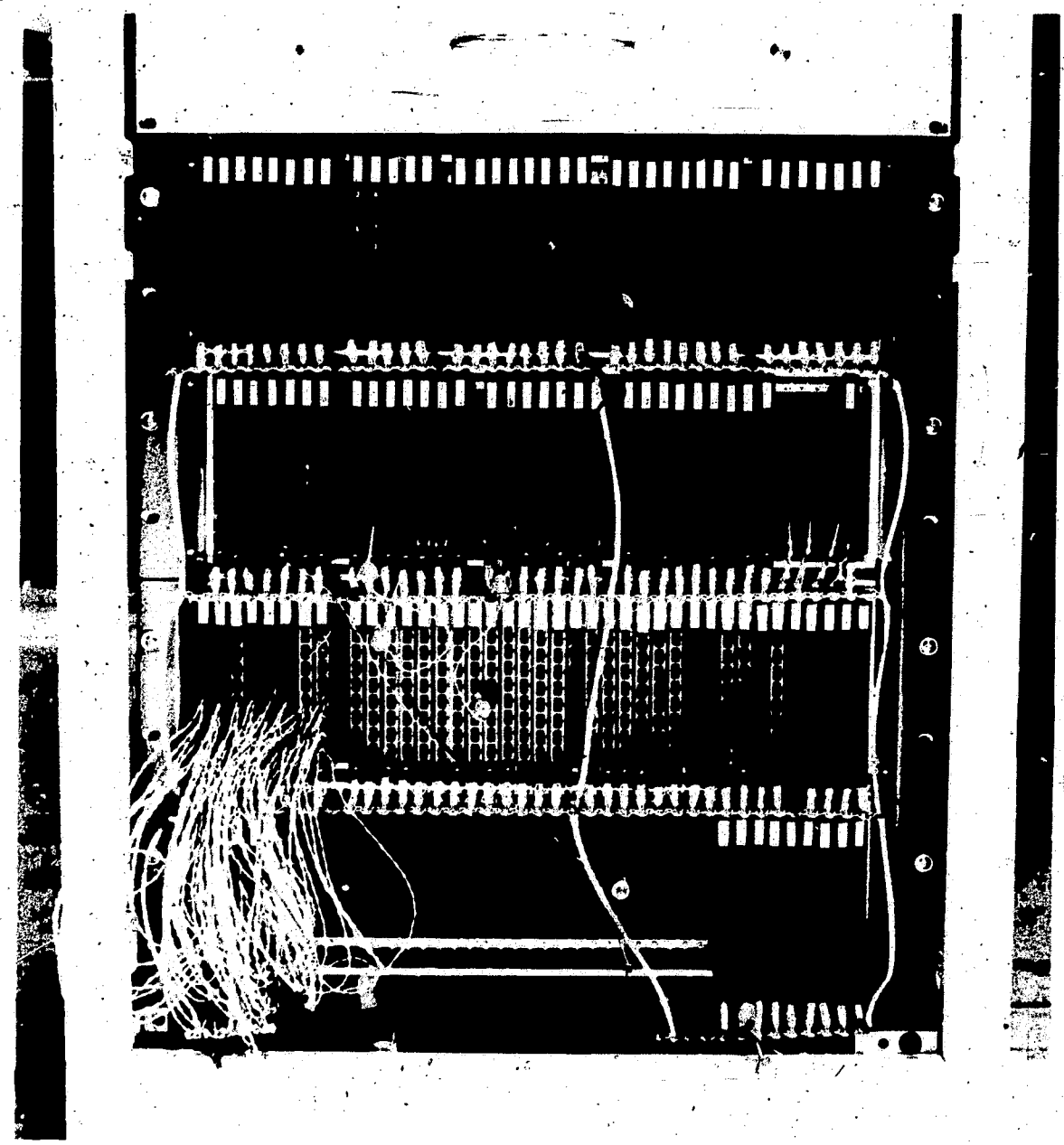
The construction techniques used in the AMS controller are the same as those previously employed in large digital laboratory projects. One parcels up the required circuitry into sections which are 8-12 circuits each and develops printed circuit boards for each of these parcels. The boards are built and interconnected through the use of a printed circuit edge connector and back plane wiring. This technique provides a very high density of circuitry, allowing convenient mixing of integrated circuits and discrete components, and takes advantage of whatever repetition is available in the circuit design. An alternative approach would be to utilize commercially available random logic panels at the cost of some density, increased difficulty in handling discrete components, and loss of the advantages of repetitive circuitry. There would be two significant advantages of the use of these logic panels: 1) all of the circuitry is available to be probed at one time, whereas with individual cards many of the circuits are hidden at any particular time, and 2) the parameters of the logic board are such that all of the interconnections are uniform in their transmission

characteristics and so the problems of operating at very high speeds are somewhat simplified.

Figures 4.2 and 4.3 are pictures of the AMS controller which was implemented.

4.2.2.2 Memory Implementation

A single bank of AMS memory (262K 64-bit words) requires the installation of 16,384 memory devices plus approximately 1500 other integrated circuits. Were the construction techniques used in the controller extended to the memory, roughly 1800 cards would be required, and this number is totally unreasonable. Instead of 8-12 circuits per card, the memory was implemented by examining the required interconnections and choosing a single card configuration which provided the minimum printed circuit card interconnections, thus saving the cost and time of the back plane wiring as well as increasing the reliability. A card configuration of four bits of eight files was chosen which required a printed circuit with 256 memory circuits and 16 support circuits. Including power, 88 printed circuit connections are required per card. Sixty-four cards are required for a complete bank. In addition to the memory card there are three other card types necessary to implement a memory bank: a dual file control card, of which 16 are required, and two bank control cards, of which one each is required. Figure 4.4 is a picture of the memory card and Figures 4.5 and 4.6 are pictures of the memory bank which was implemented. Figures 4.7 and 4.8 are pictures of the full quarter-million word system.

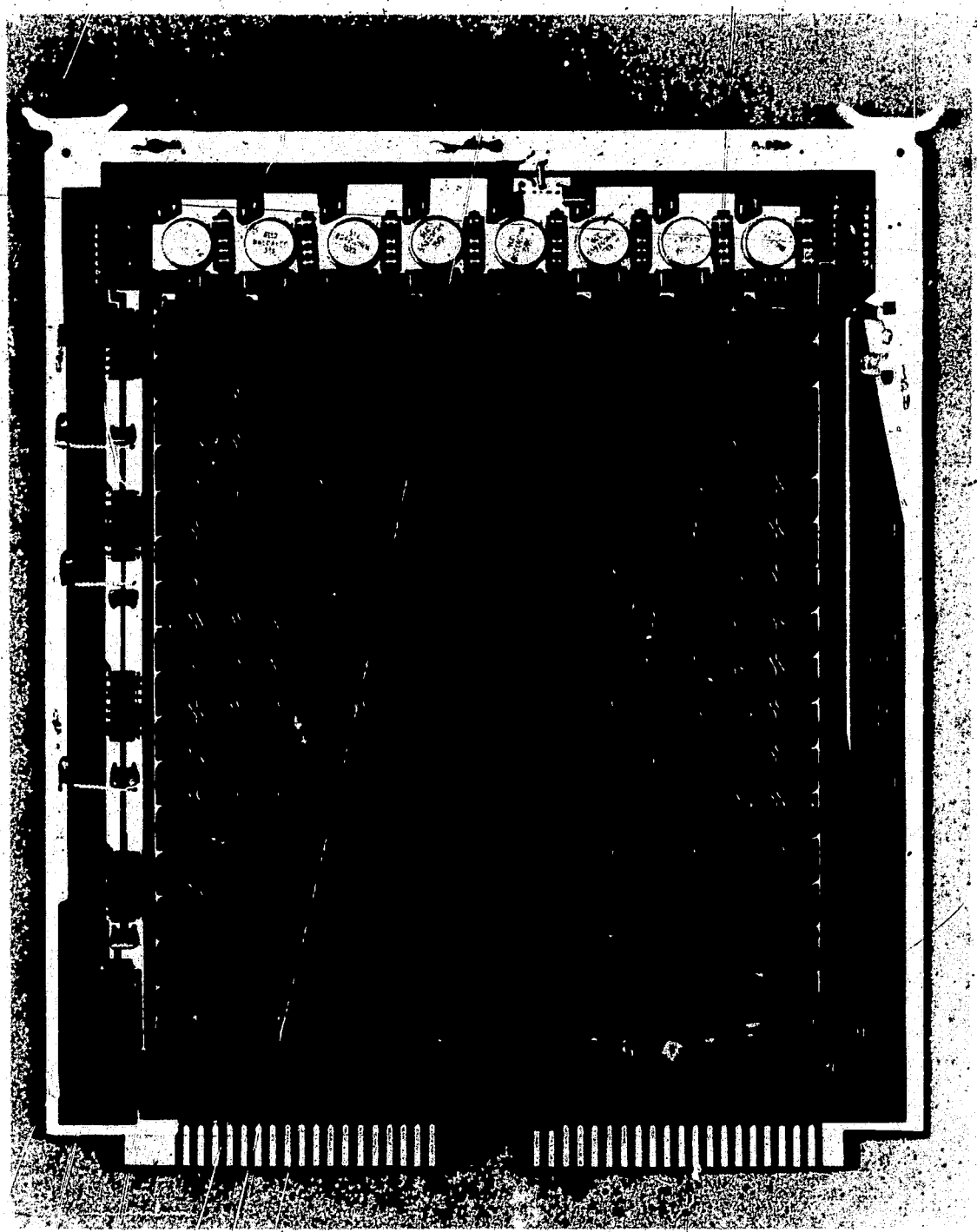


AMS Controller Front

Figure 4.2

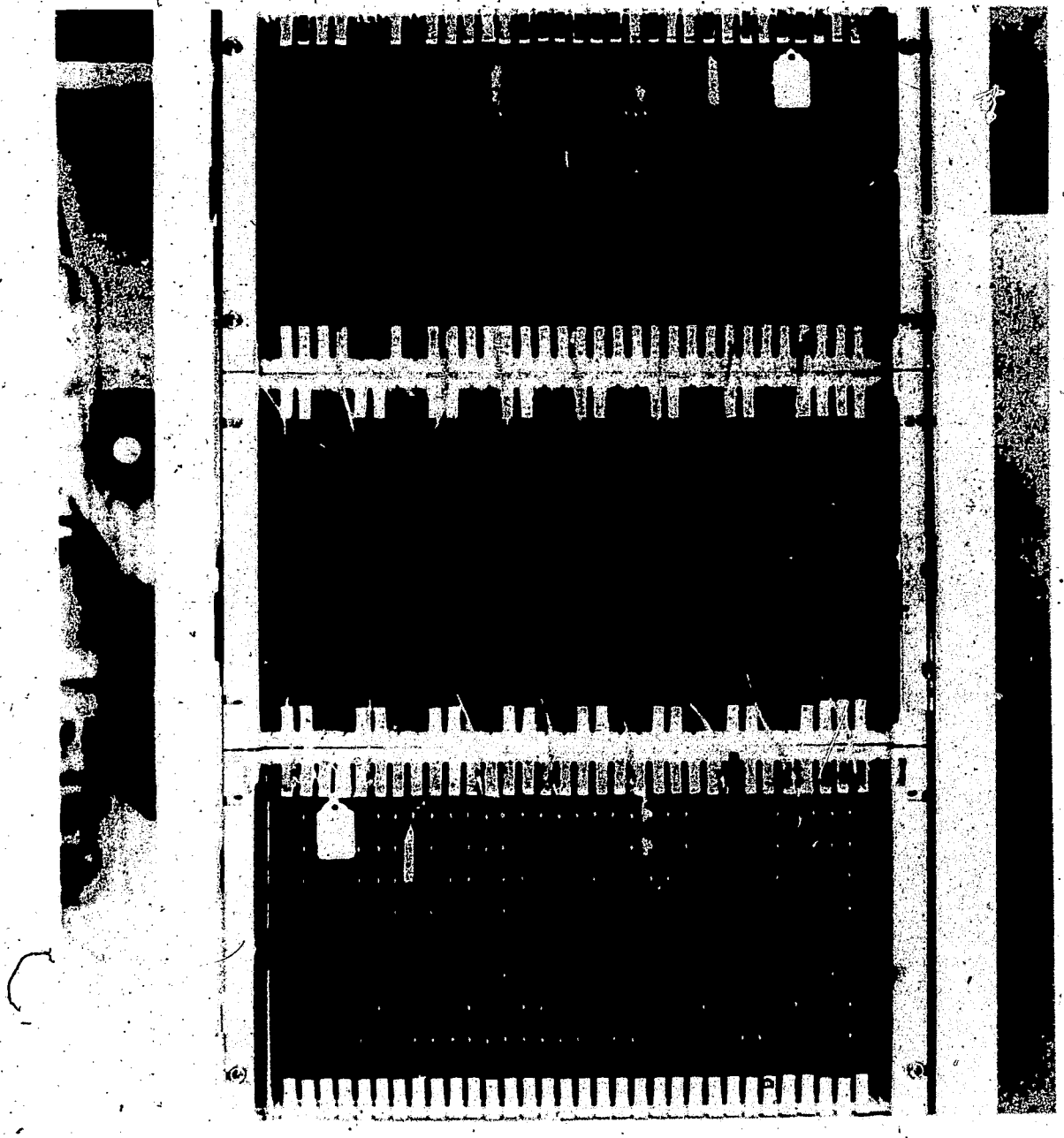
AMS Controller Rear

Figure 4.3



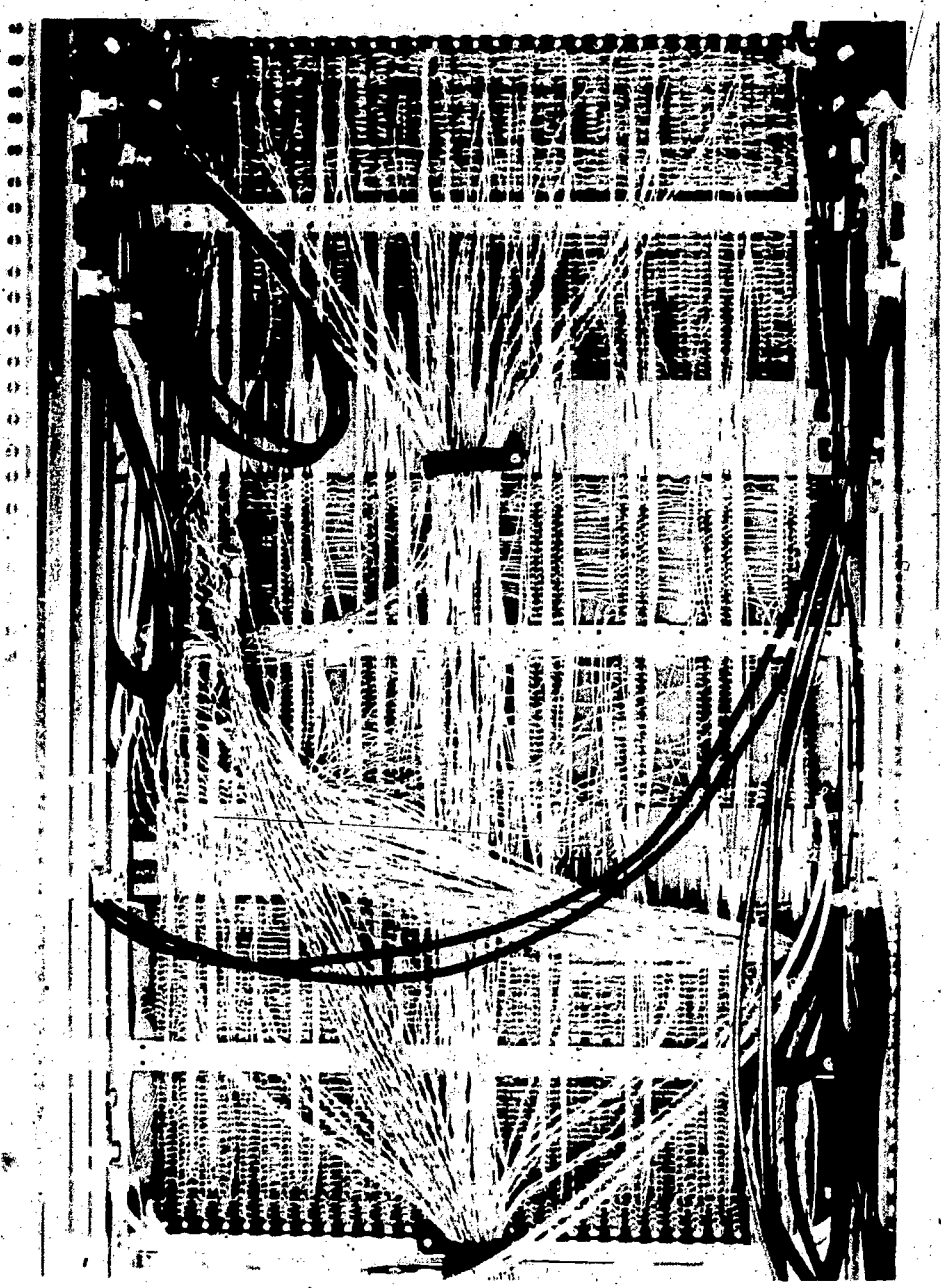
AMS Memory Card

Figure 4.4



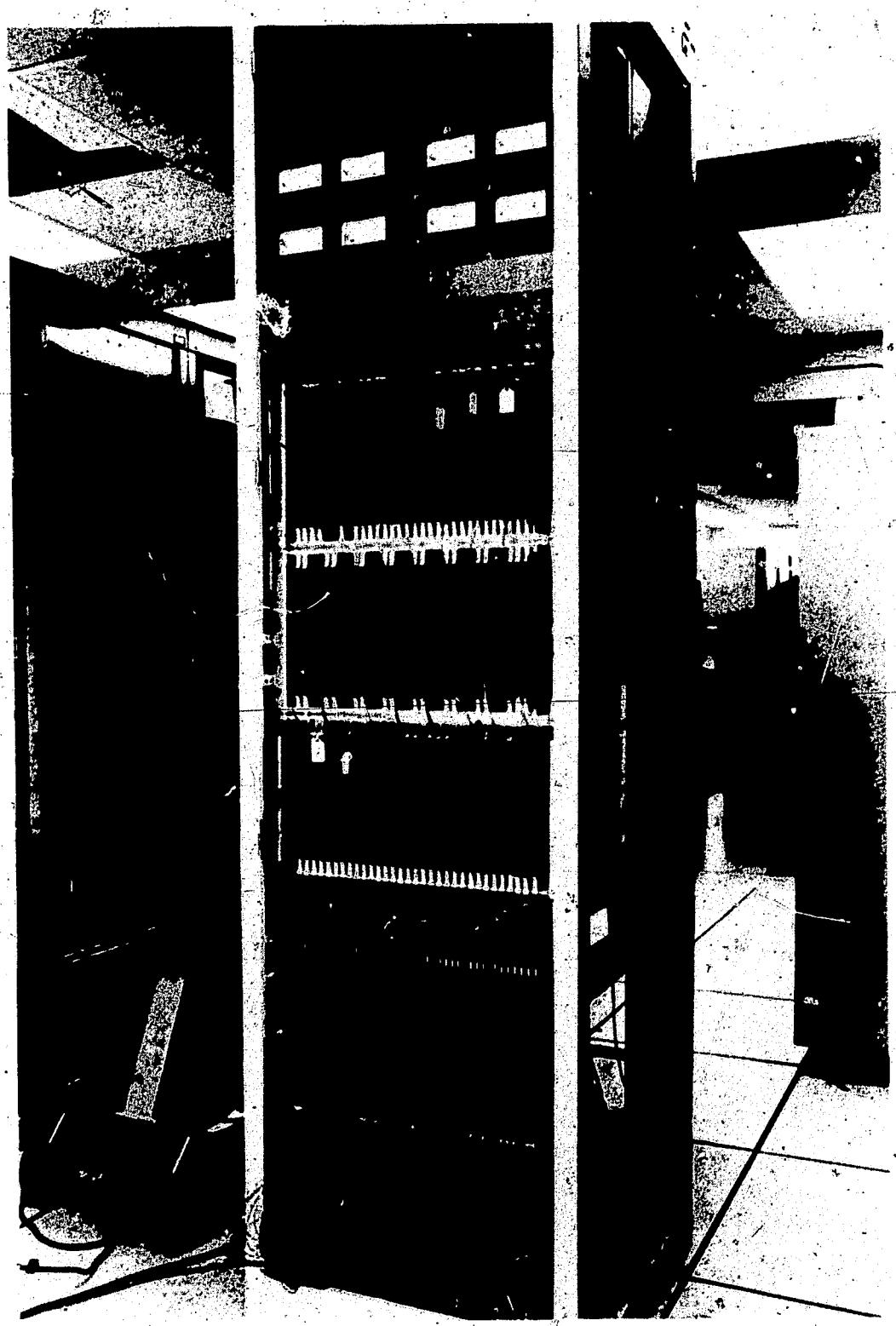
AMS Memory Front

Figure 4.5



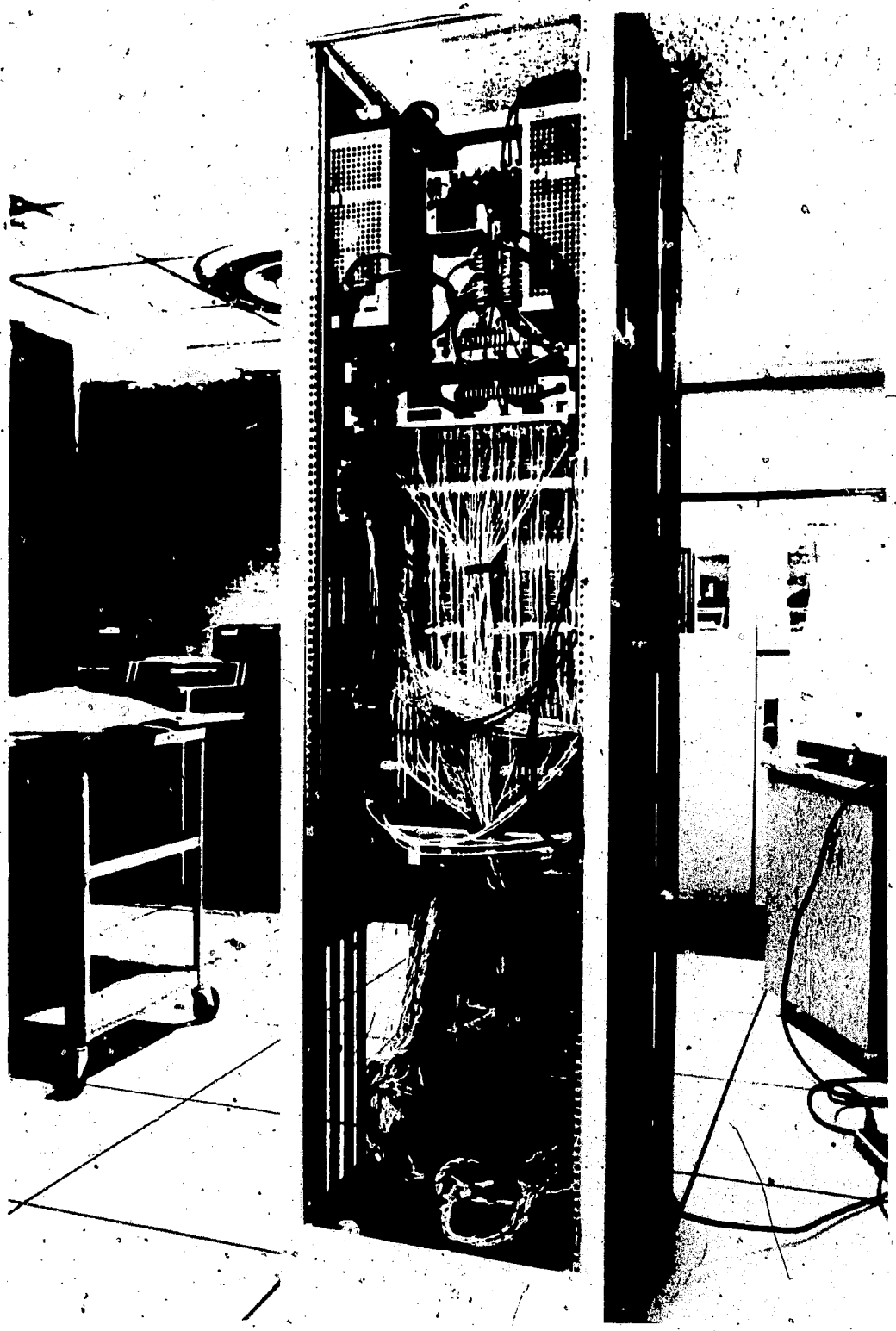
AMS Memory Rear

Figure 4.6



AMS Memory System Front

Figure 4.7



AMS Memory System Rear

Figure 4.8

CHAPTER 5

SYSTEM PERFORMANCE

5.1 Introduction

The performance of the AMS memory system will be described in two sections: 1) Controller performance, and 2) Memory performance.

5.2 Controller Performance

The PLATO system is heavily dependent on the availability of a very high-speed mass memory and presently uses two million words of Extended Core Storage (ECS). In this memory are stored many types of data necessary for the proper operation of the system. The position of an AMS-type memory as a partial supplement to the large ECS bank and as a further extension of this memory to the much larger numbers of words necessary to service thousands of users depends on its ability to provide a service level comparable to that of the present ECS system. The tasks that will be assigned to this new memory will be similar to but increased over those presently assigned to the ECS memory. The AMS controller, whose structure determines the AMS system performance, was designed with the particular characteristics of the PLATO system in mind. The corresponding performance levels of the controller were determined by modeling the task profile of the PLATO system and testing this model in side-by-side operation of PLATO and a parasitic AMS driver.

5.2.1 Performance Model Parameters

The general operation of the AMS controller can be modeled based on the variable parameters, both those determined by the software usage and the hardware constraints. The software determines the transfer parameters such as transfer length, file, file starting address, and ECS address. Not generally under software control are variables such as AMS-CPU conflicts,

file access conflicts, and phase skew between the idle address and the transfer starting address. Enough information is available to the CPU for it to minimize the AMS-CPU conflicts; file access conflicts, and phase skew, but by examining the software overhead required to evaluate this information it became apparent that this increased overhead would result in a net loss in overall performance.

AMS-ECS conflicts are those occurrences of simultaneous requests into the ECS controller by both the CPU and AMS. A reduction of the transfer rate of AMS results because AMS has a lower priority than the CPU. The occurrences of these events can be minimized by calculating the expected time of an AMS-ECS transfer and having the CPU conduct tasks which only require CM activity.

File access conflicts are the occurrence of a request by a particular AMS subcontroller to attach a file which is already attached to another subcontroller. When this situation arises, the requesting subcontroller will simply wait to start its job until the required file has been released by the other subcontroller. This situation occurs only after completion of the assigned transfer job. The CPU can minimize the occurrence of this conflict by assigning only one job pertaining to an individual file (many different data areas might be contained in one 8192 word file) per job batch (eight transfer jobs).

The AMS controller communicates the instantaneous rotational position of all of the files which are not attached to an AMS subcontroller whenever it writes status information to the ECS communication area. These "idling" files are continuously rotating at a slow rate (one revolution per 100 μsecs to refresh the volatile memory). The CPU could utilize this information to choose which of a possibly large number of jobs to assign choosing first

those which would require the minimum rotation of the required file prior to the data transfer. This would result in a general reduction of the average data access times.

The implementation of these software "speed-up" techniques would require a large amount of CPU overhead on a per job basis and is not practical. In addition, the actual performance of the controller and AMS memory system is already sufficient in the PLATO environment without these further optimizations.

The model for the AMS controller operation takes into account the following parameters: 1) Transfer length, 2) Search length, 3) Return length, 4) ECS channel access queuing and CPU-AMS access conflicts. The AMS hardware elements involved are: 1) the actual data file, 2) the eight access subcontrollers, 3) the single 600 megabit per second ECS data channel, and 4) the idle file controller.

Definitions:

Transfer Length: The CPU assigns a transfer length to an AMS transfer job by specifying between 1 and 1024 records to be transferred (a record is eight words). The data length will be transferred to or from ECS sequentially from the starting address, which is also specified by the CPU. This variable is indicated as TL (Transfer Length) and has an average value $\overline{TL}$.

Search Length: The CPU assigns a starting address (from 0 to 1023 records) where the AMS subcontroller is to start transferring with ECS, and the subcontroller must rotate its specific file (always in the forward direction) from its instantaneous position to this starting position. The Search Length is the number of record locations that the subcontroller is required to rotate its file prior to transferring data, which is the difference of the starting address and the instantaneous idle address at the time the

file was attached to the subcontroller. This length is assumed to be a uniformly distributed number between 0 and 1023 and the time required to move the search length is SL (Search Length) × .8 μsecs. The average of this length is 512 records and is indicated as $\overline{SL}$.

Return Length: Once the subcontroller has conducted its assigned data transfer, it must return the file to the control of the idle controller. It must first align the file with all of the other files controlled by the idle controller and then detach itself, thus attaching the file to the idle controller. This phase of a job's execution is a function of the size of SL, TL, and the number of rotational locations that the idle controller has moved since the file was detached. If the idle controller did not rotate at all, the total of SL + TL + RL (Return Length) would always be an integer multiple (N) of 1024, which is the file length. Instead, the sum is $N \times 1024 + \delta$, where δ = T (time)/100 μsecs because the rotational stepping rate of the idle controller is one step per 100 μsecs. The total time of SL + TL + $\overline{RL}$ if N = 1 is approximately 800 μsecs (1024 × .8 μsecs) and would therefore result in a δ of 8 so the sum of SL + TL + RL + δ = 1024 + 8 = 1032, or less than a 1% increase. Larger values of N (the largest possible value is 3) and conflicts for access to the ECS data channel cause larger values of δ. The worst case value is 72 and results when a particular file is required to wait for access to the ECS data channel for seven other maximum length transfers and had a SL = TL = RL = 1024. The worst case value of δ results in a 7% increase. Because this percentage is very small, future calculations will assume δ = 0.

ECS Channel Queuing: Even though there are eight AMS subcontrollers simultaneously executing transfer jobs, only one channel to ECS is available for the transfer of data and all data must be transferred over this channel.

This ECS channel transfers at 600 megabits per second, which is a record transfer rate of one record per .8 μsecs. Access to the channel is granted to just one AMS subcontroller at a time on a first come first serve basis. Once the channel is granted to a subcontroller, that subcontroller maintains the channel until it has completed its transfer, which is a function of the transfer length. After the completion of a transfer, the next numerical subcontroller requesting the channel is granted it.

When a subcontroller has performed the job of positioning its file properly for a data transfer, it requests access to the ECS channel in order to conduct that transfer. If the channel is busy, the requesting subcontroller waits until the channel is available.

AMS-CPU Access Conflicts: The ECS controller [17] has four ports which allow access to the ECS storage system. In the PLATO system these ports are connected to 1) the CPUs, 2) the Distributed Data Path (DDP) [18] which allows direct PPU to ECS transfers, 3) the Side Door Adapter (SDA) [19] which allows the disks to communicate directly with ECS, and 4) AMS. The DDP and SDA can be neglected with regard to their interaction with the AMS and CPU, because the activity in these units is very small. The interaction between the AMS and CPU is significant, however, and results in what is referred to as AMS-CPU ECS Access Conflicts, to be discussed later.

AMS Data File: The datafile in the AMS is an 8192 by 64-bit memory area of which 60 bits are used as data area and a single bit is used for parity check. The remaining bits are not used at this time. The datafile is a dynamic data area and so must be periodically refreshed or regenerated, and this function is accomplished by shifting the file at a minimum rate. Even those files which are not engaged in a transfer job must be shifted in order to insure that whatever data is held in them remains intact. The

manufacturer's specifications for the device require a minimum rotation rate of one shift per millisecond, and the AMS memory guarantees that one shift is performed every 0.1 msecs.

The Eight Access Subcontrollers: The subcontrollers within the AMS system are the logical units that prepare a datafile prior to a data transfer and then, after that transfer, prepare the file for return to the idle state. These units operate totally independently once they have started to conduct a job. They interface with the unit which "hands out the jobs" as they are communicated from ECS, the unit that attaches a file to a subcontroller or detaches a file from a subcontroller and the unit that allocates the ECS data channel. When the subcontrollers are operating on a file and are not waiting for another section of the controller such as waiting for the data channel, they operate at 10 megawords per second. This speed is the maximum speed of file movement.

The ECS Data Channel: The AMS-ECS data channel is port number 4 into the Control Data ECS Memory Controller. Under normal conditions, each of the four controller ports has equal access to the ECS system, and the total transfer capability of the four ports combined is 600 megabits per second, in 60-bit words. This transfer rate is available only if the requests made to the ECS controller are sequentially directed to the four different bays of the memory (ECS) because these bays are interleaved to this level of four. Under the condition that one single port is requesting access to memory and is transferring blocks of data to successive locations in ECS, the sequential request criterion is satisfied. If more than one port is requesting access to memory, or if a single port repeatedly requests access to the same bay of memory, the sequential request criterion cannot be generally satisfied, and as a result, both the individual and total data transfer rates degrade.

In the Cyber 70 series computer systems, when either of the two CPUs proceeds to conduct an ECS data transfer, hardware constraints dictate that no CPU activity can be simultaneously performed even in the CPU that is not executing the ECS read/write instruction. For this reason, it is vitally important that all ECS operations required by the CPU be conducted at their maximum possible rate to reduce the CPU dead time. If AMS activity is heavy, there is a large possibility that both the CPU and AMS will request ECS activity at the same time, and the result would be a degradation in the performance of the CPU system. The AMS-ECS channel is different from the other three ports into ECS by the inclusion of a special signal which indicates that the port of ECS that is assigned to the CPU system is requesting ECS access. The AMS controller uses this new signal to reduce its transfer requests to a minimum when the CPU is requesting. The transfer cannot be totally stopped because the datafiles are dynamic and require at least one shift per msec. This reduced rate causes only a maximum of 0.3% conflicts for the CPU, though the AMS-ECS transfer rate can be significantly degraded.

Idle File Controller: All AMS files that are not under the control of a subcontroller are placed under the control of the Idle File Controller. This unit performs the necessary refreshing of the data and keeps track of the rotational position of each file. Since the data is in shift-registers, there is no such thing as absolute addressing. Instead, particular data bits can be in any physical position. The data address of the physical position is a function of how many shifts have occurred since an arbitrary starting instant. An address register is kept by the idle controller, and all files are aligned with this register. When the control of a file is transferred from the idle controller to a subcontroller, the contents of the

idle address register are also transferred to the subcontroller. This information is used by the subcontroller to determine how to rotate the file. When a subcontroller is finished with a file and wants to reattach it to the idle controller, it rotates the file until it is again aligned with the idle control address register and then detaches. The idle controller rotates files at 10,000 shifts per second.

5.2.2 Model Performance

In the PLATO environment, the most indicative factor concerning the level of performance of the AMS controller is the time required to complete all of the jobs in one batch since all of the transfers related to a particular user's demand would be packed into one batch, and the CPU must wait or otherwise engage itself while that batch is in process. In addition, even with processing overlapped with swapping, the average time for swapping in and out must be less than the average process time; otherwise the CPU will eventually run out of work to do. A batch contains from one to eight jobs, with each job having its individual parameters. The parameters of each job and their relation to the parameters of the remaining jobs determine the time required to completely execute a batch.

For a running system, it will be shown that usually only the execution time of the first batch among several will be important, because the average time required by the CPUs to process an individual user's data is longer than the average time required for the AMS controller to execute the batch for the next user. The AMS controller will, therefore, be ahead and continue to get further ahead as long as there are batches to do.

The simplest batch is one containing all null jobs or no-op's. A particular bit is set aside in the job-control word sent from the CPU to indicate a no-op job. When the AMS controller encounters a no-op job for

a subcontroller, it proceeds to assign it and the subcontroller accepts it. However, the subcontroller immediately indicates that it has completed the assigned job and is ready for another. The time required for a subcontroller to execute a no-op job is approximately 5 μsecs. If all of the subcontrollers have no-op jobs, the batch is 5 μsecs as well. The AMS controller does not indicate this situation to the CPU except once each 30 μsecs, however, to avoid clogging the ECS channel with status updates.

The next simplest batch is one containing just one job, and this batch presents a special case of the batch process model. When the AMS controller engages in a single job batch, no overlapping can be employed. The batch execution time is simply the job execution time, which is:

$$\text{Time (T)} = SL + TL + RL.$$

(Search + Transfer + Return)

As stated before, the sum of:

$$SL + TL + RL$$

must be an integer multiple of 1024 records if the rotation of the idle controller is ignored. The integer is a function of TL and the exact relation between the starting address of the transfer and the idle address when the job was started. If the file is positioned at a location which is before but not within the area that is to be transferred, the SL will be that distance to the beginning of the transfer area, the TL will be the transfer length, and the RL will be (1024 - TL - SL). In this case, the integer is 1. If the address transferred from the idle controller is within the transfer length area of the file, the file must be rotated through the remainder of the transfer length and around to the beginning of the transfer area. Then TL is cycled through, and at this point, more than one complete cycle has already been completed. The RL will be (2048 - SL - TL) and the

integer is two. In the special case where the idle address is very close to the transfer starting address and the TL is very close to 1024, a total of three revolutions might be necessary to complete a job. The SL will be 1024, the TL is 1024, and the RL is 1024. The integer is 3.

The average time to process a single job batch assuming a uniform distribution of job lengths and no correlation between the starting address and the idle address, is:

$$P(N=1) \times 819.2 + P(N=2) \times 2 \times 819.2 + P(N=3) \times 3 \times 819.2 \text{ µsecs}$$

where P(N=x) is the probability that the number of cycles equals x.

P(N=1) is the probability that only a single rotation will be required, which is the probability that the idle address falls outside of the transfer field. If we assume an average transfer length of 512, we have:

$$P(N=1) = \overline{TL} / 1024 = 512 / 1024 = .5.$$

P(N=2) is the probability that two rotations of the file will be required to transfer and return, which is the probability that the idle address falls within the transfer field.

$$P(N=2) = (1024 - \overline{TL}) / 1024 = 512 / 1024 = .5.$$

P(N=3) is the probability that three rotations of the file will be required in order to process a job. This condition occurs only when the idle address is almost exactly the same as the transfer starting address. Its probability is very small and will be neglected.

Therefore, the average time to process a single job, assuming that no other jobs are competing for the ECS channel and the CPU-AMS conflicts are negligible, is:

$$T = .5 \times 819.2 + .5 \times 2 \times 819.2 = 1230 \text{ µsecs.}$$

For non-uniform distributions of TL, the probability of a batch requiring 2 revolutions is:

$$P(2048) = \int_{0}^{1023} P(TL)\,d(TL)$$

and

$$P(1024) = 1-P(2048)$$

and

Average Job Time = Average Batch Time =

$$819.2\times P(1024)+1638.4\times P(2048).$$

Given a linear distribution from a maximum at TL = 1024 going to zero at TL = 0, P(1024) = 1/3 and P(2048) = 2/3 and Average Batch Time = 1365 μsecs.

More typical batches encountered in normal operation of the AMS system in the PLATO environment are those which contain more than one job. In the evaluation of these more complex jobs, additional factors need to be considered. In particular, since more than one subcontroller is operational, only the access time to the first job will be of concern, and the queuing for the AMS-ECS channel will be dominant.

The overall time required to execute a batch is determined by the sum of the access time to the first job, total time to transfer, and the average time to return the last job to the idle controller. Since it is assumed that there is no correlation between the idle address and the file starting addresses and that the starting addresses are randomly placed, the value of the first term can be calculated as a function of the number of jobs in a batch and the P(TL).

The distribution of access times to the first job for a uniform distribution of TL can be shown to be:

$$P(FSL) = \int_0^{1023} (TL/1024) \times n(1-TL/1024)\ d(TL)$$

and results in an access length with an average value of 1024/(n+1).

For non-uniform distributions of access times, the distribution of P(FSL) is:

$$P(FSL) = \int_0^{1023} P(TL) \times n \times (1-P(TL))\ d\ TL)$$

In the case of a uniform P(TL), the average difference between the access time of the first job and the access time of the second job, which determines the amount of data that must be transferred in order for the access time to the second job to be covered up is:

$$(1/n)-(1/(n+1))=1/n\times(n+1)).$$

For a batch using all eight subcontrollers, the average access to the first batch is 1024/9 or 113 records or 91 μsecs. The length from the access to the first job to the second job is 14.2 records or 113 words. In the PLATO system, the minimum usable data parcel is greater than 400 words, and therefore a very high probability exists that the transfer of the data from the first job will completely overlap the access to the second job.

Once the first job has been accessed, the ECS data channel is the time-determinant factor, and as such, the sum of all of the transfer lengths enters into the total batch time. After the last has finished with the data channel, it must be returned to its idle position and at this point the batch will be complete. There is no overlapping available for this return length; thus, the expected value is simply 512 cycles.

The total batch time therefore is as follows:

$$BT=(1024/(n+1))+(n\times\overline{TL})+512 \text{ cycles}$$

if the minimum transfer length is 1024/((n+1)×n). Table 5.1 shows expected

total batch time and minimum transfer lengths to maintain high AMS efficiency as a function of the n and $\overline{TL}$. These numbers assume that the distribution of transfer lengths is uniform, that the transfer starting address is uncorrelated to the idle address, and that the transfer addresses are randomly located.

Table 5.1
Batch Times and Minimum TL (records)

n	2	3	4	5	6	7	8
$\overline{TL}$[a]							
100	1053	1068	1116	1182	1258	1340	1425
200	1253	1368	1516	1682	1858	2040	2225
400	1653	1968	2316	2682	3058	3440	3825
600	2053	2568	3116	3682	4258	4840	5425
800	2453	3168	3916	4682	5458	6240	7025
1000	2853	3768	4716	5682	6658	7640	8625
minTL	170	51	34	24	18	14	11

[a] The total amount transferred is the $\overline{TL} \times n$.

The efficiency of the AMS controller can be seen by calculating an effective access time which is the difference between the time required to actually transfer the required data and the total batch time divided by the number of different files transferred.

$$\text{Effective Access Time (EAT)} = (BT-(TL \times n))/n$$

When the transfer lengths are subtracted, the effective access time is only a function of the number of subcontrollers active. See Table 5.2.

Table 5.2
EAT vs. n

n	2	3	4	5	6	7	8
EAT (cycles)	426	256	179	136	109	91	78
EAT (μsecs)	341	205	143	109	87	73	62

All of the above data applies only to the first batch of possibly several. In Chapter 2, the procedure for handling multiple batches is detailed. In this case, the time to process batches after the first is entirely limited by the transfer time since even the first access and return times are overlapped.

Two additional factors affect the overall performance of the AMS memory in the PLATO system: the possible saturation of the ECS data channel and the reduction of the data transfer rate of the AMS due to its low priority with respect to the CPU.

In a working PLATO system, the percentage of the available transfer bandwidth into ECS used up by the CPUs is relatively small. This situation prevents the severe degrading of the CPU performance since the CPUs are idle while such transfers take place. The amount of ECS-CPU transfer time measured in a system with 400 users active is approximately 18%. The extrapolation of this number to a 1000-active-user situation gives an estimate of 30% for CPU-ECS activity. Given this 30% requirement of the CPU, the measured fact that each user requires major CPU service on an average of once each four seconds, and an estimation of the data base required to service each user, the total utilization of the 600-megabit-per-second ECS channel can be calculated. It is estimated that each PLATO user will at most require 1) student status information (student bank) of 500 words, 2) lesson material of an average of 5000 words, 3) common data blocks of 1000 words and miscellaneous data blocks of less than 1000 words. In addition, the student bank, and perhaps the common data block, must be written back. The total data requirement would be less than 9000 words every four seconds. On the average only 200 words of lesson material will be read into central memory. One thousand users would require 250 processes

per second or 2.25 megawords per second at a maximum. This data flow represents 135 million bits per second or 22.5% of the ECS data channel. Between the AMS and the CPU, 52.5% of the data channel might be required but 57.5% would still remain for peak absorption and other uses.

As stated, it is estimated that 30% of the ECS data channel will be consumed in CPU transfers. The AMS controller assumes a low priority with respect to the CPU in order to insure that AMS transfers do not interfere with CPU transfers and thereby further degrade the CPU performance. However, this action does degrade the performance of the AMS controller, because only 70% of 600 megabits per second are available, so the time required to transfer data files will be correspondingly lengthened. Since, after the initial access to data is accomplished, the time required to process a batch is limited by the time required to transfer the data and not the access to files, the 30% reduction in the transfer rate will be readily apparent in the operation of AMS. In the calculations previously detailed, the access to first transfer and the time to return the last file to the idle controller are unchanged, but a 30% overhead must be added to the transfer time. In most batch mixes, approximately 25% will be added to the total batch time. The calculations concerning the ECS channel saturation remain unchanged, however.

Simulated batches were configured and submitted to the AMS controller to determine both the operation of the controller and the accuracy of the modeling used to evaluate the performance. Small jobs are the most difficult for the AMS controller to handle because the overhead is constant for all transfer lengths and therefore is most evident for short transfers. The student banks required when a user requests CPU activity by pressing a key is the smallest data field that PLATO would use; thus this file was used as

the source of jobs. A parasitic CPU program was run at a different control point from PLATO, at a higher priority than PLATO, and used the student bank list supplied by PLATO to assemble jobs for the AMS controller. File lengths of 512 words were used. Since the parasitic program was operating at a control point above PLATO, all of the time spent executing this logic detracted from the time available to PLATO. Various different loads were tried from slightly over 100 users to over 220 users, which resulted in a variation from five to eight jobs per poll on the average. The results of these tests are shown in Table 5.3.

Table 5.3
Empirical Data

Users	Jobs/Batch	ms/Batch Meas.[a]	ms/Batch Calc[b]
116	5.19	1.50	1.499
120	5.77	1.65	1.522
155	6.04	1.66	1.535
185	7.15	1.90	1.625
220	7.87	2.23	1.934

[a] ms/Batch Meas. is milliseconds per batch measured

[b] ms/Batch Calc. is milliseconds per batch calculated

The measured values were consistently slightly higher than the calculated values, which indicates some degradation due to the CPU-ECS transfers. If reasonable amounts of CPU time are assumed to determine the amount of ECS data channel bandwidth by the CPU, (5% at 116 users to 10% at 220 users) plus an additional 10% for the ECS bandwidth required by the parasitic test program, the results in Table 5.4 are obtained.

Table 5.4
Empirical Data vs. Calculated Data

Users	Jobs/Batch	ms/Batch M	ms/Batch C	Error
116	5.19	1.50	1.622	7.5%
120	5.77	1.65	1.659	.5%
155	6.04	1.66	1.688	1.7%
185	7.15	1.90	1.813	-4.8%
220	7.87	2.23	2.195	-1.6%

The average transfer per job is only 300 μsecs, which is short enough to make AMS usable even for short jobs.

Overall, the AMS controller proves itself to be entirely usable in the PLATO system even when operating with only short jobs and would be a usable addition to the memory hierarchy.

5.3 Memory Performance

Several problems were encountered in the assembly and testing of the memory portion of the AMS system. The principal problems were those normally associated with the manufacture of any large digital system: system noise and component reliability.

5.3.1 Memory Noise

In a bank of AMS memory, there are over 17,000 integrated circuits located on 82 printed circuit cards. These cards are interconnected for both power and signal purposes through a system of back-plane wiring and printed-circuit edge connectors. There are 88 connections (some of which are paralleled for power supplies) to each of the circuit cards. In addition, it is the nature of the MOS memory circuits, which by far dominated the number of circuits used (16,384 per bank), that they conduct the majority of their current only during transitions of the clock pulses. Because of this characteristic, the amount of current required of the power supplies is very irregular both on a short-term basis and a medium-term basis. The current required for an individual device is 1000 times as high during a clock transition than between these transitions. The current for an individual file is 20 times as high while it is in the active state (being searched, transferred, or returned by a subcontroller) than when it is in an inactive state (under the control of the idle controller). This large variation in the power supply current requirements causes noise problems.

In any power distribution system, some residual resistance exists between the power supply and the load, and the variation in the current causes a corresponding variation in the voltage drop between the supply and the load. In addition, in large power distribution systems, the inductance in the power cabling becomes significant and the speed of the change of the load current develops a voltage drop determined by the inductance. In anticipation of the above problems, the basic power supply distribution system utilized 1/2" × 1/2" aluminum stock as main busses and three #8 wires paralleled to interconnect chassis to the main buss. This power network proved to be adequate for long term (Current × Resistance) regulation.

Only very massive amounts of power bussing and very low impedence power supplies are capable of limiting the amount of power supply noise caused by the short-term variations in the power current. In general, the power distribution system alone is not capable of handling this noise, and additional capacitors (high frequency shunts) need to be added. In the original design of the AMS memory modules, more than ten 0.1 μfd capacitors were located on each circuit board, but after operation of the memory, an additional 16 capacitors needed to be added to each board to sufficiently reduce the power supply noise. Once a sufficient amount of capacity was available and it was properly distributed, the noise problems within the memory plane were overcome.

5.3.2 Device Failures

More than 17,000 integrated circuits are assembled to construct a single bank of AMS memory (262k words). Of these, one device comprises the bulk of 16,384 and a second is used in a quantity of 512. These two devices also accounted for almost all of the device failures encountered.

The modes of failure of the two devices were similar and took two forms: 1) immediate failure upon initial insertion, and 2) failure after several hours of operation. Failure percentages for the two devices are tabulated in Table 5.5.

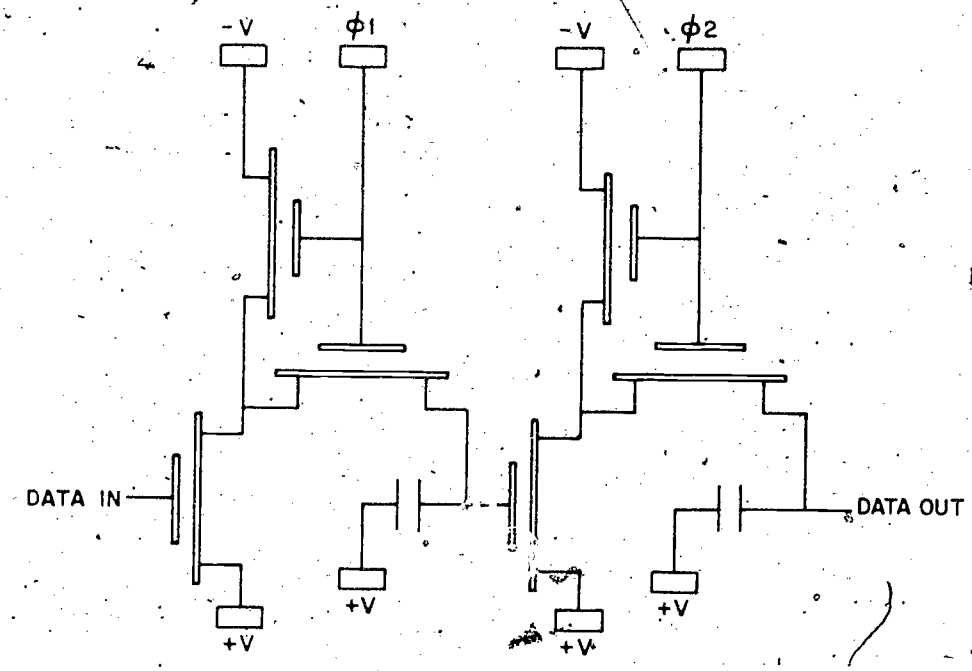
Table 5.5
Device Failure Statistics

Device	Usage	Failure (Initial)	Failure (Later)
MM5013	16,384	420 - 3%	610 - 4%
MH0026	512	48 - 9%	5 - 1%

5.3.3 Memory Reliability

Once the assembly and testing problems were overcome and the memory could be fully tested, the major concern was total reliability of the memory system. The AMS memory was designed to operate in the PLATO system, and since this system services almost 1000 users, it would be unsatisfactory for any memory element to fail except on very rare occasions. The memory section of the AMS memory did not prove to be capable of such a high level of reliability. The reason for this general volatility proved to be inherent in the operation of the P-channel MOS devices used.

The basic storage element within the P-Channel MOS dynamic shift register is a simple capacitor. Figure 5.1 is a schematic of this basic cell and shows a capacitor connected between the substrate and the gate of the transmission gates. This capacitor serves as the storage element and is refreshed every cycle of the clock, which also shifts the data of one cell (shown) into the next cell and shifts the data from the previous cell into this cell. The implementation of the capacitor is in the form of a reverse-biased diode junction, which has a capacity proportional to $V^{-2/3}$ and has a leakage current proportional to e^{T}, where T is the temperature of



P-CHANNEL MOS DYNAMIC MEMORY CELL

Figure 5.1

the junction. The amount of time that data is stored on a capacitor is a function of the capacity and the temperature. The manufacturer's specification guarantees that the device will maintain data over a frequency range from 0.01 MHz to 2.5 MHz, corresponding to a cycle time of from 100 μsecs to 0.4 μsecs over an ambient temperature of 0°C to 70°C. The AMS system limits the operation of the devices to a range of 100 μsecs to 0.8 μsecs. However, although the devices are rated to operate over this range, they will not reliably operate if the frequency is quickly varied from one extreme to the other. The reason for this failure is as follows.

Leakage through the storage capacitor is an exponential function of the termperature of the junction. At slow speeds, the temperature of the junction is very low (even though the ambient temperature might be relatively high) because the nearby transistors are not switching very fast and saturated MOS transistors dissipate the bulk of their power during the actual switching function. Since the capacitor leakage rate is low, long periods between refreshes can be tolerated, as is true during low-speed operation. At the other extreme, during high-speed operation, the temperature of the junction is quite high and the leakage is very high as well. However, the cell is refreshed very often by the clock and so data is maintained. However, when the data rate is instantly changed from a high rate to a low rate, the temperature of the cell does not change instantly, and for a short period of time the temperature of the junction is high but the refresh rate is low. During this time the margins are greatly reduced and an occasional bit is dropped.

This problem is very difficult to observe carefully and as such has been diagnosed by observation and through discussions with engineers of the

integrated circuit manufacturer. No mention of any problem of this sort is contained in the data sheets on the uses of the device.

CHAPTER 6

CONCLUSIONS

The purpose of this work was to discover and demonstrate a new method of controlling serially-organized memories in such a fashion as to make them usable as high-performance swapping memories in interactive computer systems. This purpose was accomplished.

In the coming years, the semiconductor industry will be introducing higher density, higher performance memories. It is my belief that as the density and speeds increase, optimum operation will be obtained by serial memories, because they are potentially simpler and because of their lower interconnect requirements. If effective means have been demonstrated for the operation of these memories, they will be more widely accepted, with the result being the availability of less expensive mass memory systems.

Recent strides in the technologies associated with Random Access Memories (RAM) have reduced the cost of these memories below even the parts costs of the P-Channel MOS devices used in this experiment. This development is one indicator of the enormous momentum in the semiconductor industry. It also shows that when the semiconductor industry attacks a problem in force, as it did in finding a device to compete with ferrite core memories, very remarkable advances can result. If a similarly intensive effort were placed in finding a product competitive with magnetic disk memory, a very inexpensive serial technology might emerge. CCD and Electron Beam memories might be the hint of the future technologies in this area. Once a really inexpensive device is available, techniques such as detailed in this report can be applied to further impact the mass memory market.

The details of this report do not totally define a mass memory system in its optimum form. Several modifications need to be added and more

investigation needs to be conducted. In particular, the data reliability problems that were encountered point directly to the need for at least single bit error correction being added. The fact that the ECS data-channel transfer times are the dominant factor in the performance of the system indicates that additional ports ought to be added to the memory controller. The addition of auxiliary ports would allow data from several sources to be simultaneously transferred into AMS. Much of the data transferred into AMS originates from sources other than the CPU. Disk transfers could be passed to AMS via these new ports. Implementation of these and other modifications to the AMS controller described in this report would result in an even more effective serial memory controller.

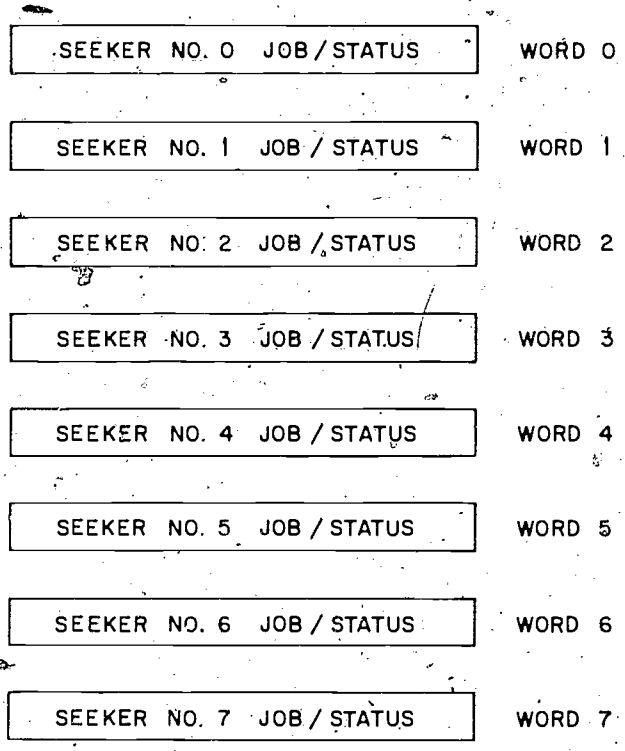
CHAPTER 7

APPENDIX

7.1 Control Word Formats

Following are the word formats of the 60-bit words used to communicate job parameters from the CPU to the AMS controller and the job status from the AMS controller to the CPU. This communication is conducted via an ECS communication area. In addition, the 12-bit word formats used by the supervisory PPU channel are shown.

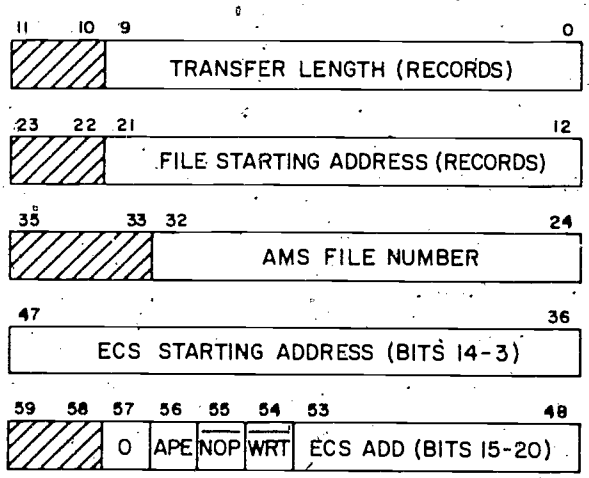
7.1.1 ECS COMMUNICATION RECORD FORMAT



(16 SUCH RECORDS ARE IN ECS)

Figure 7.1

7.1.2 CONTROL WORD FORMAT



AMS 60-BIT TRANSFER CONTROL WORD

Figure 7.2

Explanation of parameter field designations used in Figure 7.2:

Transfer Length: The number (10 bits) is the number of records (sets of eight words) to be transferred.

File Starting Address: This address (10 bits) is the address of the first record within the designated file that is to be transferred.

AMS File Number: This number (9 bits) is the number of the actual AMS file that is to be the source or sink for the transferred data.

ECS Starting Address: This field (18 bits) is the address of the first record within ECS that is to be the sink or source of the data transferred from AMS. All data is written or read from consecutive locations after this one.

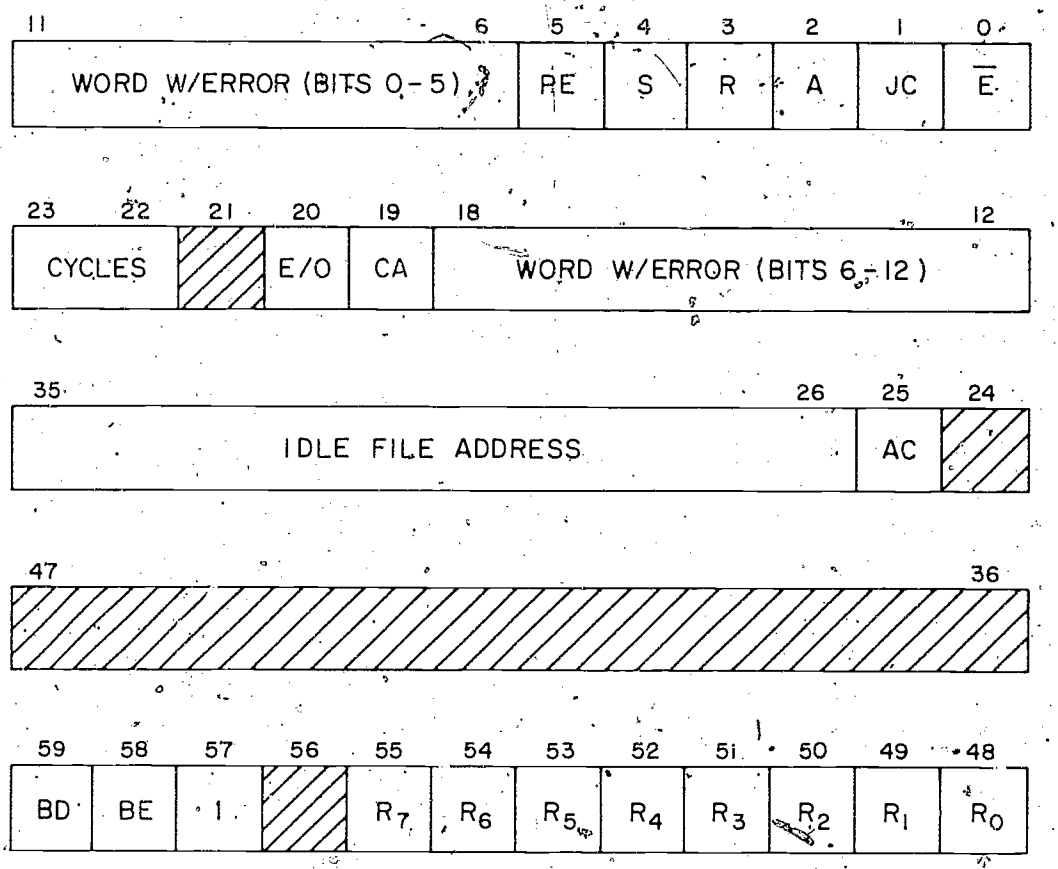
APE: Abort on parity error: This flag when set causes the AMS controller to abort a data transfer immediately if a read parity error is discovered. This flag applies only to read AMS operations because on an ECS read, the ECS controller automatically performs this operation.

NOP: No-Operation: A control word received by the AMS Controller with this bit cleared (set to "0") is interpreted identically as if it were all zeros. This provides the AMS with a "pass" or "fill" instruction.

WRT: Write: An AMS control word with this bit cleared will cause an AMS read operation (AMS to ECS data transfer). If this bit is set, an ECS read will result (ECS to AMS data transfer).

All words read by the AMS controller with bit 57 cleared are interpreted as Transfer Control Words. The AMS controller sets bit 57 to a "1" when it writes AMS Transfer Status words to ECS.

7.1.3 STATUS WORD FORMAT



AMS 60-BIT STATUS WORD

Figure 7.3

Explanation of terms and fields used in Figure 7.3:

E: Engaged: This bit is 0 if the seeker corresponding to this word number is presently engaged in the execution of a job. Otherwise it is a 1.

JC: Job Conflict: This bit indicates whether job conflict exists for this seeker. A job conflict is the condition whereby new job parameters are being supplied to a seeker which is already busy. This is not an error condition but rather indicates that the seeker in question has a new job to start when it completes the one it is presently performing.

A: Abort: This bit indicates that this seeker has aborted its job prematurely due to either a read error in AMS or an abort indication from the ECS controller.

R: Return: This bit indicates that this seeker is presently in its return phase of a job execution.

S: Search (Seek): This bit indicates that this seeker is presently in its search or seek phase of a job execution.

PE: Parity Error: This bit indicates that a parity error has been detected either by the ECS controller on an ECS read or by the AMS controller on an AMS read.

Word with Error: This field (13 bits) indicates the address of the word that flagged the parity error if PE is set during an AMS read operation. On an ECS read, only the top 10 bits are meaningful and indicate the AMS address present when the parity

error was sensed. The exact ECS address can be determined from this and the starting ECS address.

CA: Control Abort: This bit indicates that an abort was issued by the ECS controller during the last control read/write operation.

E/O: Even/Odd: The AMS controller writes the least significant bit of the batch number that an individual subcontroller obtained its job from in this location. When a status word is written by the AMS controller, any individual subcontroller might be either processing a job from that batch or from the next batch. By examining the batch number and the E/O bit, the CPU can determine which situation exists at that moment. This flag bit is intended as a maintenance and statistics-gathering aid and would not normally be examined by the CPU.

Cycle: The number of total memory cycles (records divided by 1024) that a seeker requires to complete a job (search, transfer, return) is indicated here. The seeker must always rotate a file an integer number of times from start to finish since the relative phase of the file must be the same (to line up with the idle address) at the end of job as it was at the beginning of that job. This information can be used in statistical studies of performance.

AC: Access Conflict: The setting of this bit indicates that this seeker has attempted to attach to a file that was already under the control of another seeker and thereby not attachable. The

seeker will wait until the file is detached by the other seeker and proceed from there.

Idle File Address: The exact relative rotational position of those files that are under the control of the Idle File Controller is indicated here. The CPU can use this information to minimize the search cycle if it assigns those jobs that it has to assign that would require the minimum rotational shift prior to transfer. Use of this feature would significantly reduce the access time to data but has no (or little) effect on the total job cycle.

P_7-P_0: Primary$_7$-Primary$_0$: These bits (the subscript corresponds to seeker numbers) indicate whether a seeker is presently conducting a primary job (the E/O bit is the same as the least significant bit of the ECS address that the control word was read from) or a secondary job (the E/O bit is different from the ECS address). Note: The ECS address in question is the record address only as the word address pertains to the individual seekers only.

BE: Batch Error: A "1" in this bit location indicates that the entire batch has been completed but that there was an error. The remainder of the status record will have to be examined to determine the nature of that error. If a seeker encounters an error in the execution of a job, it is not allowed to look ahead to the next batch for a new job, as to do so would destroy the error status information resulting from the error.

BD: Batch Done (without error): When this bit is set, the entire batch has been completed and therefore the CPU can depend on the data that was transferred. The CPU would typically look only at this bit to determine if a job has been completed. Once this bit has been set by the AMS controller, the CPU is free to rewrite the status information with new control information.

Bit 57 is always written to a 1 in an AMS status word.

7.2 Schematic Diagrams

Following are copies of the schematic diagrams of the AMS controller and memory system.

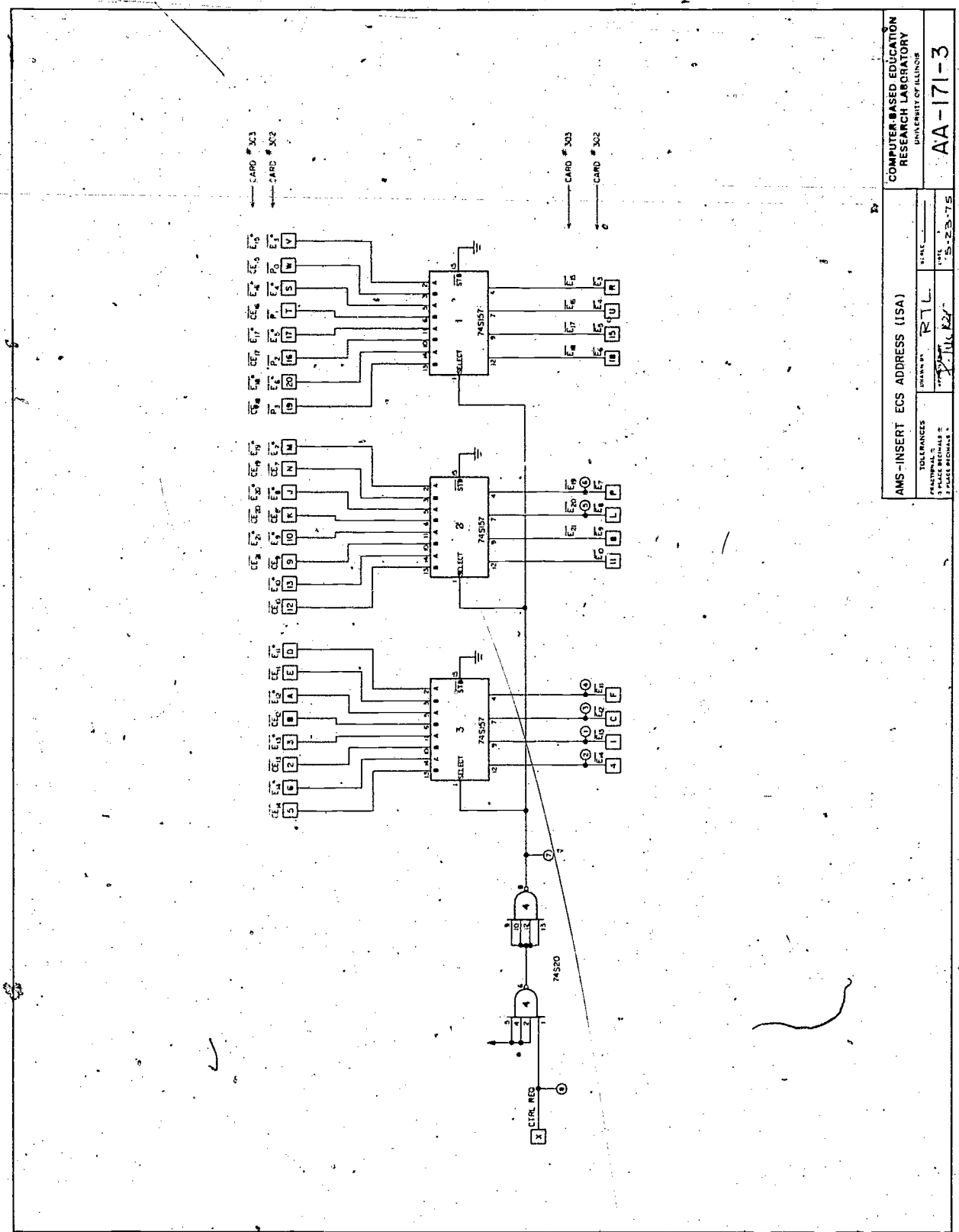

CARD # 303
CARD # 302
CARD # 303
CARD # 302
745157
745157
745157
74S20
CTRL REG
AMS-INSERT ECS ADDRESS (ISA)
COMPUTER-BASED EDUCATION RESEARCH LABORATORY
UNIVERSITY OF ILLINOIS
AA-171-3
RTL
5-23-75

Figure 7.4

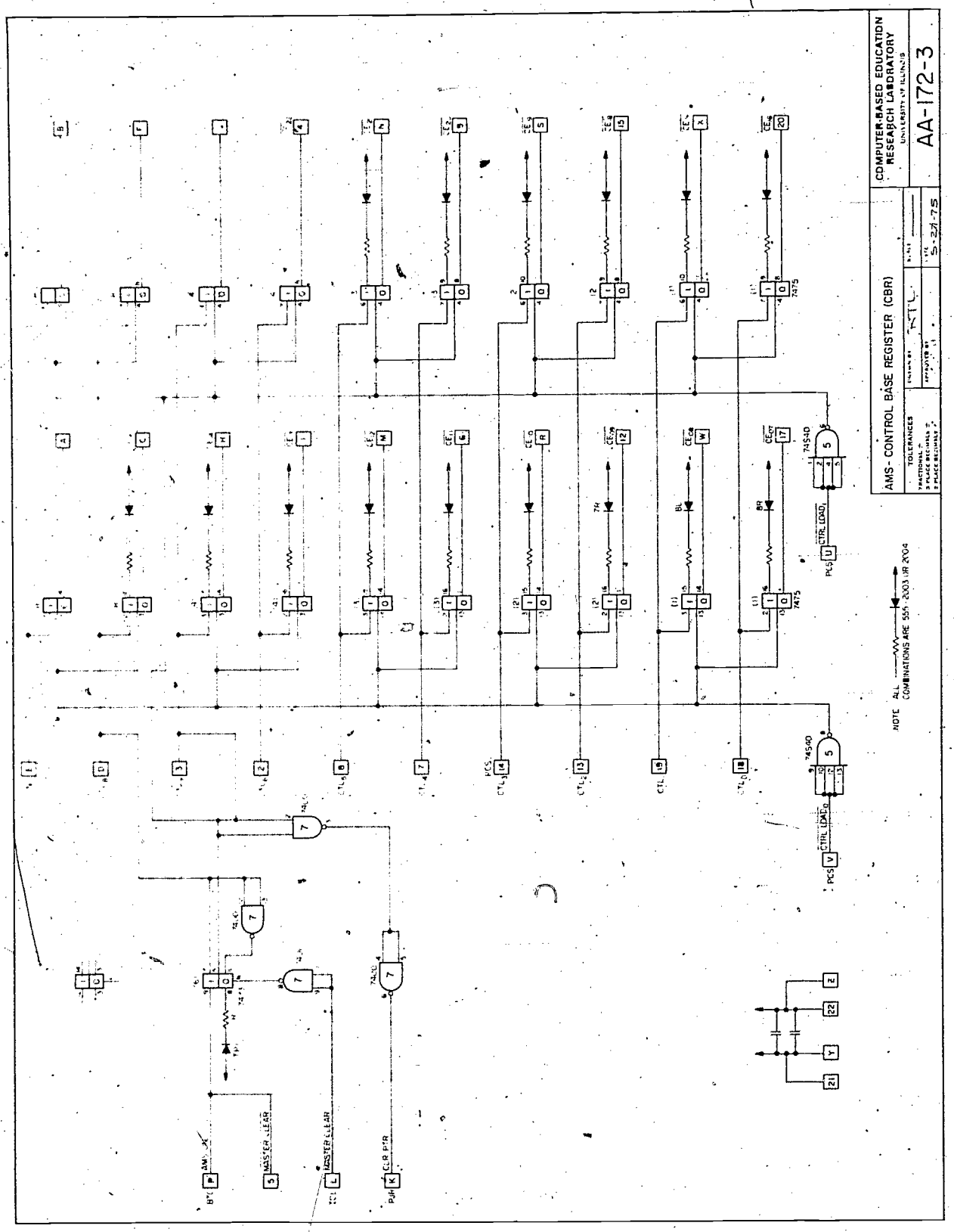

AMS - CONTROL BASE REGISTER (CBR)
COMPUTER-BASED EDUCATION RESEARCH LABORATORY
AA-172-3
S-24-75
NOTE ALL COMBINATIONS ARE 555-2003 OR 2004
CTRL LOAD
MASTER CLEAR
CLR PTR

Figure 7.5

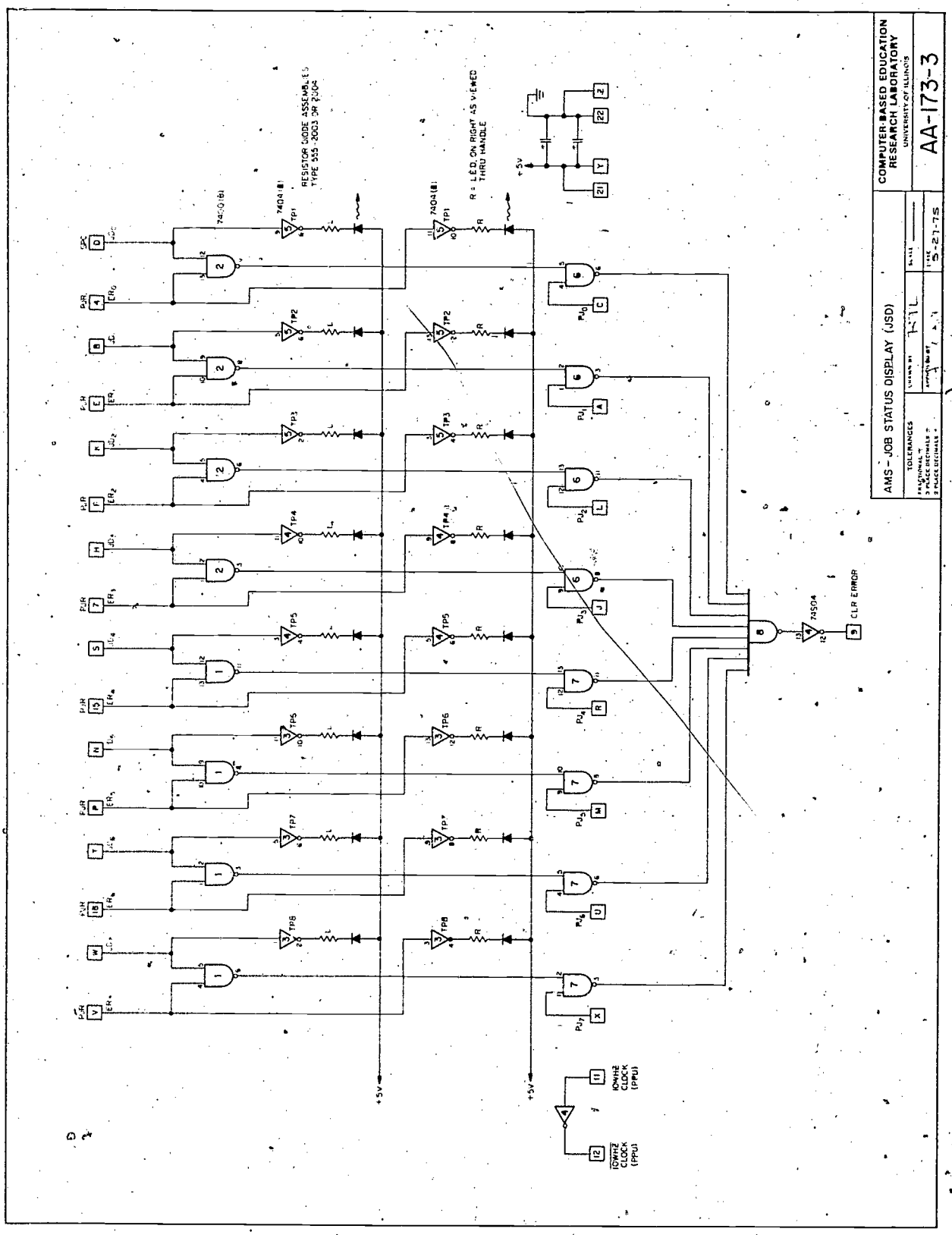

COMPUTER-BASED EDUCATION RESEARCH LABORATORY
UNIVERSITY OF ILLINOIS
AA-173-3
AMS - JOB STATUS DISPLAY (JSD)
TOLERANCES
5-27-75
RESISTOR CODE ASSEMBLIES TYPE 555-2003 OR 2004
R = L.E.D. ON RIGHT AS VIEWED THRU HANDLE
7404 (8)
74S04
CLR ERROR
+5V
10MHZ CLOCK (PPU)
10MHZ CLOCK (PPU)

Figure 7.6

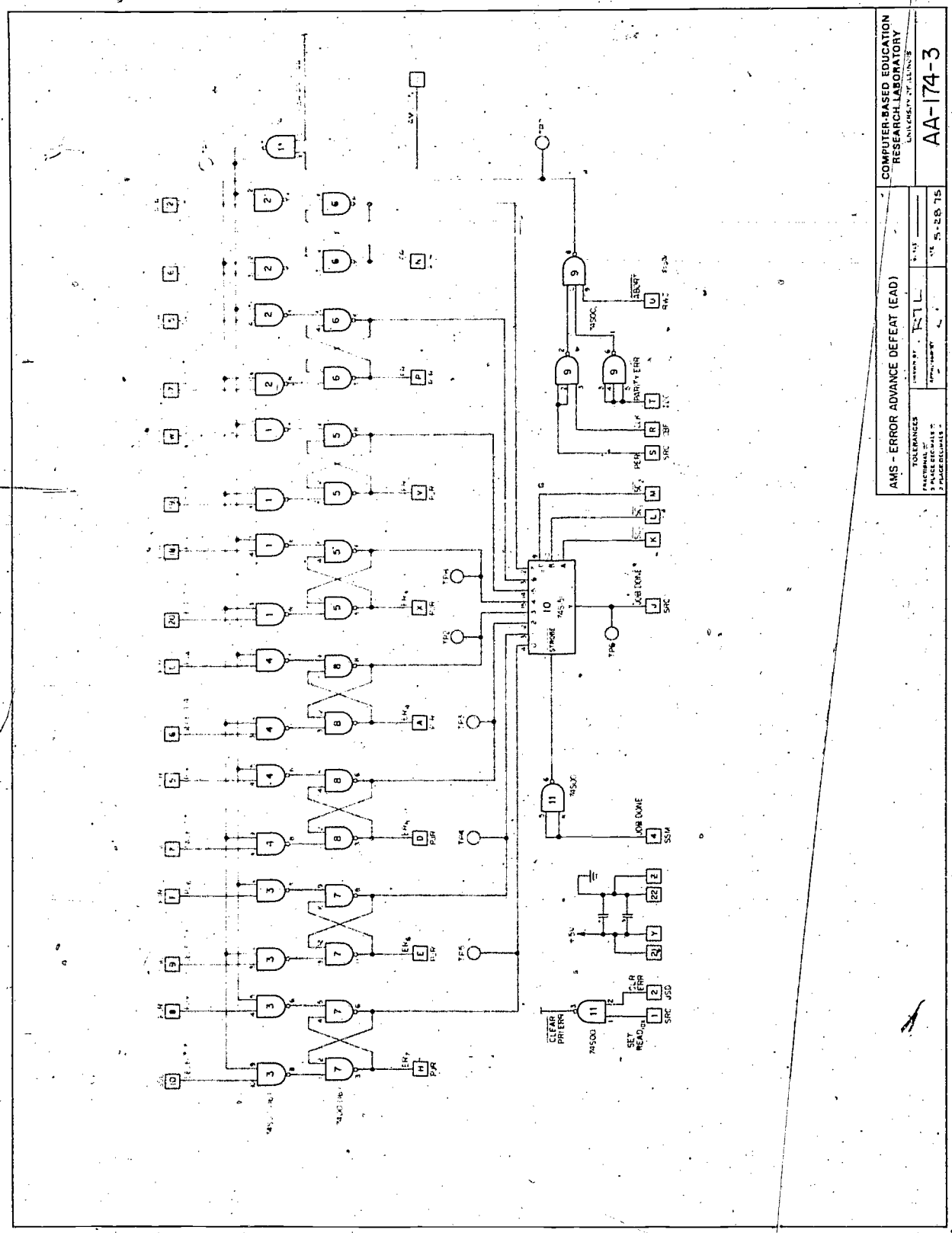

AMS - ERROR ADVANCE DEFEAT (EAD)
COMPUTER-BASED EDUCATION RESEARCH LABORATORY
AA-174-3
TOLERANCES
5-28-75
PARITY ERR
CLEAR PRI ERR
SET READ
LOW DONE
+5V

Figure 7.7

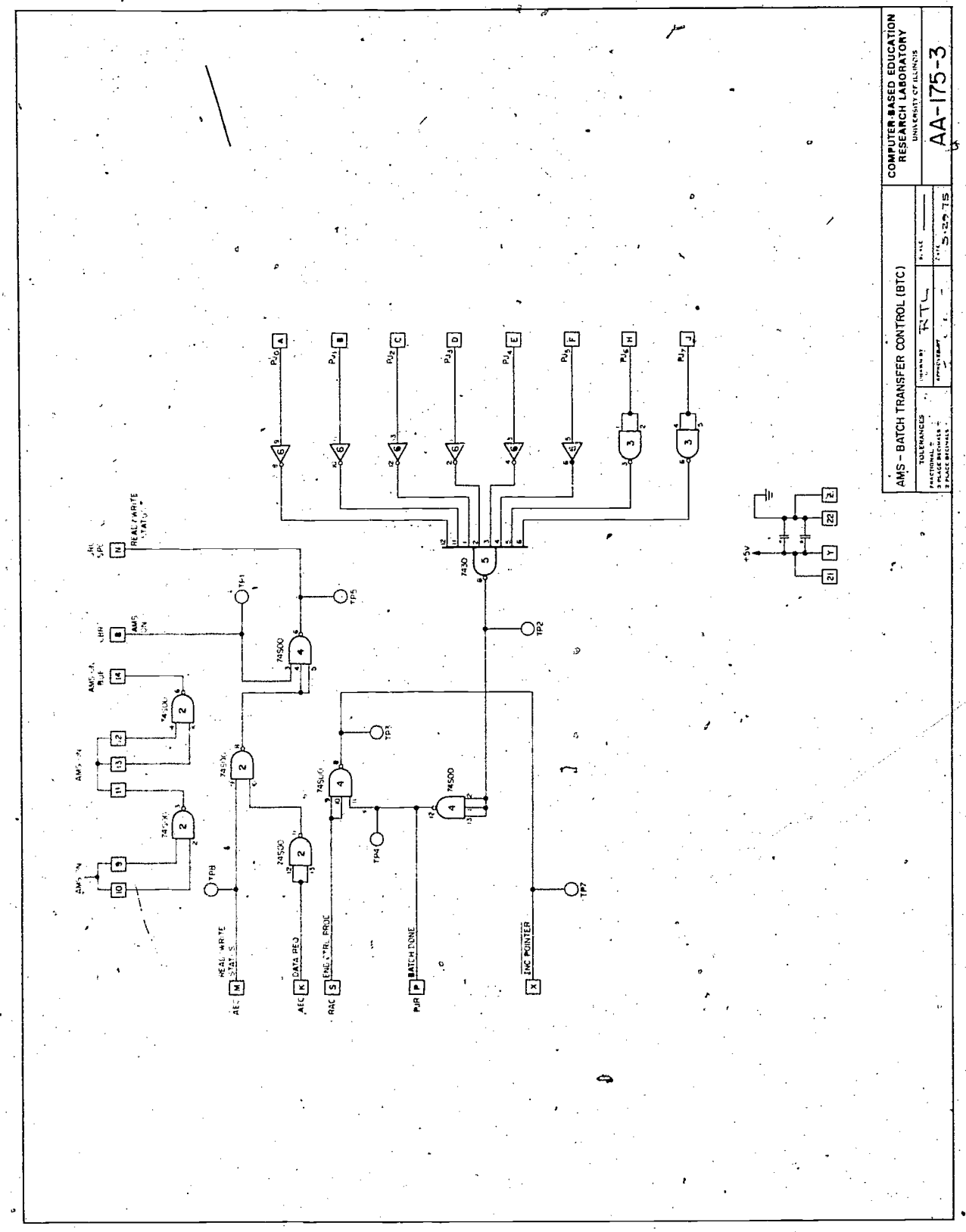

AMS - BATCH TRANSFER CONTROL (BTC)
COMPUTER-BASED EDUCATION RESEARCH LABORATORY
UNIVERSITY OF ILLINOIS
AA-175-3
TOLERANCES
DRAWN BY RTL
5-29-75
7430
74500
TP1
TP2
TP3
TP4
TP5
TP6
TP7
TP8
+5V
DATA REQ
ENG CNTRL PROC
BATCH DONE
INC POINTER

Figure 7.8

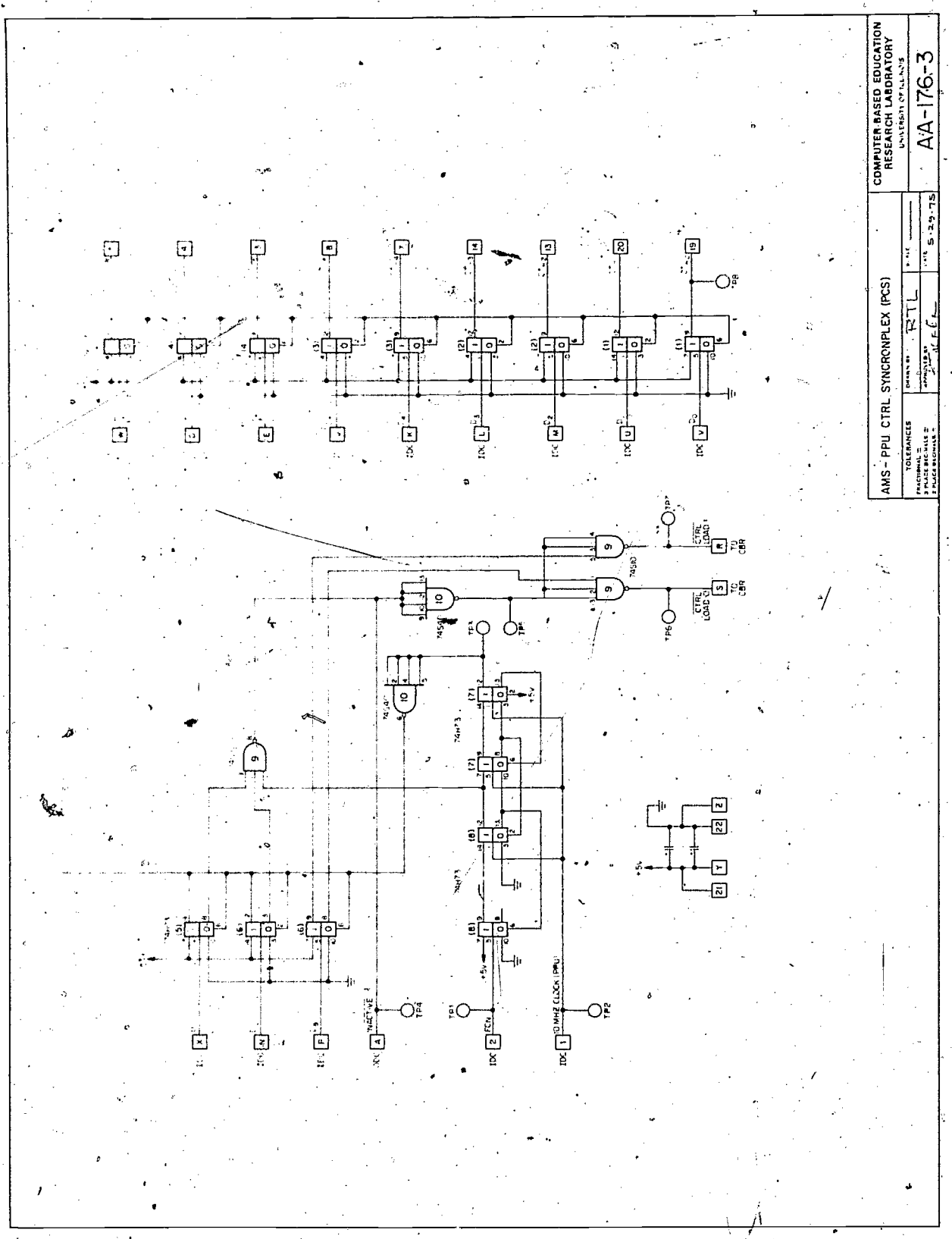

AMS - PPU CTRL SYNCRONPLEX (PCS)
COMPUTER BASED EDUCATION RESEARCH LABORATORY
A/A-176-3
RTL
TOLERANCES
CTRL LOAD 1
CTRL LOAD 0
TO CBR
10 MHZ CLOCK (PPU)
+5v
TP1
TP2
TP3
TP4
TP5
TP6

Figure 7.9

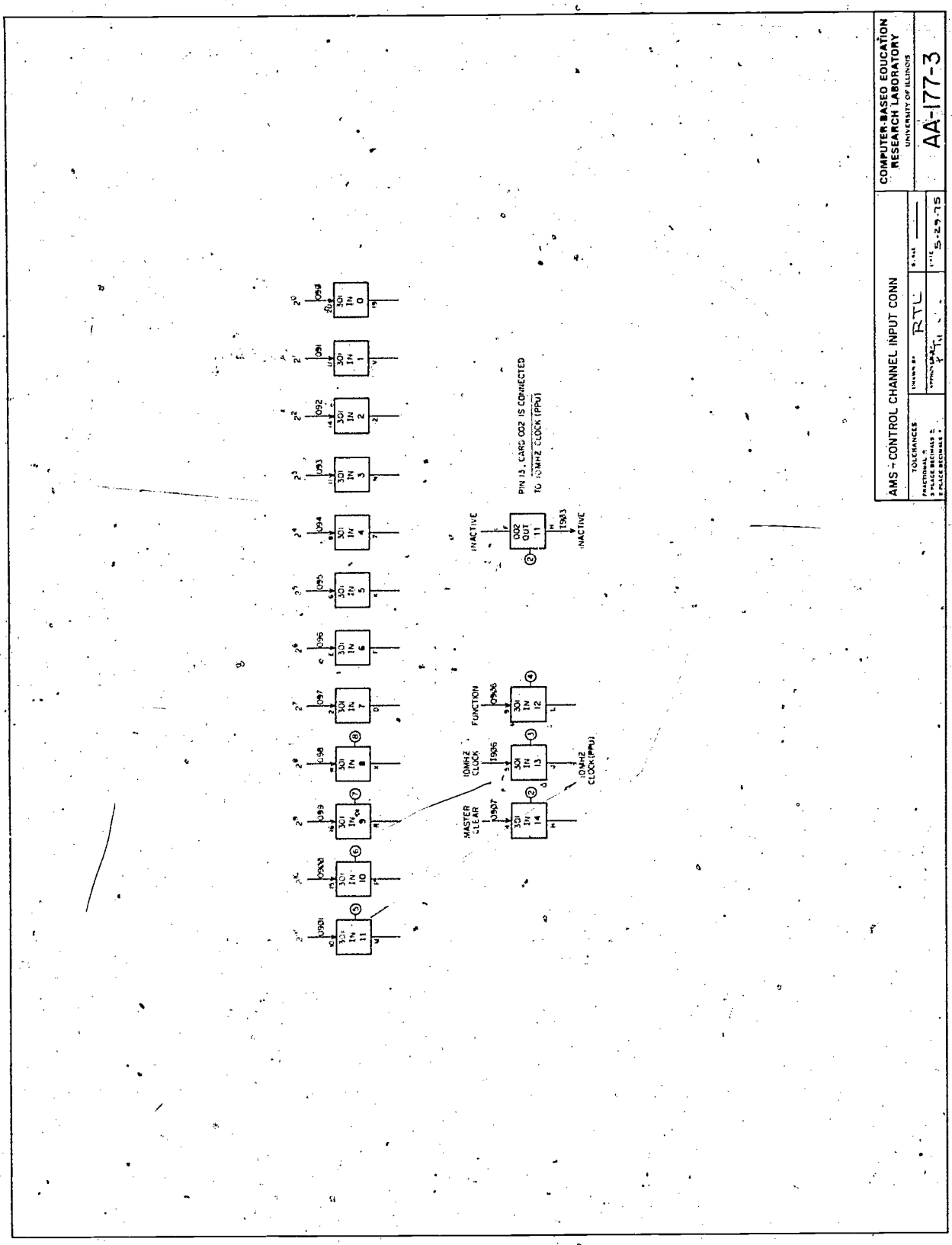

AMS - CONTROL CHANNEL INPUT CONN
COMPUTER-BASED EDUCATION RESEARCH LABORATORY
UNIVERSITY OF ILLINOIS
AA-177-3
5-29-75
RTL
INACTIVE
PIN 13, CARD 002 IS CONNECTED TO 10MHZ CLOCK (PPU)
FUNCTION
10MHZ CLOCK
MASTER CLEAR
10MHZ CLOCK (PPU)

Figure 7.10

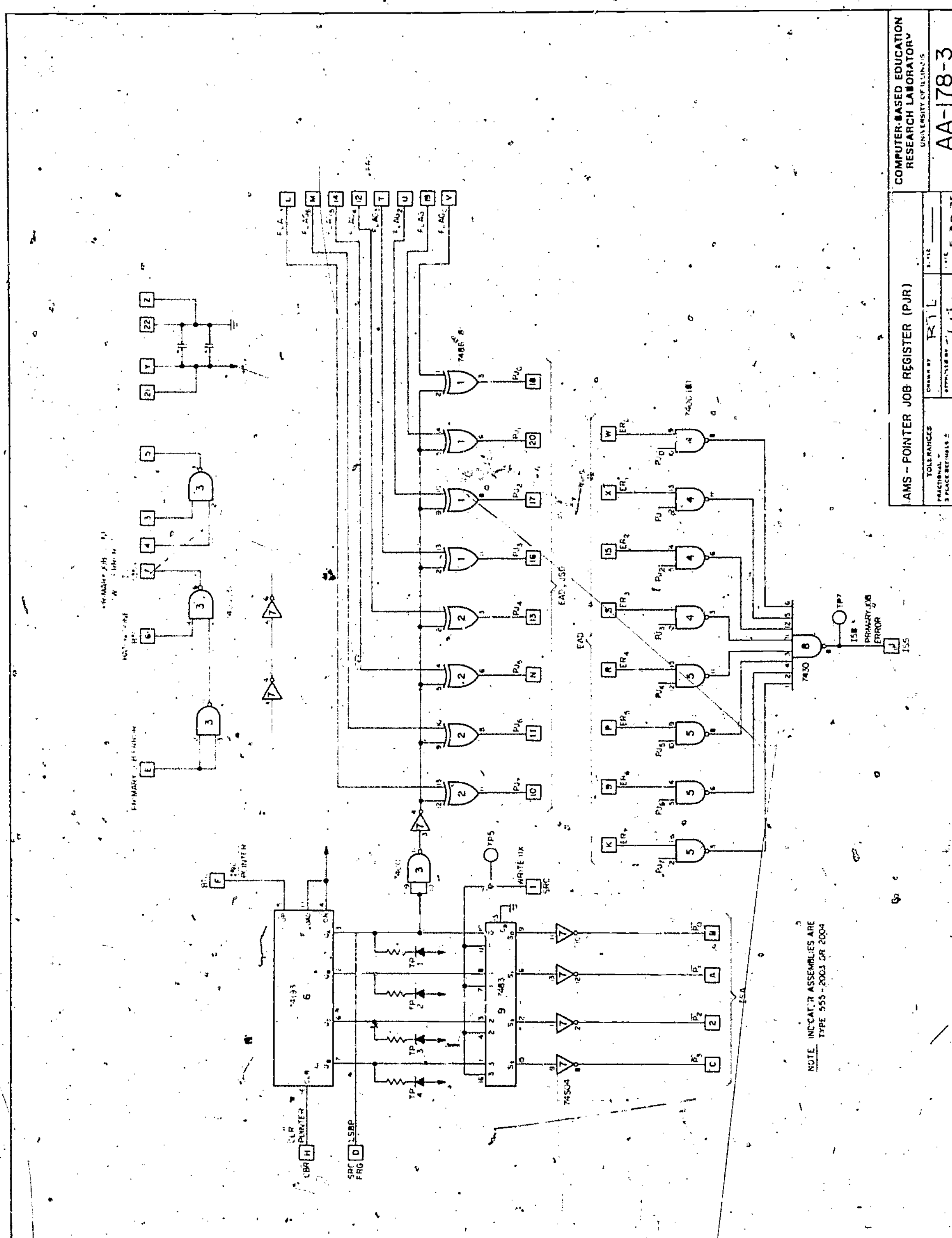

COMPUTER-BASED EDUCATION RESEARCH LABORATORY
UNIVERSITY OF ILLINOIS
AA-178-3
AMS - POINTER JOB REGISTER (PJR)
NOTE INDICATOR ASSEMBLIES ARE TYPE 555-2003 OR 2004
PRIMARY JOB ERROR

Figure 7.11

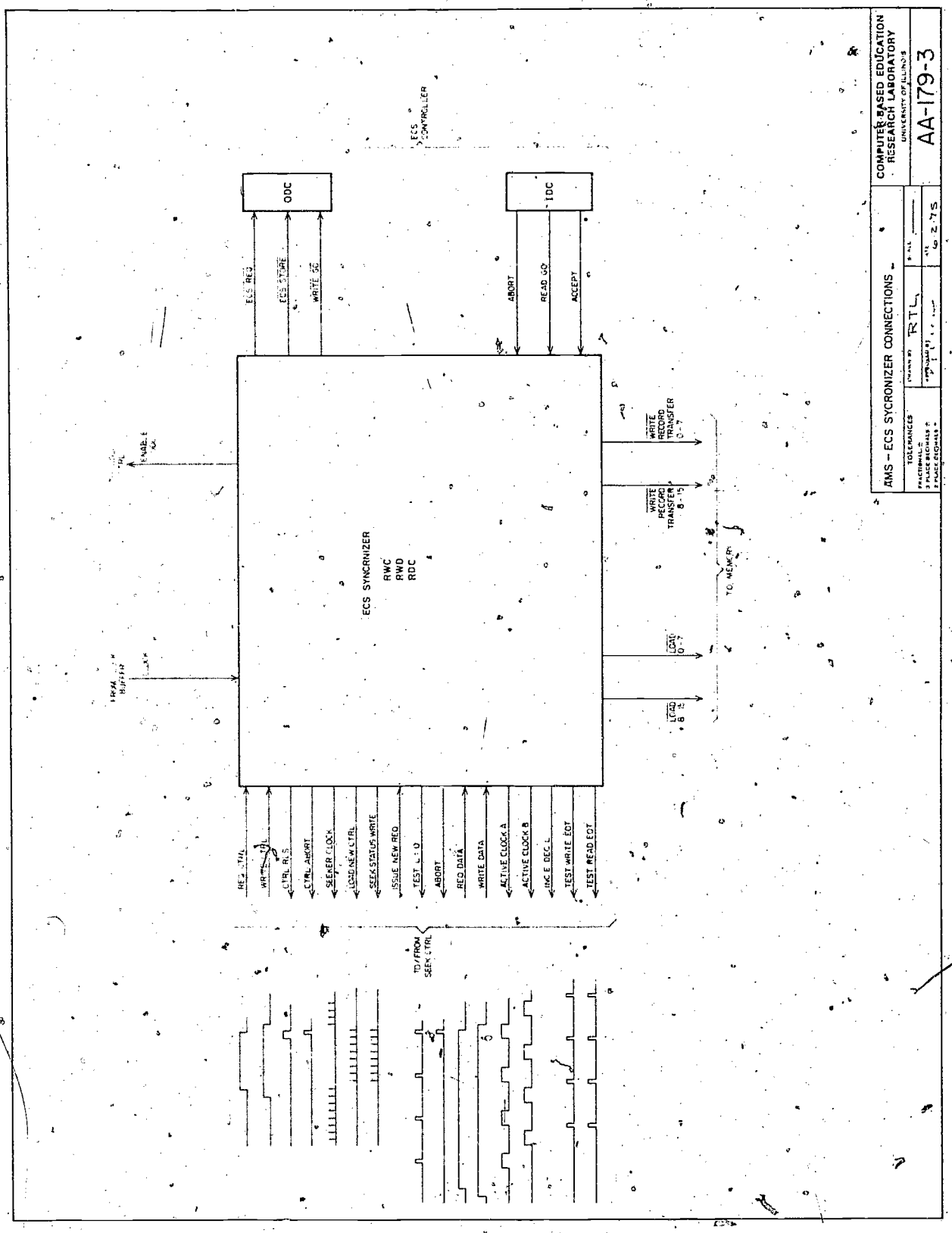

ECS CONTROLLER
ODC
IDC
ECS REQ
ECS STROBE
WRITE DC
ABORT
READ GO
ACCEPT
ECS SYNCRONIZER
RWC
RWD
RDC
WRITE RECORD TRANSFER 0-7
WRITE RECORD TRANSFER 8-15
TO MEMORY
LOAD 0-7
LOAD 8-15
REQ CTRL
WRITE CTRL
CTRL RLS
CTRL ABORT
SEEKER CLOCK
LOAD NEW CTRL
SEEK STATUS WRITE
ISSUE NEW REQ
TEST LOAD
ABORT
REQ DATA
WRITE DATA
ACTIVE CLOCK A
ACTIVE CLOCK B
INCR DECR
TEST WRITE EOT
TEST READ EOT
TO/FROM SEEK CTRL
AMS - ECS SYCRONIZER CONNECTIONS
COMPUTER BASED EDUCATION RESEARCH LABORATORY
UNIVERSITY OF ILLINOIS
AA-179-3

Figure 7.12

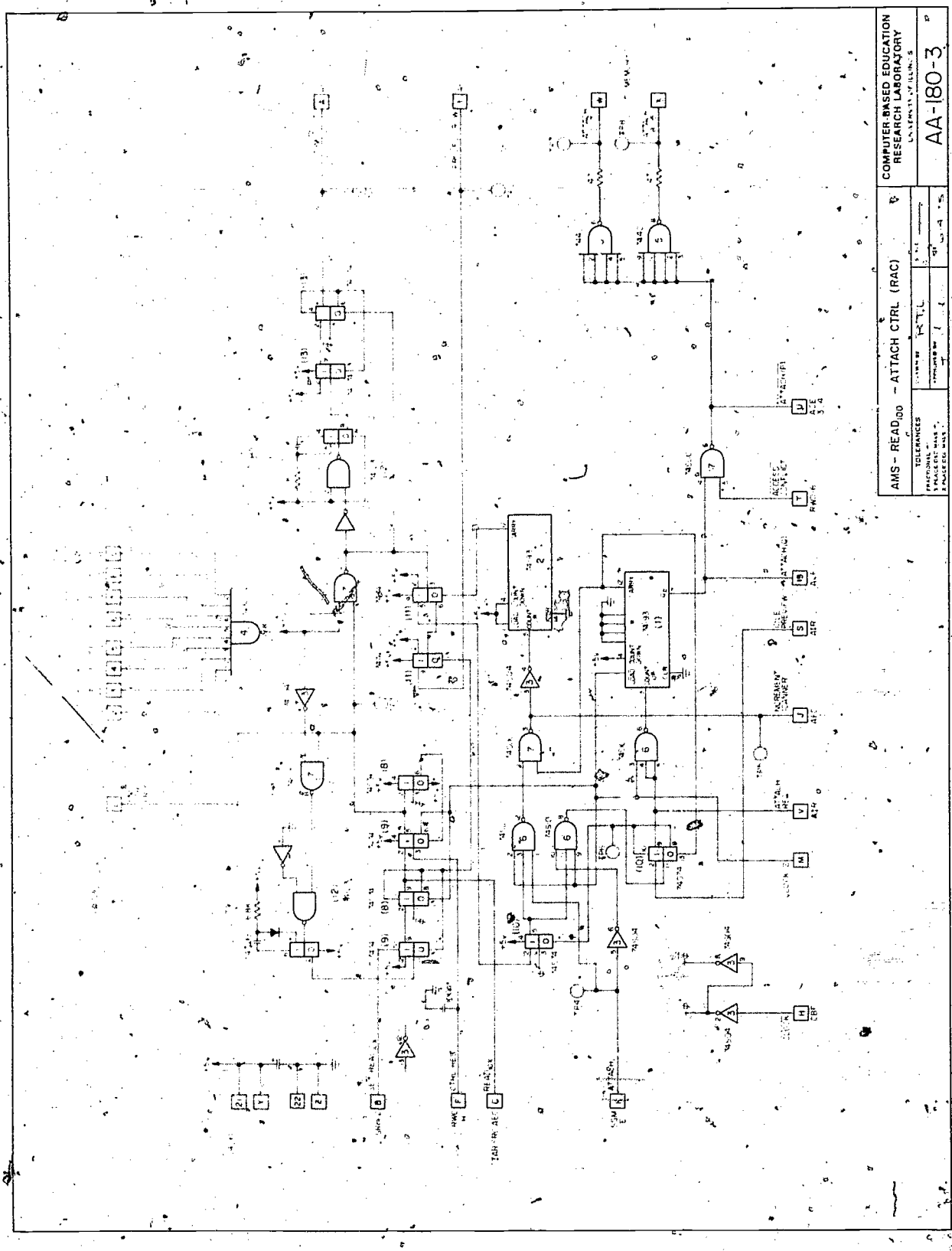

COMPUTER-BASED EDUCATION RESEARCH LABORATORY
AA-180-3
AMS - READ100 - ATTACH CTRL (RAC)
TOLERANCES

Figure 7.13

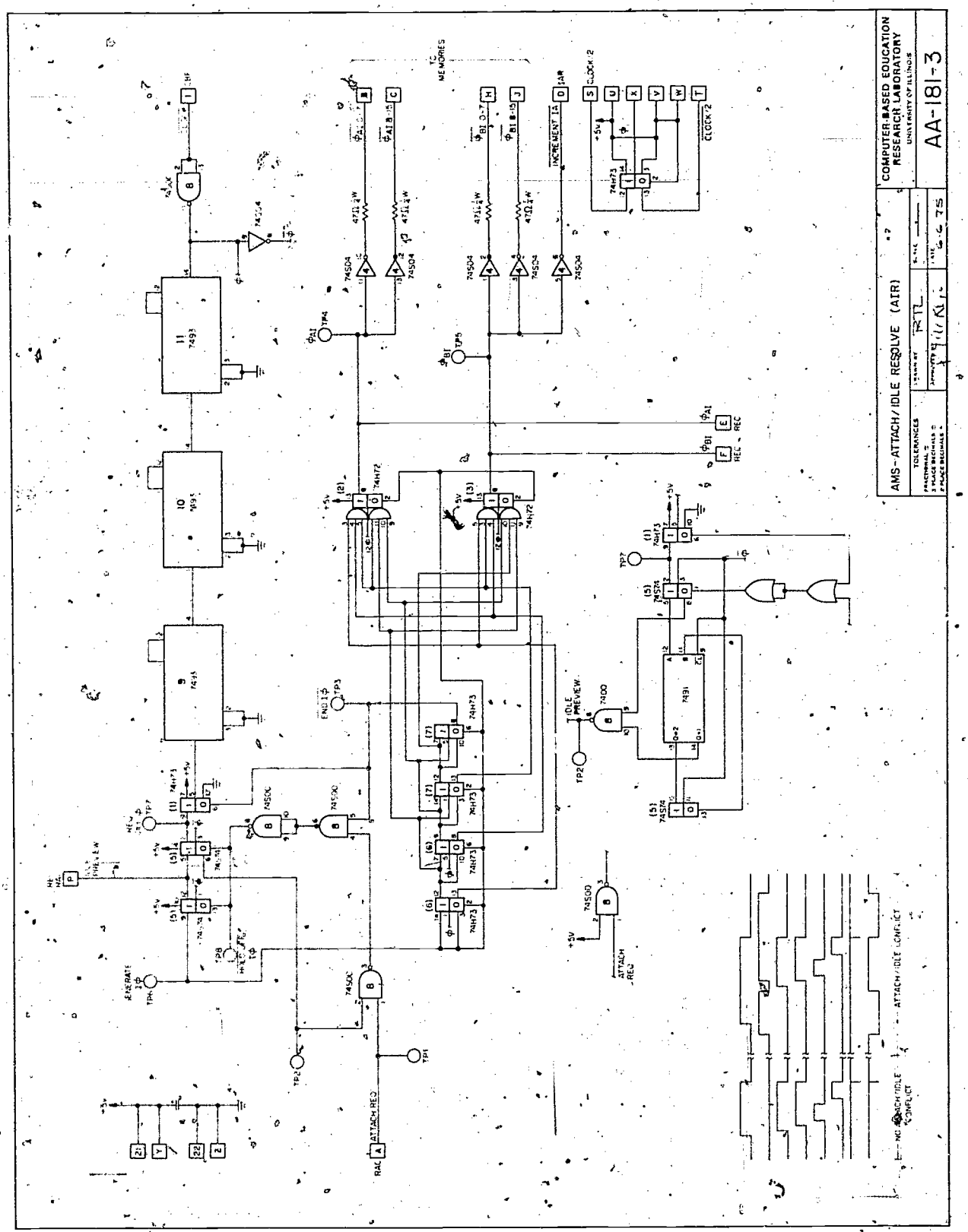

AMS-ATTACH/IDLE RESOLVE (AIR)
COMPUTER-BASED EDUCATION RESEARCH LABORATORY
UNIVERSITY OF ILLINOIS
AA-181-3
TOLERANCES
FRACTIONAL ±
2 PLACE DECIMALS ±
3 PLACE DECIMALS ±
DRAWN BY
APPROVED
DATE
6-6-75
IC MEMORIES
INCREMENT IA
CLOCK*2
ATTACH REQ
IDLE PREVIEW
GENERATE
ATTACH/IDLE CONFLICT
NO ATTACH/IDLE CONFLICT

Figure 7.14

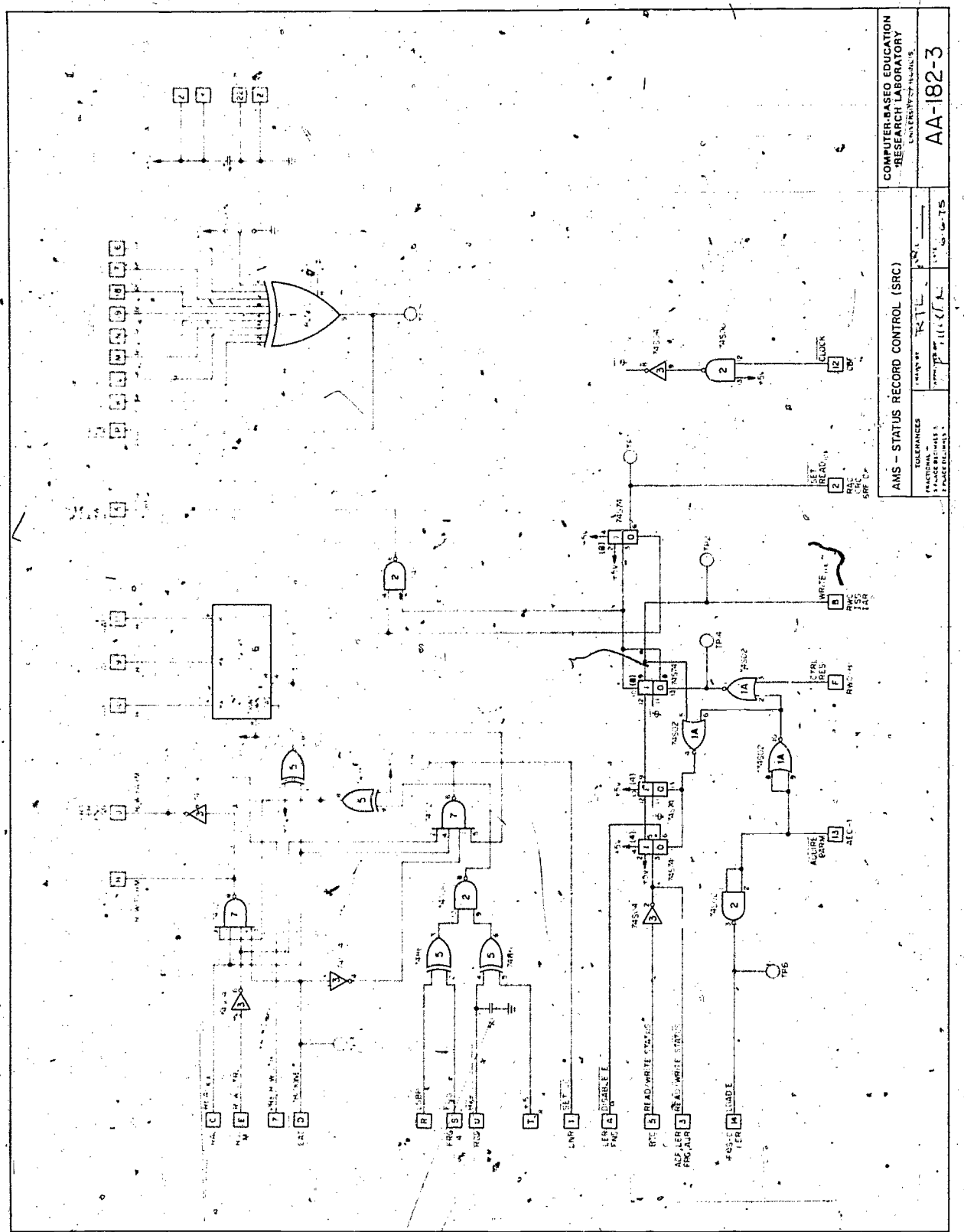

COMPUTER-BASED EDUCATION RESEARCH LABORATORY
AA-182-3
AMS – STATUS RECORD CONTROL (SRC)
TOLERANCES
CLOCK
READ/WRITE
LOAD

Figure 7.15

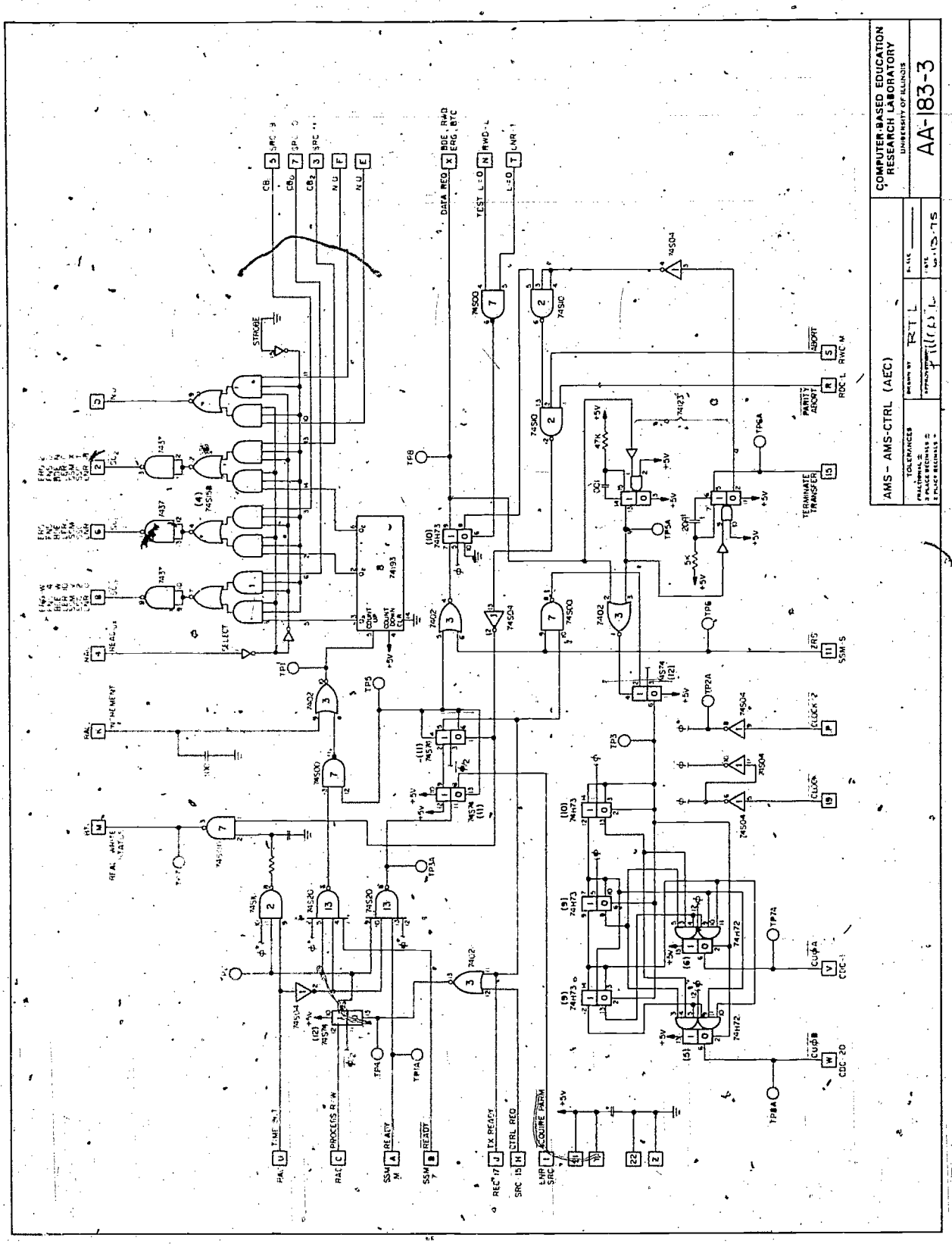

AMS - AMS-CTRL (AEC)
COMPUTER-BASED EDUCATION RESEARCH LABORATORY
UNIVERSITY OF ILLINOIS
AA-183-3
DRAWN BY RTL
DATE 10-15-75
DATA REQ
TEST L=0
L=0
STROBE
SELECT
COUNT DOWN
TERMINATE TRANSFER
PARITY ADDR
ABORT
CLOCK 2
CLOCK
REAL TIME SYNC
PROCESS R/W
SSM READY
TX READY
CTRL REQ
ACQUIRE PERM
CDC-1
CDC-20

Figure 7.16

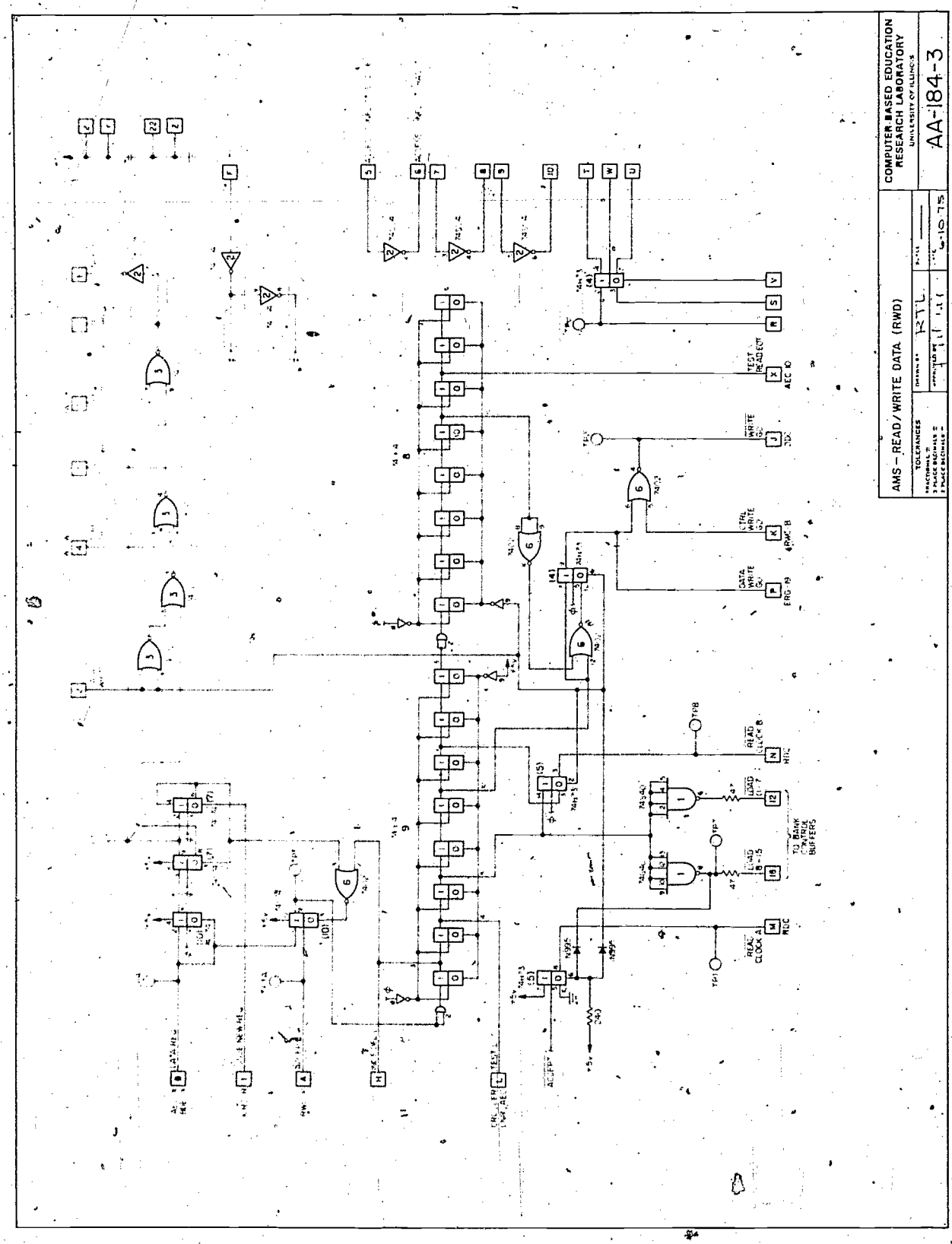

COMPUTER-BASED EDUCATION RESEARCH LABORATORY
UNIVERSITY OF ILLINOIS
AA-184-3
AMS – READ/WRITE DATA (RWD)
TOLERANCES
DRAWN BY
RTL
6-10-75
TEST READ OUT
AEC 10
WRITE
CTRL WRITE
DATA WRITE
READ CLOCK B
READ CLOCK A
LOAD
TO BANK CONTROL BUFFERS

Figure 7.17

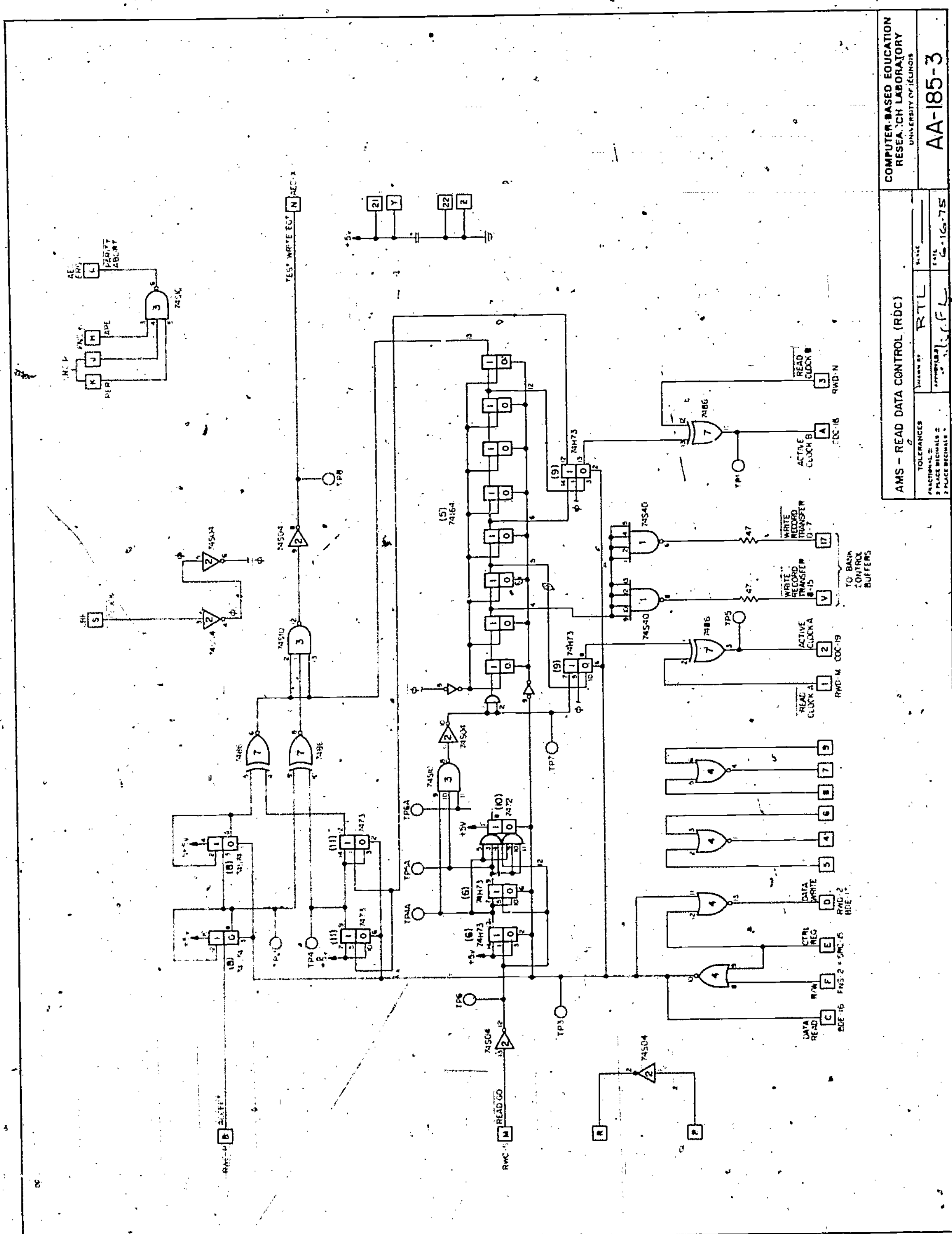

COMPUTER-BASED EDUCATION
RESEARCH LABORATORY
UNIVERSITY OF ILLINOIS
AA-185-3
AMS - READ DATA CONTROL (RDC)

Figure 7.18

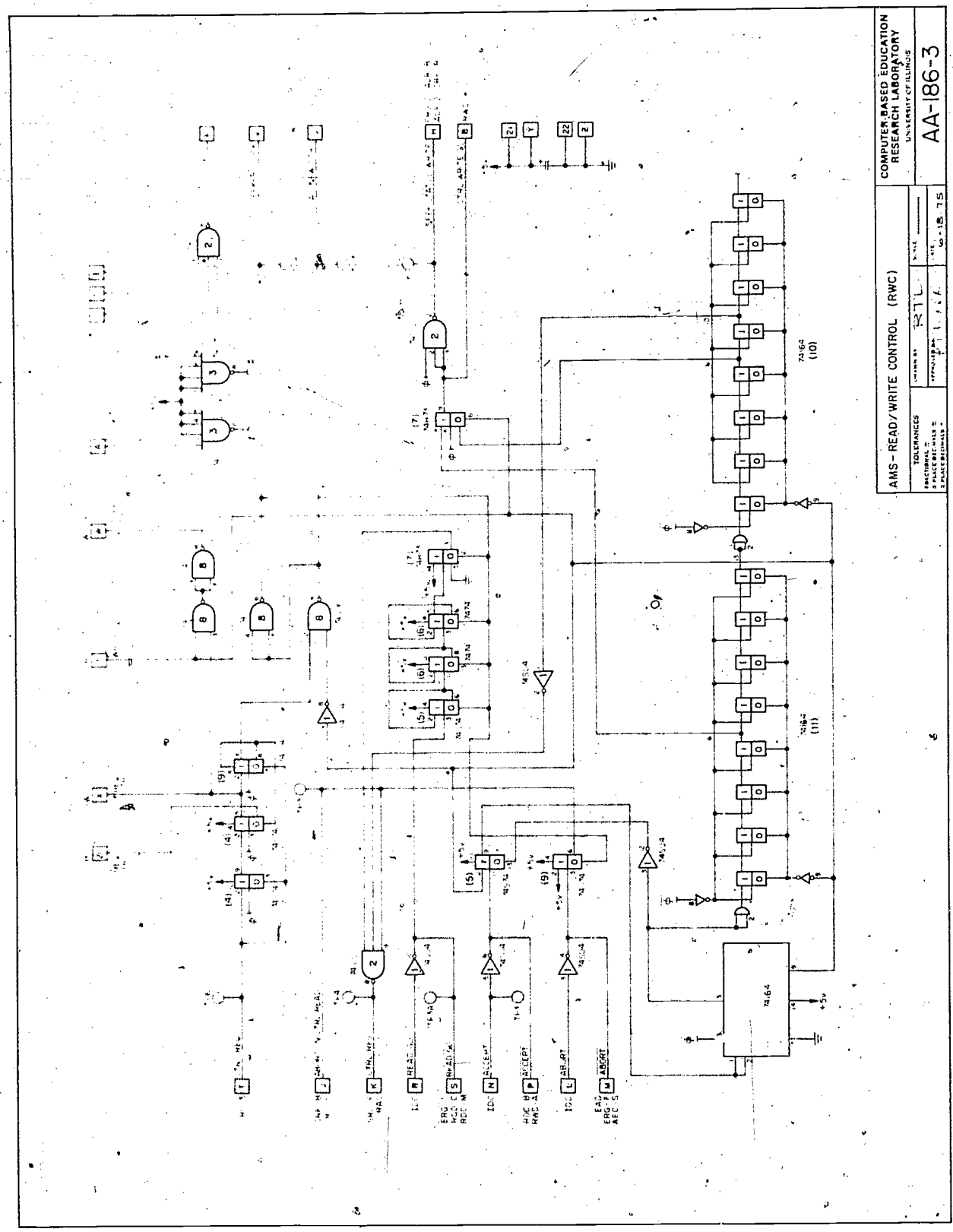

AMS - READ/WRITE CONTROL (RWC)
COMPUTER-BASED EDUCATION RESEARCH LABORATORY
UNIVERSITY OF ILLINOIS
AA-186-3
74164 (10)
74164 (11)
74164
HEAD
ABSENT
ACCEPT
ABORT

Figure 7.19

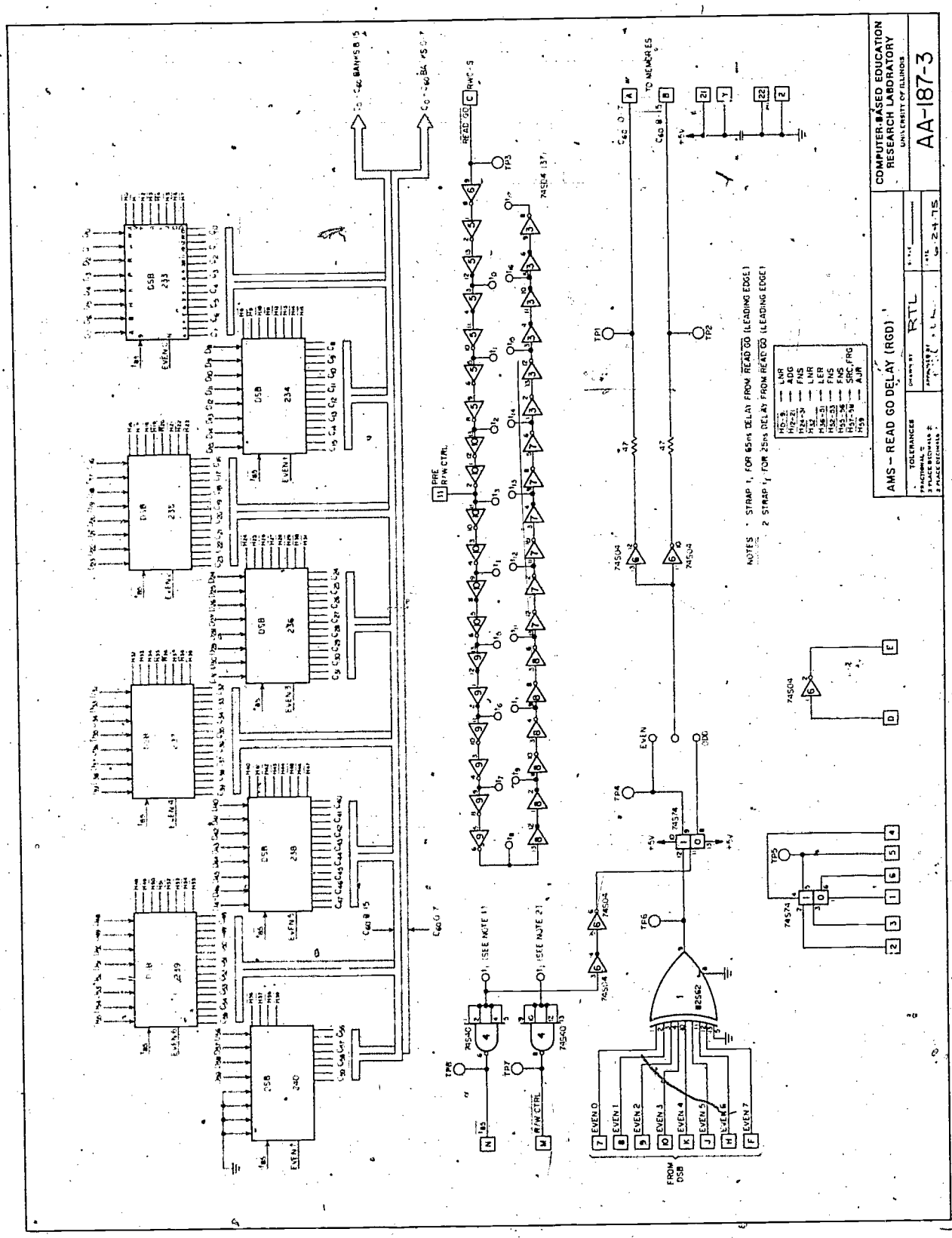

AMS – READ GO DELAY (RGD)
COMPUTER-BASED EDUCATION RESEARCH LABORATORY
UNIVERSITY OF ILLINOIS
AA-187-3
RTL
TOLERANCES
NOTES: 1. STRAP t_1 FOR 65ns DELAY FROM READ GO (LEADING EDGE)
2. STRAP t_1 FOR 25ns DELAY FROM READ GO (LEADING EDGE)
TO MEMORIES
FROM DSB
EVEN 0
EVEN 1
EVEN 2
EVEN 3
EVEN 4
EVEN 5
EVEN 6
EVEN 7
READ GO
R/W CTRL
PRE R/W CTRL
LNR
ADG
FNS
LNR
LER
FNS
FNS
SRC FRG
AJR

Figure 7.20

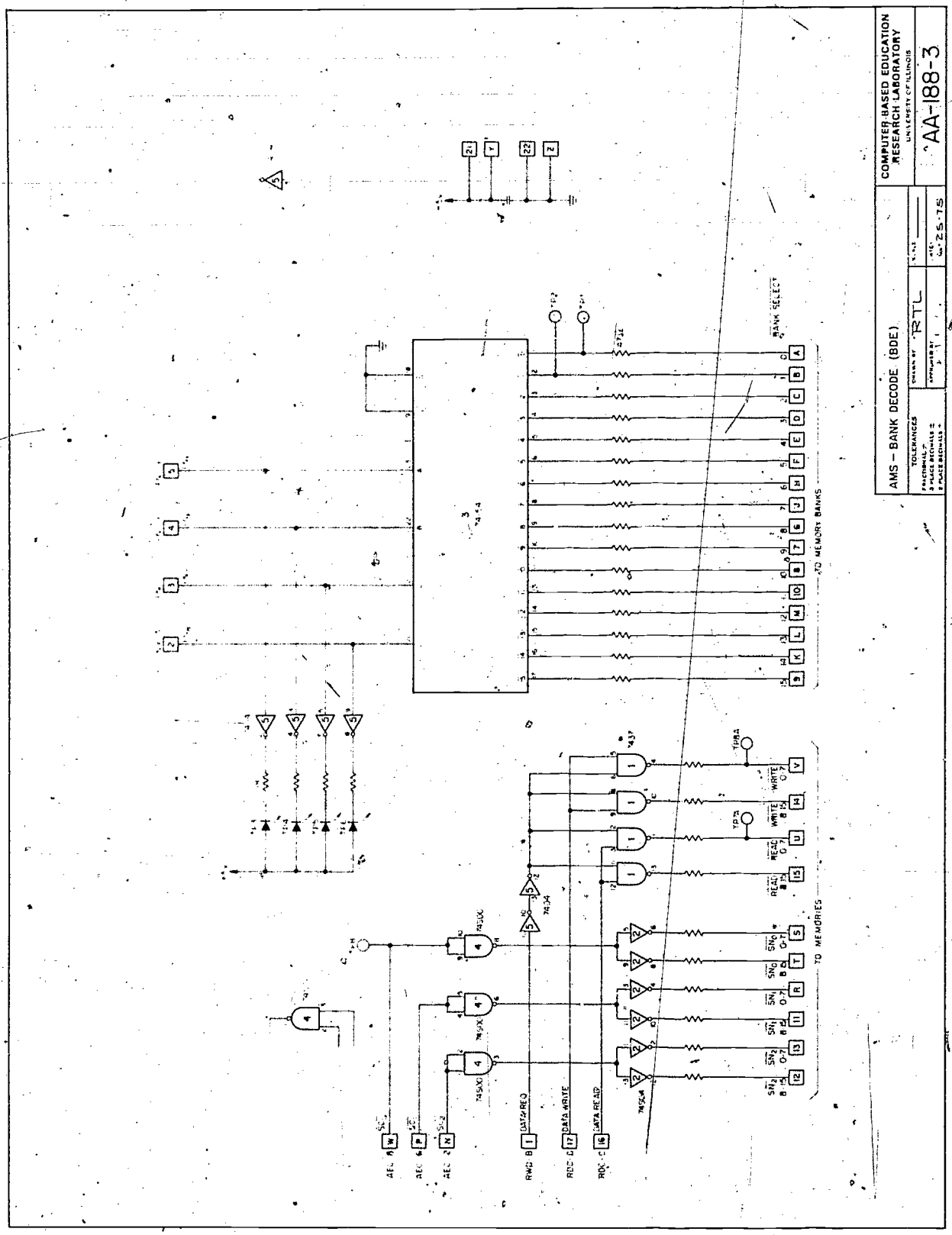

AMS – BANK DECODE (BDE)
COMPUTER-BASED EDUCATION RESEARCH LABORATORY
UNIVERSITY OF ILLINOIS
AA-188-3
RTL
BANK SELECT
TO MEMORY BANKS
TO MEMORIES
DATA READ
DATA WRITE

Figure 7.21

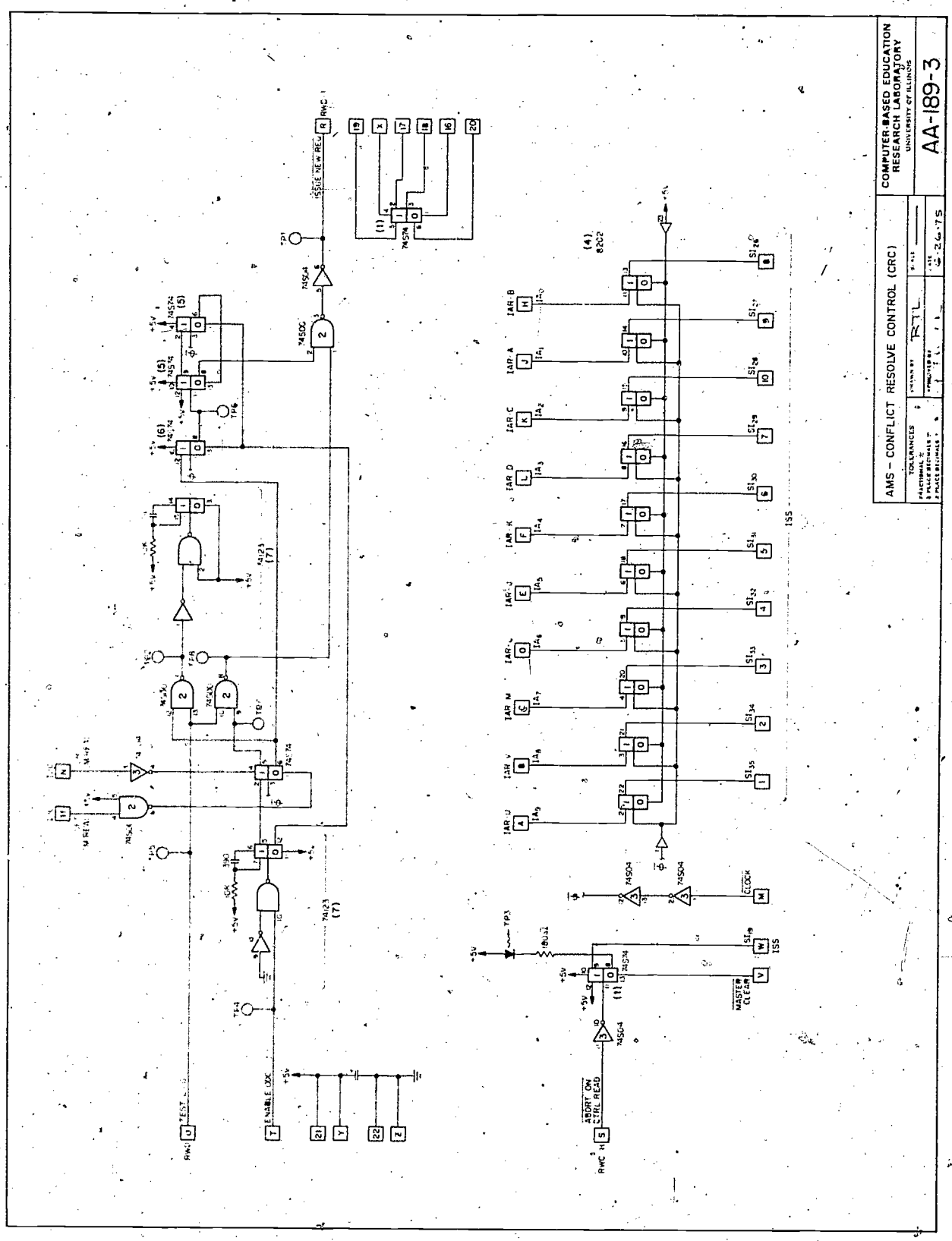

AMS – CONFLICT RESOLVE CONTROL (CRC)
COMPUTER-BASED EDUCATION RESEARCH LABORATORY
UNIVERSITY OF ILLINOIS
AA-189-3
RTL
ISSUE NEW REQ
ENABLE DXX
TEST
ABORT ON CTRL READ
MASTER CLEAR
CLOCK
IAR B
IAR A
IAR C
IAR D
IAR K
IAR J
IAR N
IAR M
IAR V
IAR U
ISS

Figure 7.22

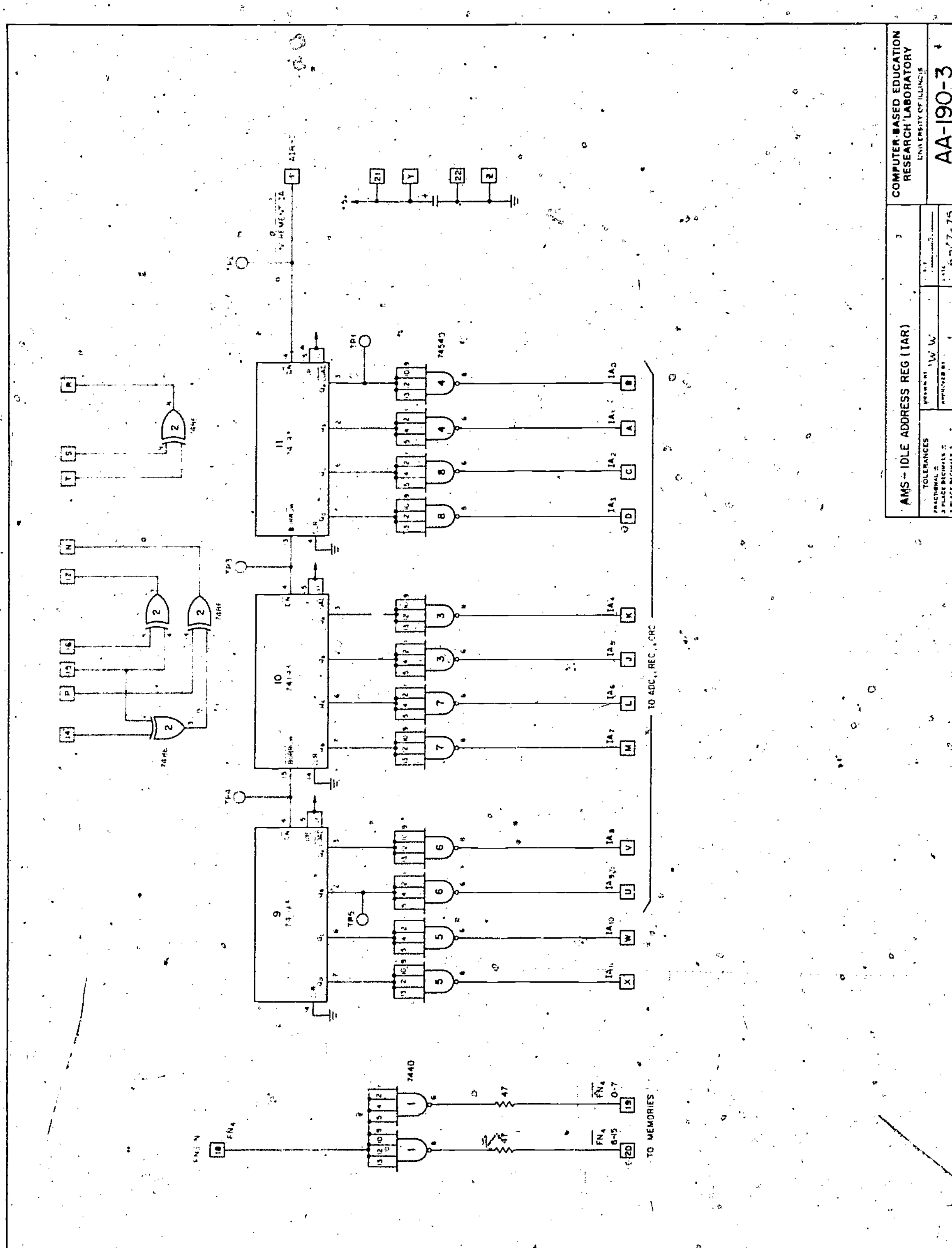

COMPUTER-BASED EDUCATION RESEARCH LABORATORY
UNIVERSITY OF ILLINOIS
AA-190-3
AMS - IDLE ADDRESS REG (IAR)
6-27-75
TO MEMORIES

Figure 7.23

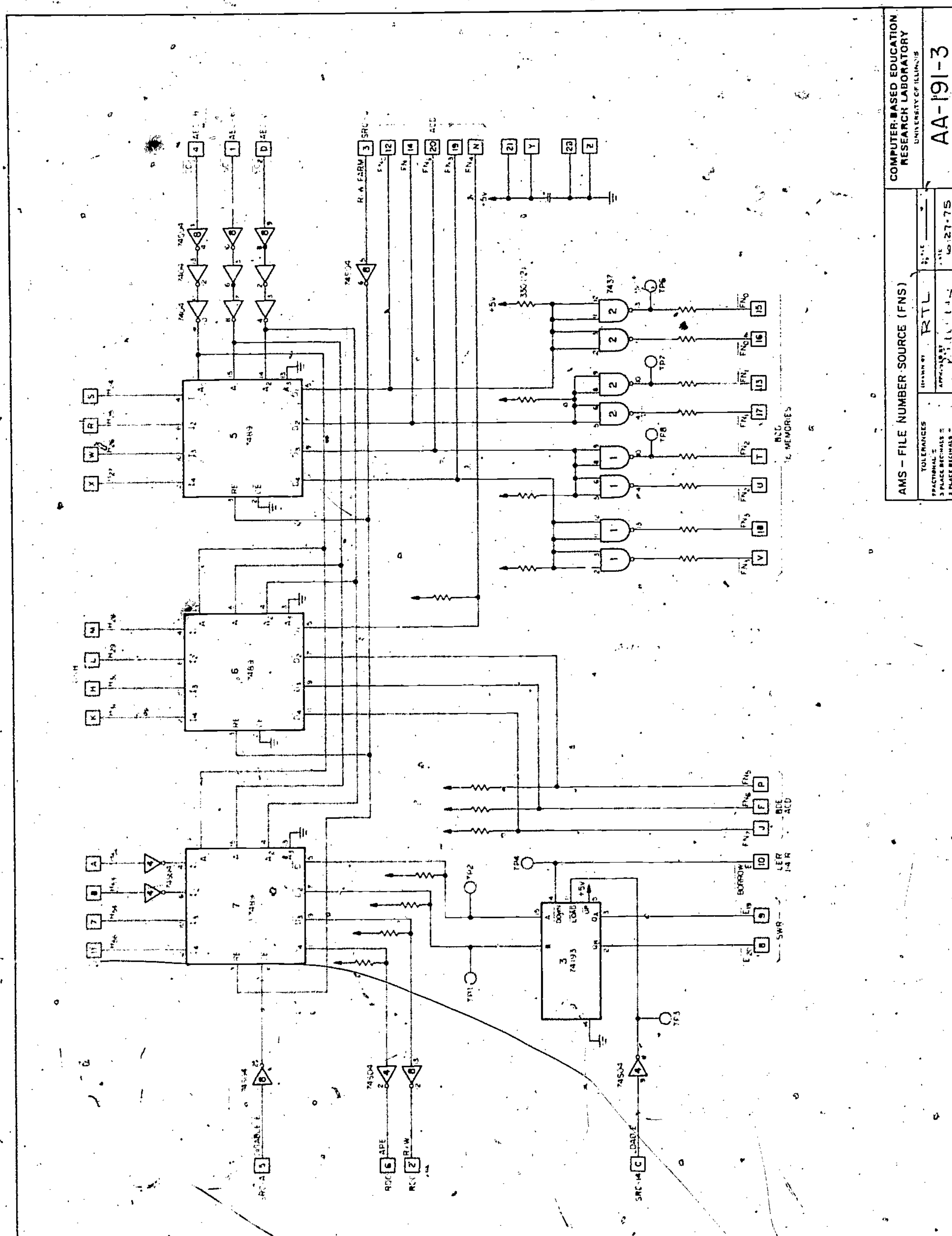

COMPUTER-BASED EDUCATION RESEARCH LABORATORY
UNIVERSITY OF ILLINOIS
AA-191-3
AMS – FILE NUMBER SOURCE (FNS)
RTL

Figure 7.24

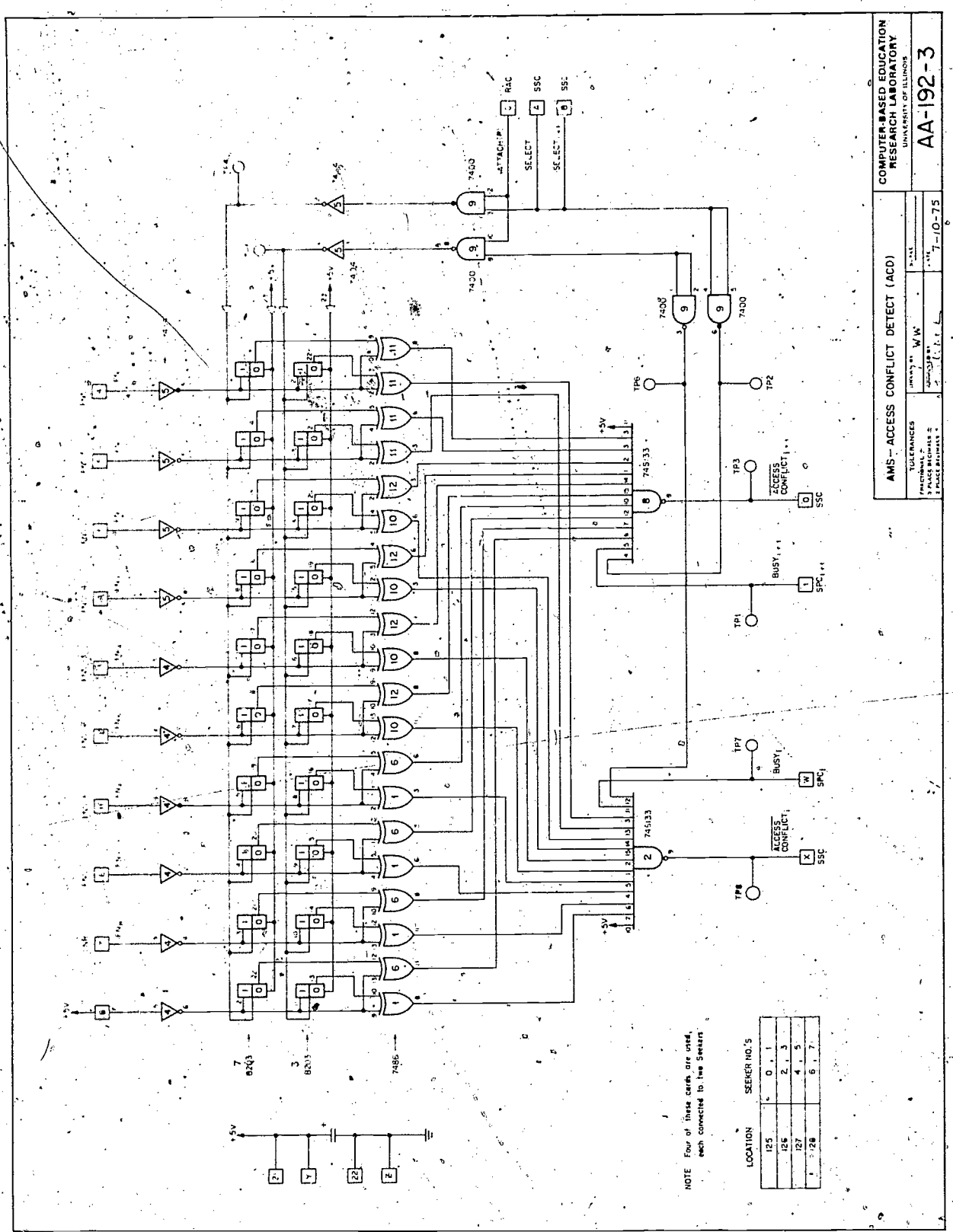

AMS — ACCESS CONFLICT DETECT (ACD)
COMPUTER-BASED EDUCATION RESEARCH LABORATORY
UNIVERSITY OF ILLINOIS
AA-192-3
7-10-75
RAC
SSC
SSC+1
SELECT
SELECT+1
7400
74S133
TP1
TP2
TP3
TP6
TP7
TP8
BUSY, r+1
SPC, r+1
ACCESS CONFLICT, r+1
BUSY, r
SPC, r
ACCESS CONFLICT, r
8213
7486
NOTE Four of these cards are used, each connected to two Seekers
LOCATION
SEEKER NO.'S
125
126
127
128

Figure 7.25

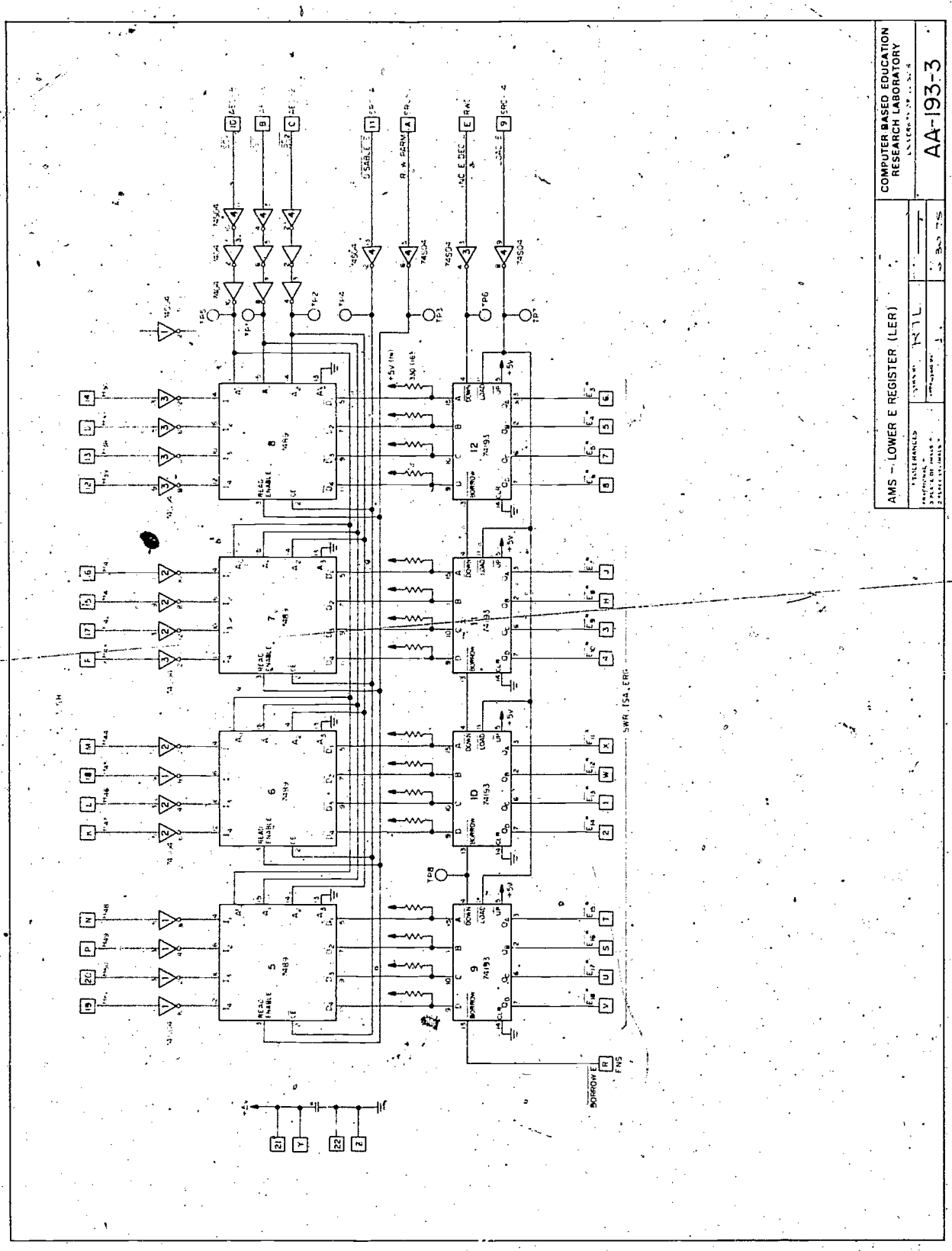

COMPUTER BASED EDUCATION RESEARCH LABORATORY
AA-193-3
AMS - LOWER E REGISTER (LER)
74193
74504
+5V
TP1
TP2
TP3
TP4
TP5
TP6
TP7
TP8
BORROW E
FMS

Figure 7.26

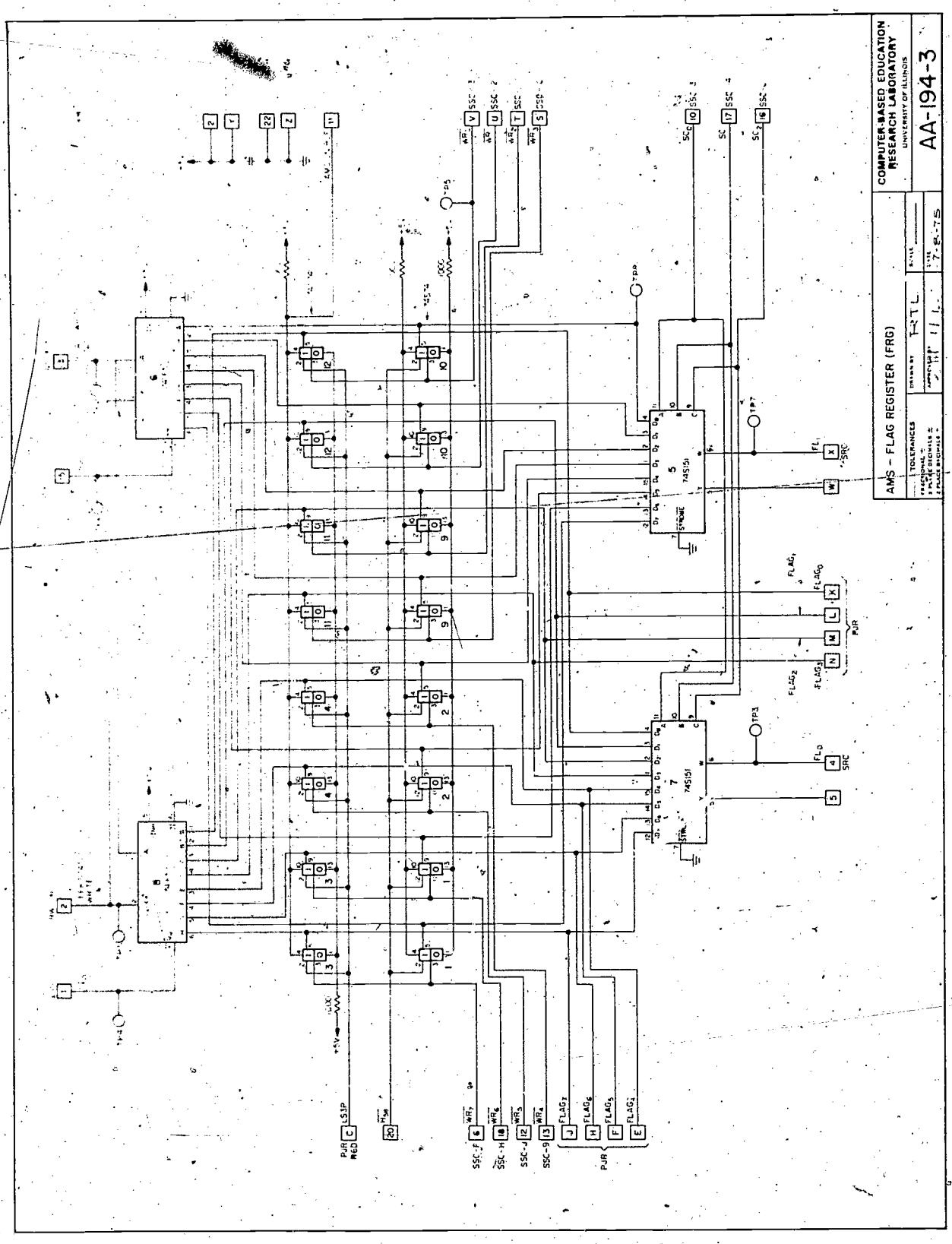

COMPUTER-BASED EDUCATION RESEARCH LABORATORY
UNIVERSITY OF ILLINOIS
AA-194-3
AMS - FLAG REGISTER (FRG)
DRAWN BY T-RTL
DATE 7-8-75
TOLERANCES
74S151
SSC-F
SSC-H
SSC-J
SSC-9
FLAG7
FLAG6
FLAG5
FLAG4
PJR
LS1P

Figure 7.27

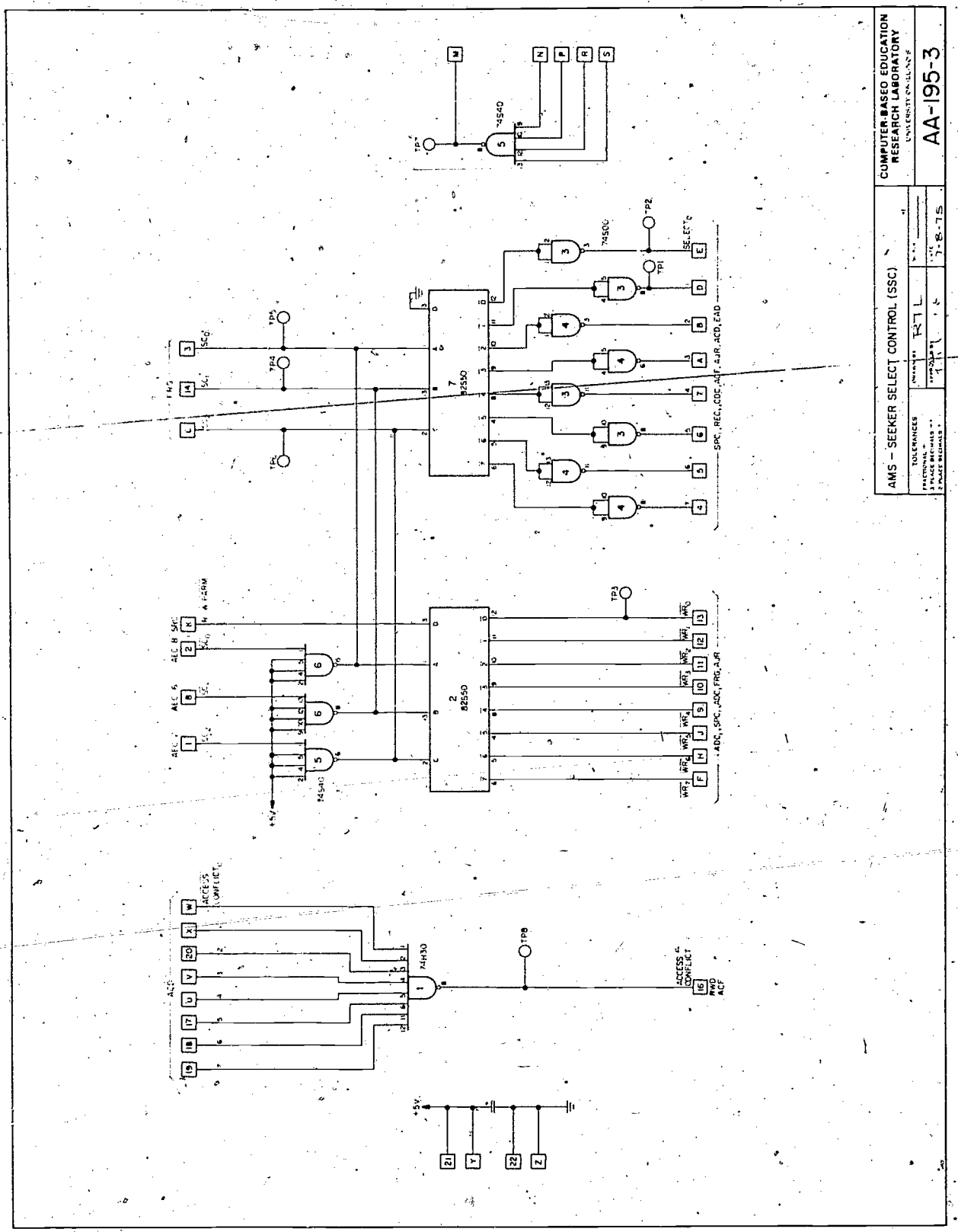

COMPUTER-BASED EDUCATION RESEARCH LABORATORY
AA-195-3
AMS – SEEKER SELECT CONTROL (SSC)
R7L
7-8-75
TOLERANCES
74540
74500
74H30
82S50
TP1
TP2
TP3
TP4
TP5
TP6
TP8
+5V
ACCESS CONFLICT
ACCESS CONFLICT RWD ACF
SPC, REC, CDC, ACF, AJR, ACD, EAD
ADC, SPC, ADC, FRG, AJR

Figure 7.28

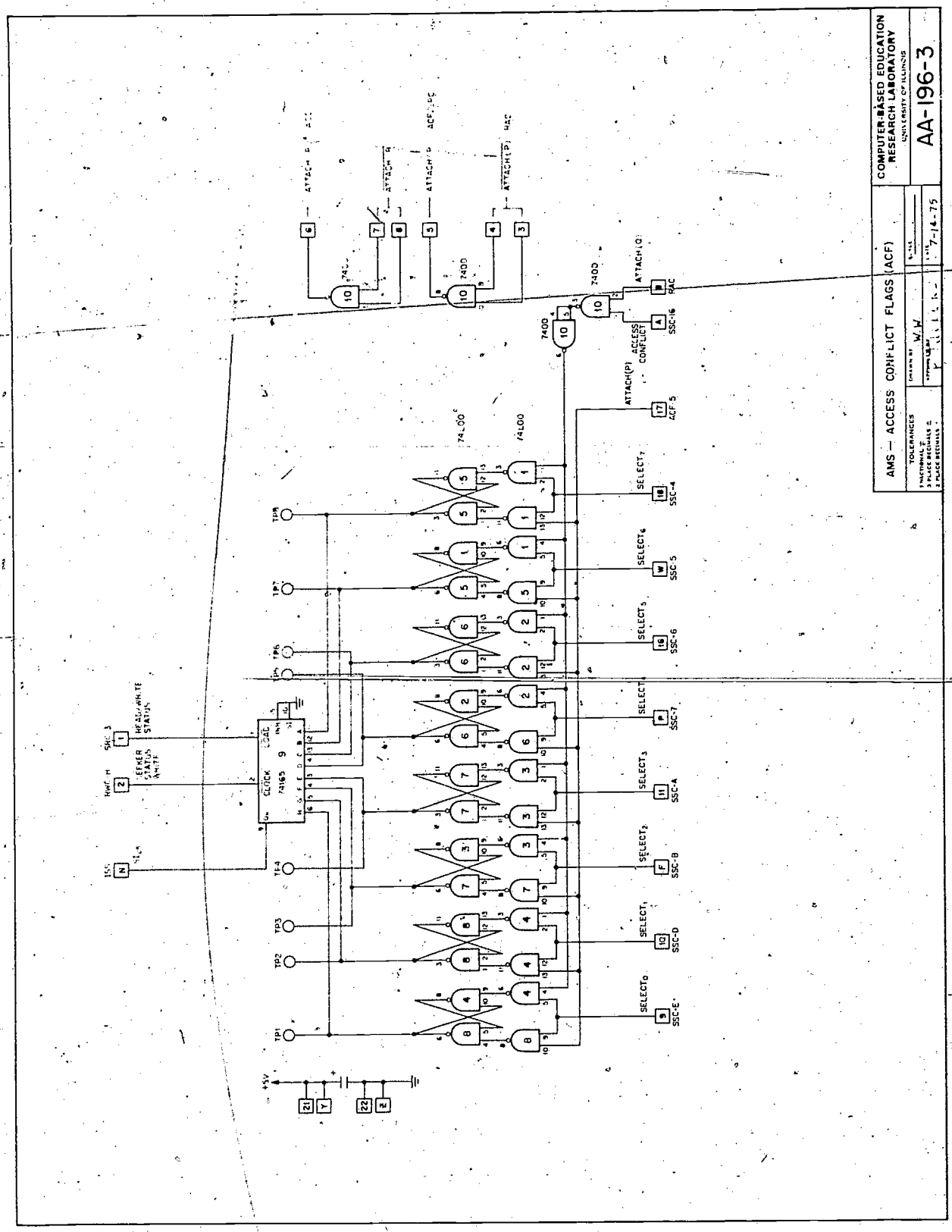

AMS — ACCESS CONFLICT FLAGS (ACF)
COMPUTER-BASED EDUCATION RESEARCH LABORATORY
UNIVERSITY OF ILLINOIS
AA-196-3
7-14-75
W.H
ATTACH(Q)
ATTACH(P)
ACCESS CONFLICT
RAC
ACF 5
SELECT7
SELECT6
SELECT5
SELECT4
SELECT3
SELECT2
SELECT1
SELECT0
SSC-4
SSC-5
SSC-6
SSC-7
SSC-A
SSC-B
SSC-D
SSC-E
7400
74LS00
74165
CLOCK
LOAD
TP1
TP2
TP3
TP4
TP5
TP6
TP7
+5V

Figure 7.29

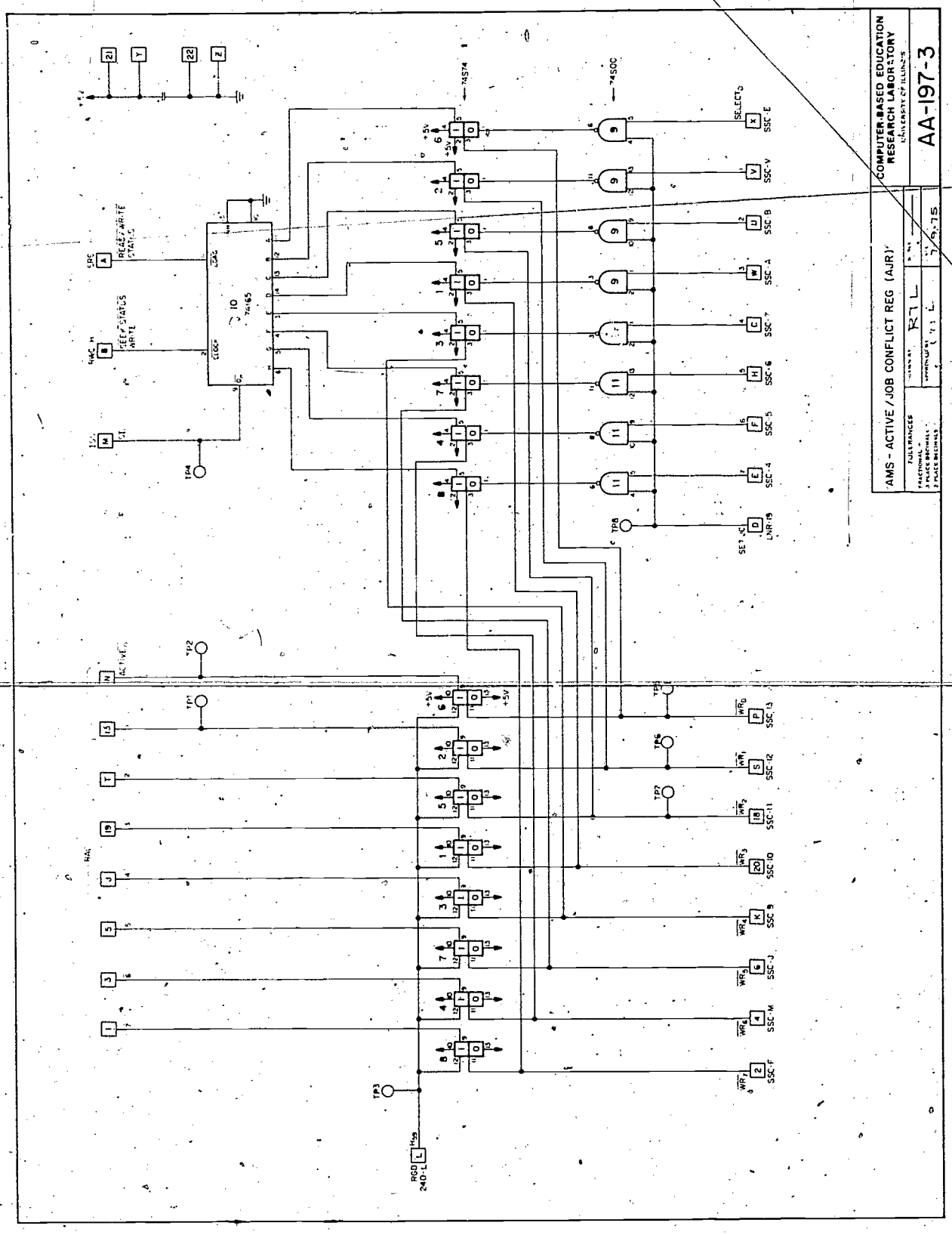

AMS - ACTIVE / JOB CONFLICT REG (AJR)
COMPUTER-BASED EDUCATION RESEARCH LABORATORY
AA-197-3
74S74
74S00
74S65
SSC-E
SSC-V
SSC-B
SSC-A
SSC-7
SSC-6
SSC-5
SSC-4
SSC-13
SSC-12
SSC-11
SSC-10
SSC-9
SSC-J
SSC-M
SSC-F
TP1
TP2
TP3
TP4
TP5
TP6
TP8
+5V

Figure 7.30

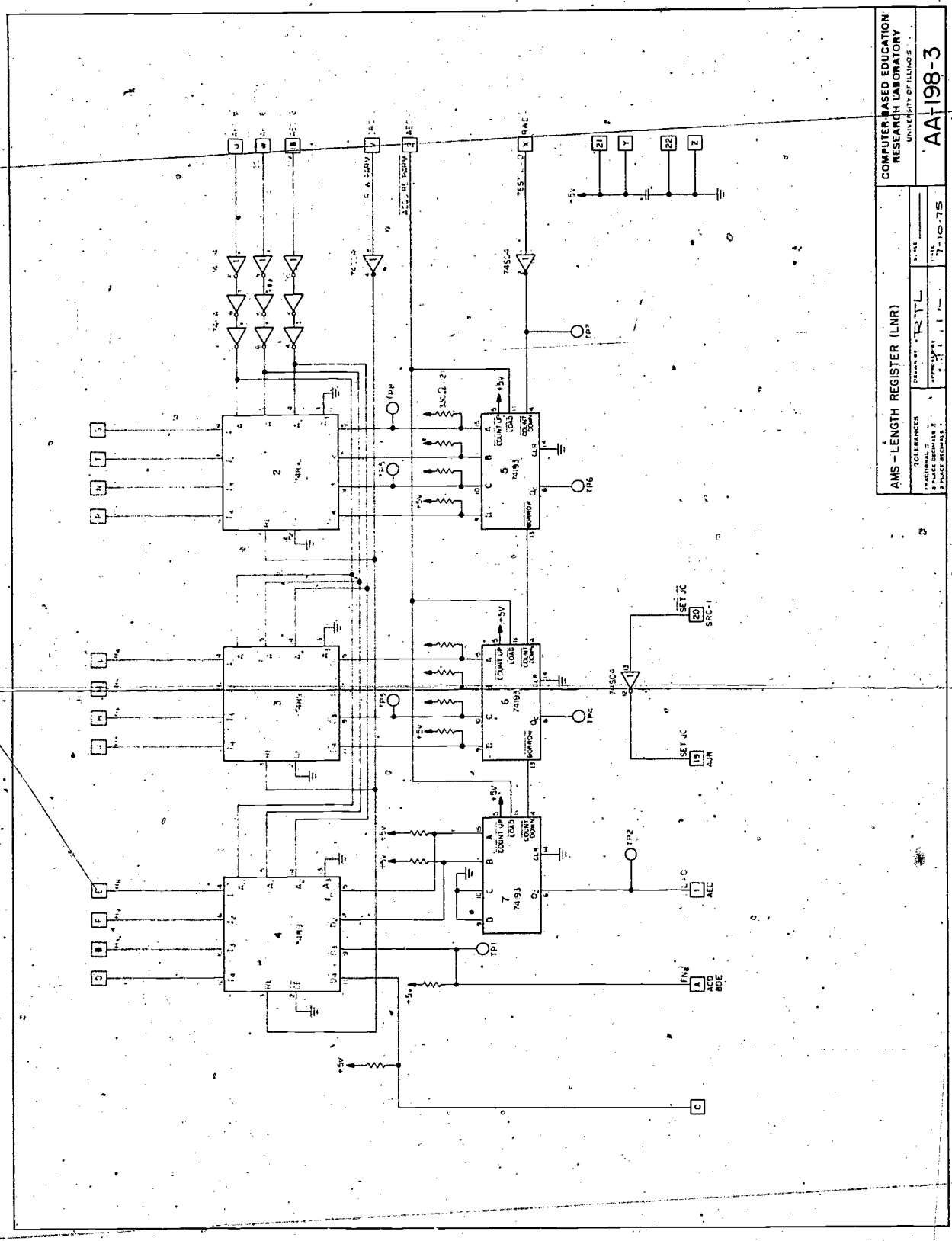

COMPUTER-BASED EDUCATION RESEARCH LABORATORY
UNIVERSITY OF ILLINOIS
AA-198-3
AMS – LENGTH REGISTER (LNR)
RTL
7-10-75
TOLERANCES
FRACTIONAL ±
3 PLACE DECIMALS ±
2 PLACE DECIMALS ±
74193
74S04
SET JC
SRC-1
AJR
AEC
ADD
BDE
TP1
TP2
TP3
TP4
TP5
TP6
+5V

Figure 7.31

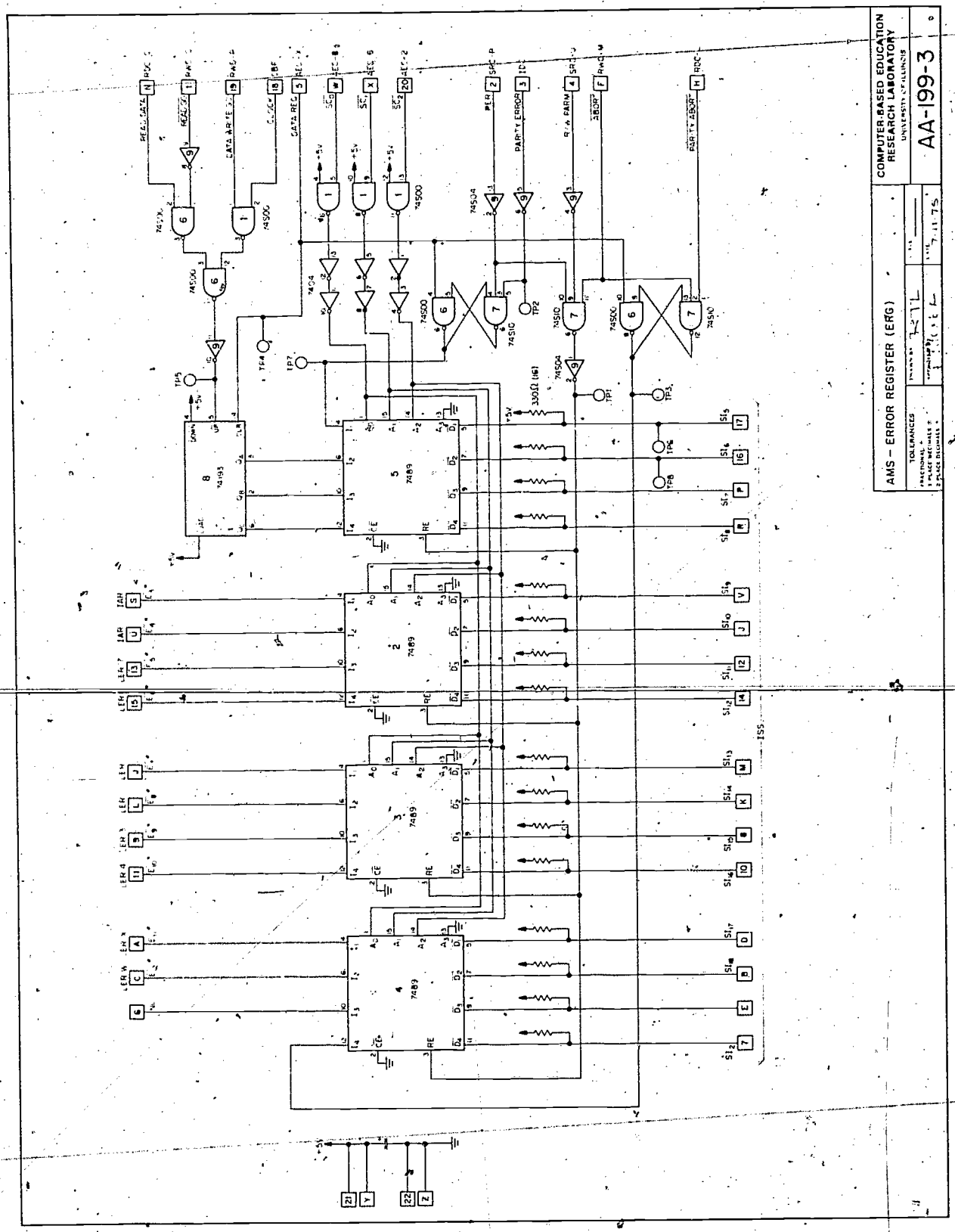

COMPUTER-BASED EDUCATION RESEARCH LABORATORY
UNIVERSITY OF ILLINOIS
AA-199-3
AMS – ERROR REGISTER (ERG)
TOLERANCES
FRACTIONAL ±
1 PLACE DECIMALS ±
2 PLACE DECIMALS ±
7-11-75
74193
7489
74500
74504
74S10
330Ω (4)
+5V
TP1
TP2
TP3
TP4
TP5
TP6
TP7
PARITY ERROR
PARITY ABORT

Figure 7.32

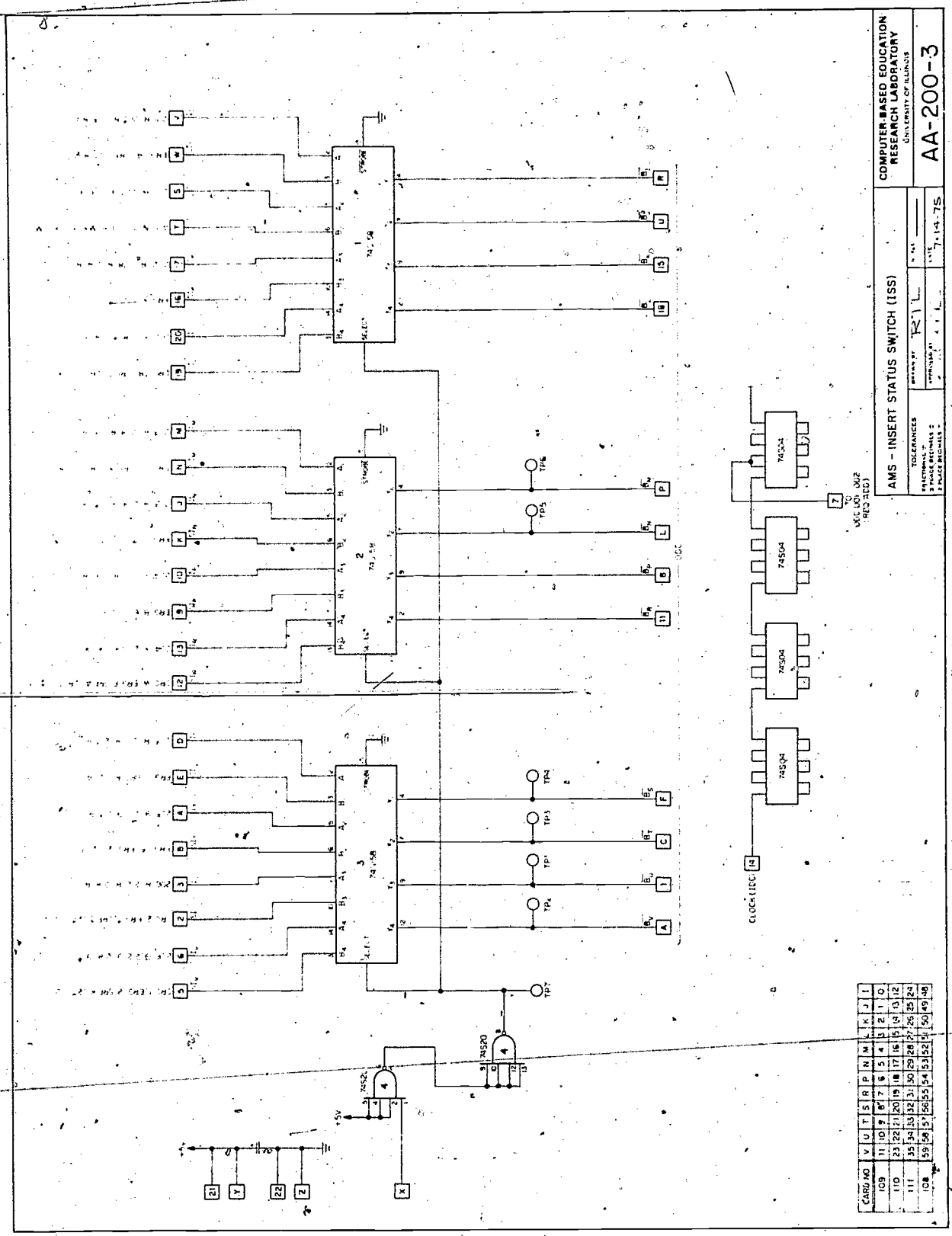

AMS - INSERT STATUS SWITCH (ISS)
COMPUTER-BASED EDUCATION RESEARCH LABORATORY
UNIVERSITY OF ILLINOIS
AA-200-3
7-14-75
74504
74504
74504
74504
74520
74S2
CLOCK (ICC)
CARD NO
109
110
111
108

Figure 7.33

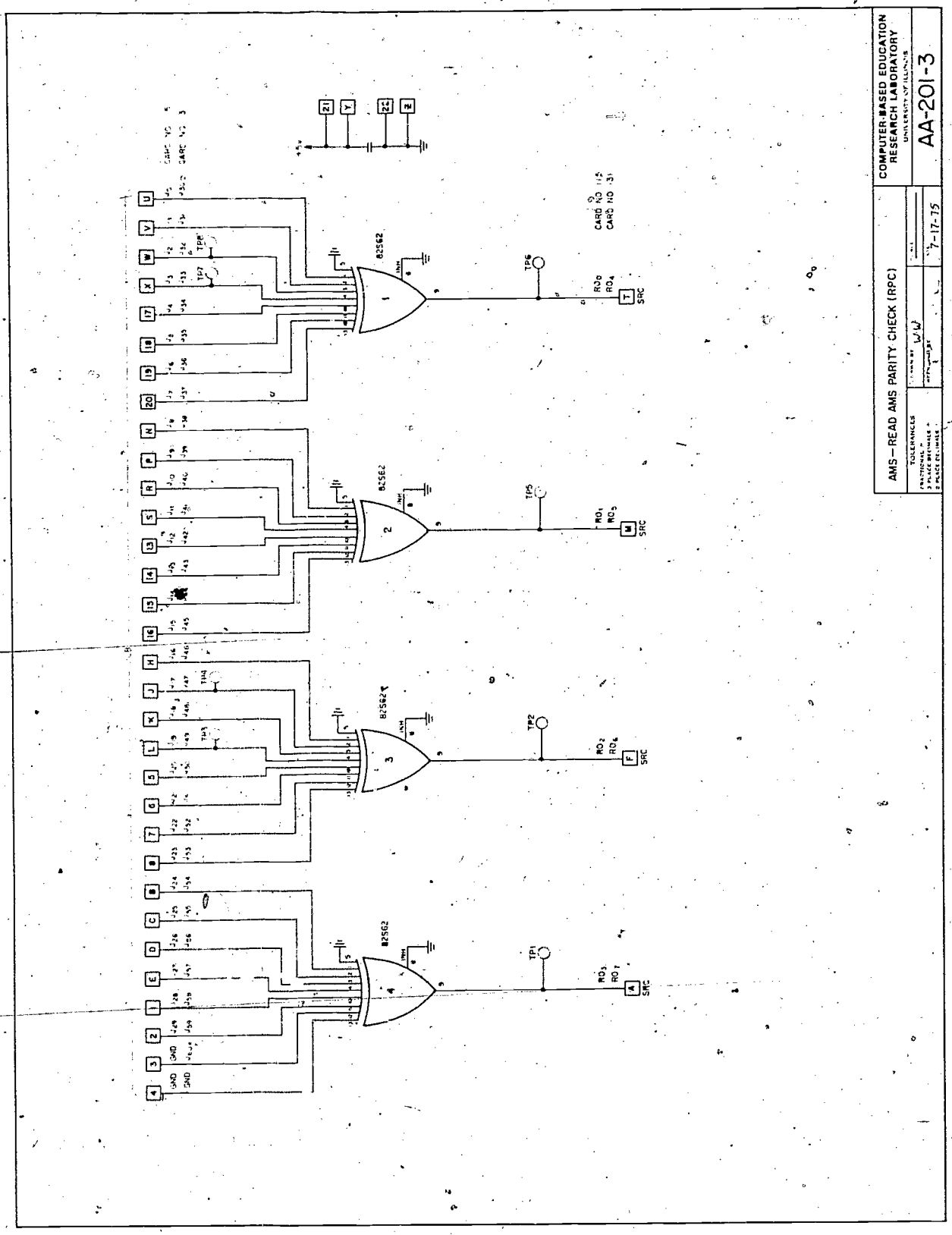

AMS—READ AMS PARITY CHECK (RPC)
COMPUTER-BASED EDUCATION RESEARCH LABORATORY
AA-201-3
7-17-75
82562
TP6
TP5
TP2
TP1
SRC

Figure 7.34

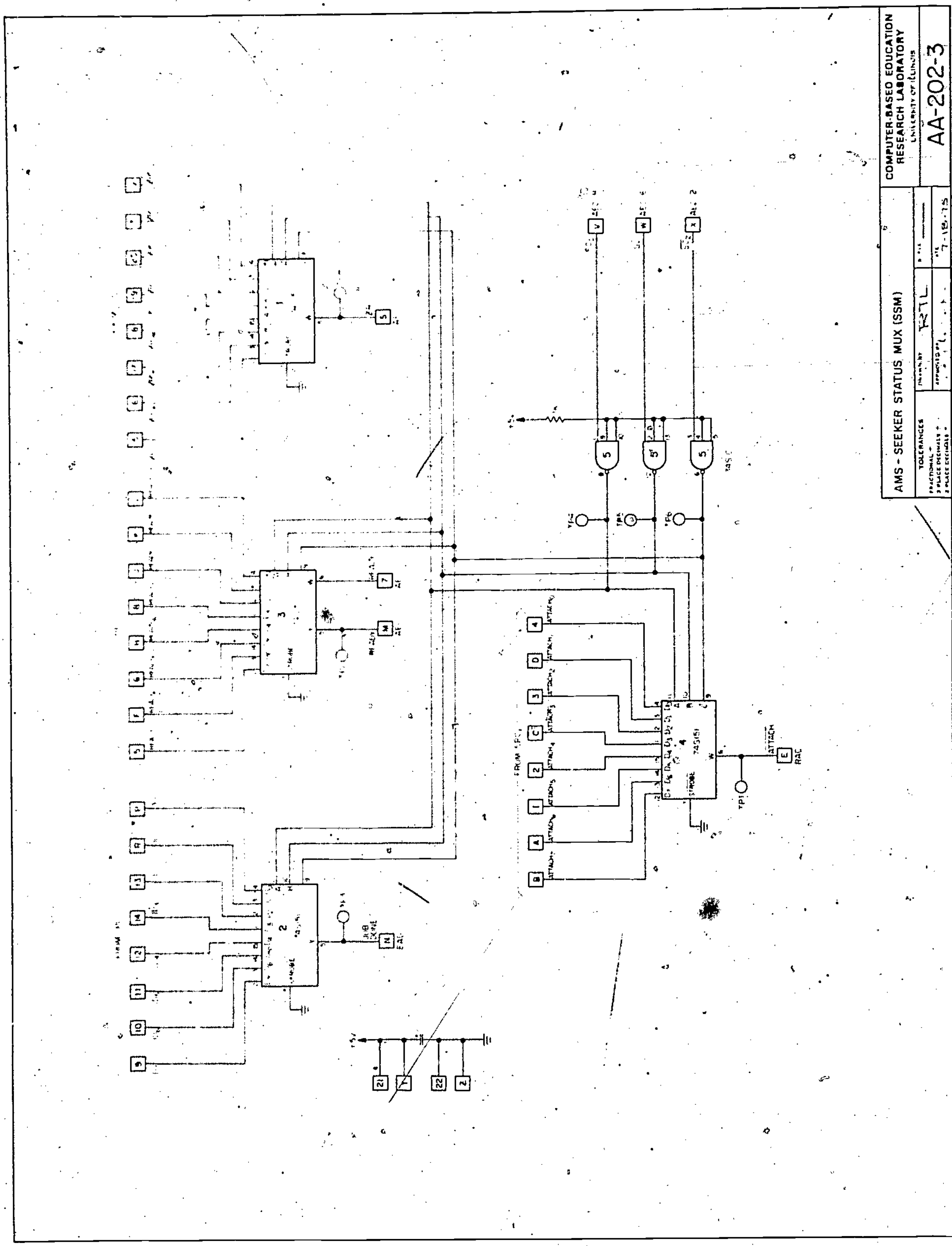

COMPUTER-BASED EDUCATION
RESEARCH LABORATORY
AA-202-3
AMS - SEEKER STATUS MUX (SSM)
RTL
74S151

Figure 7.35

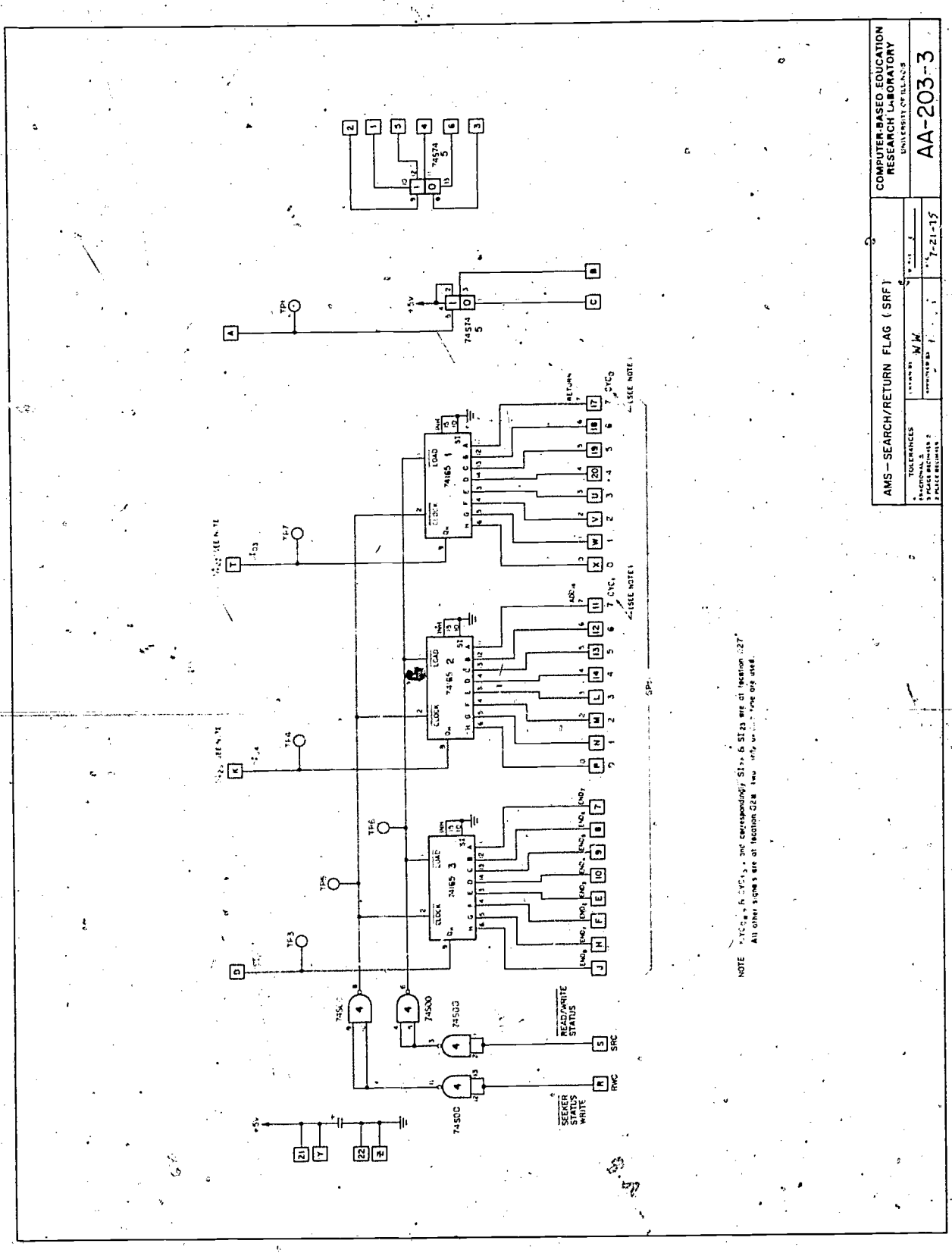

AMS — SEARCH/RETURN FLAG (SRF)
COMPUTER-BASED EDUCATION RESEARCH LABORATORY
UNIVERSITY OF ILLINOIS
AA-203-3
7-21-15
TOLERANCES
74574
74165
74500
74503
74510
READ/WRITE STATUS
SEEKER STATUS WRITE
SRC
RWC
RETURN
LOAD
CLOCK
TP1
TP3
TP4
TP5
TP6
TP7

Figure 7.36

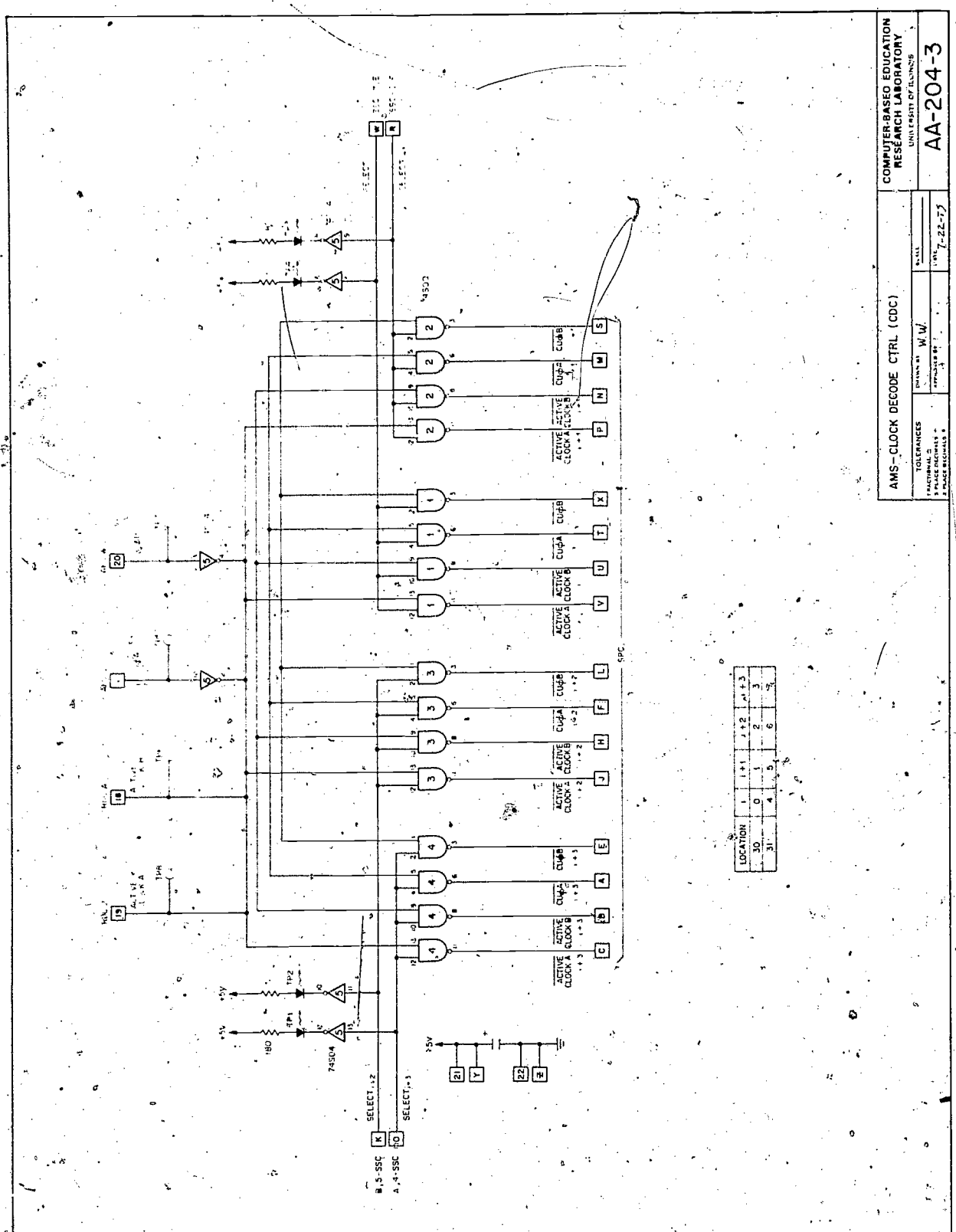

AMS-CLOCK DECODE CTRL (CDC)
COMPUTER-BASED EDUCATION RESEARCH LABORATORY
AA-204-3
7-22-77
W.W.
LOCATION
SELECT.+2
SELECT.+3
74S04
180
TP1
TP2
+5V

Figure 7.37

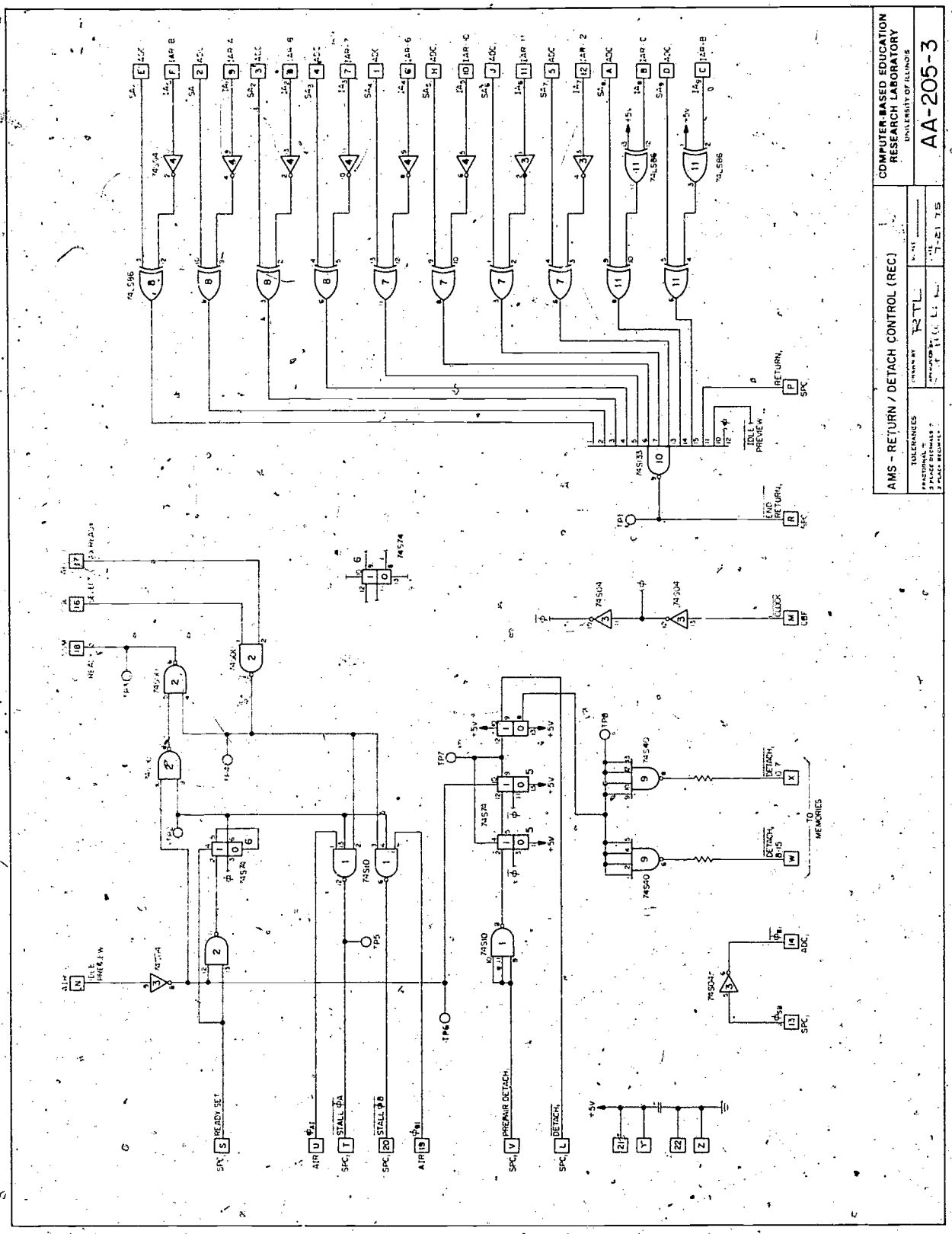

AMS - RETURN / DETACH CONTROL (REC)
COMPUTER-BASED EDUCATION RESEARCH LABORATORY
UNIVERSITY OF ILLINOIS
AA-205-3
TOLERANCES
DRAWN BY R2TL
7-21-75
RETURN,
END RETURN,
IDLE PREVIEW
READY SET
STALL ΦA
STALL ΦB
PREMAIN DETACH,
DETACH,
DETACH 0-7
DETACH 8-15
TO MEMORIES

Figure 7.38

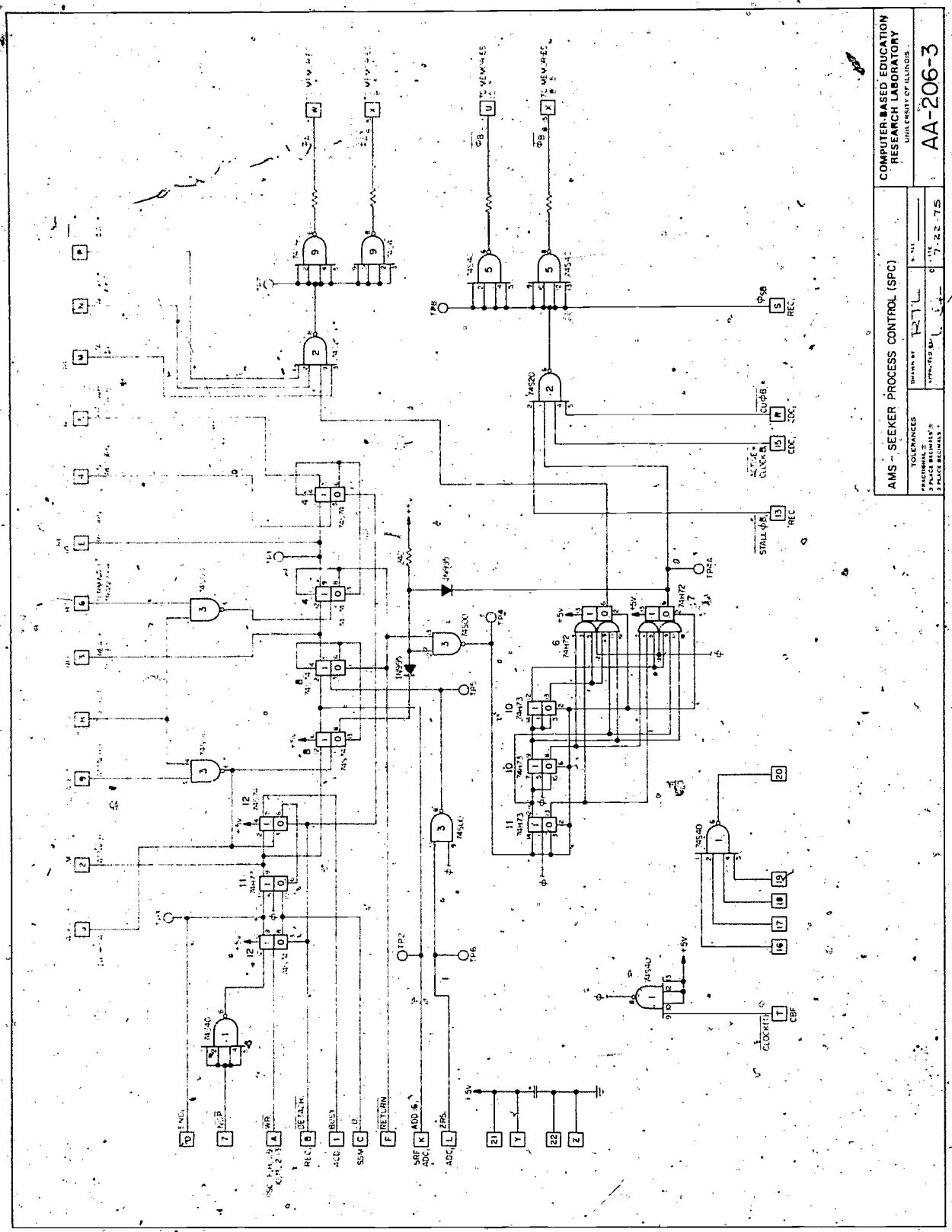

COMPUTER-BASED EDUCATION RESEARCH LABORATORY
UNIVERSITY OF ILLINOIS
AA-206-3
AMS - SEEKER PROCESS CONTROL (SPC)
TOLERANCES
7-22-75

Figure 7.39

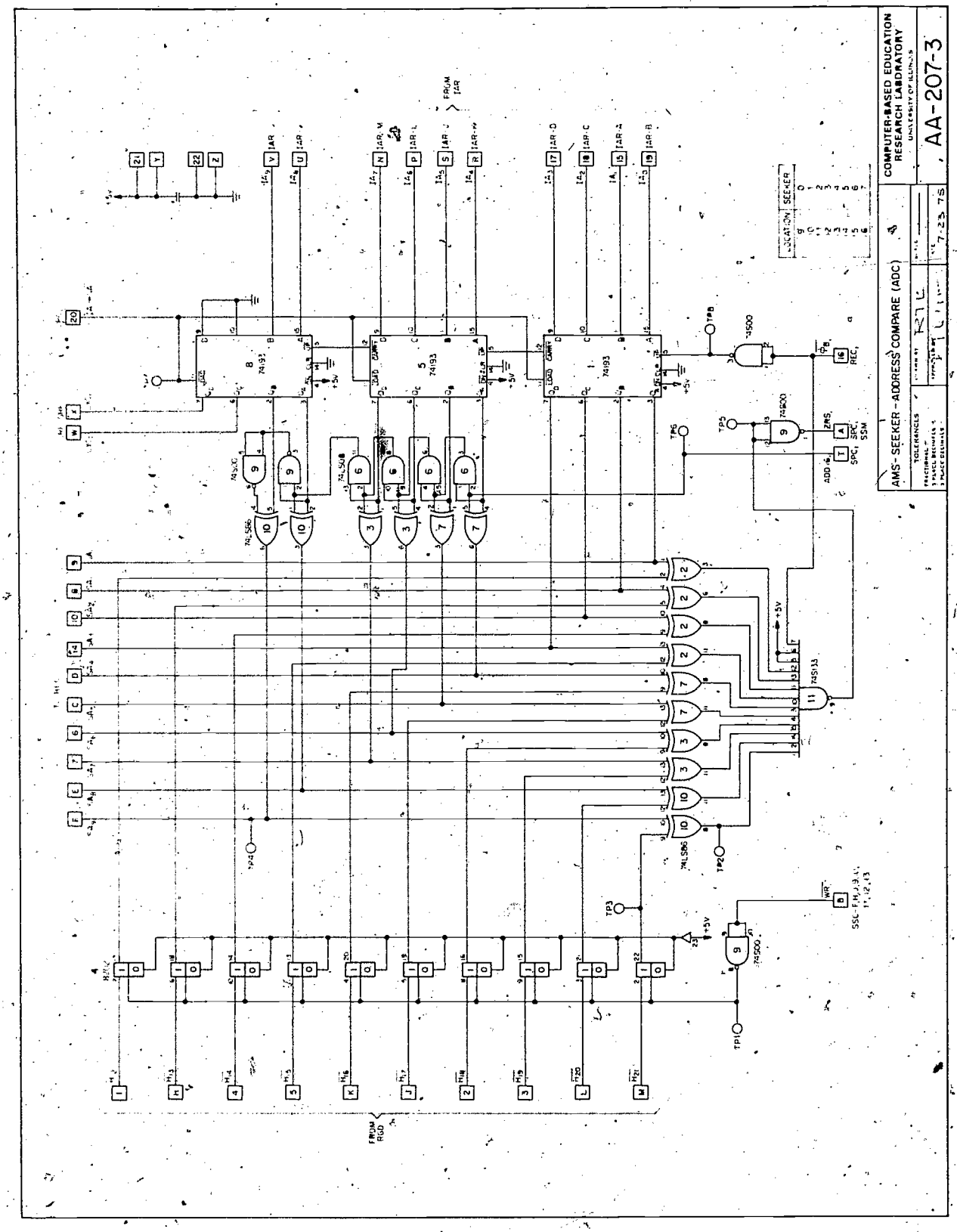

COMPUTER-BASED EDUCATION RESEARCH LABORATORY
AMS-SEEKER-ADDRESS COMPARE (ADC)
AA-207-3
7-23-75
LOCATION SEEKER
FROM IAR
FROM RGD
74193
74LS86
74S00
74S133

Figure 7.40

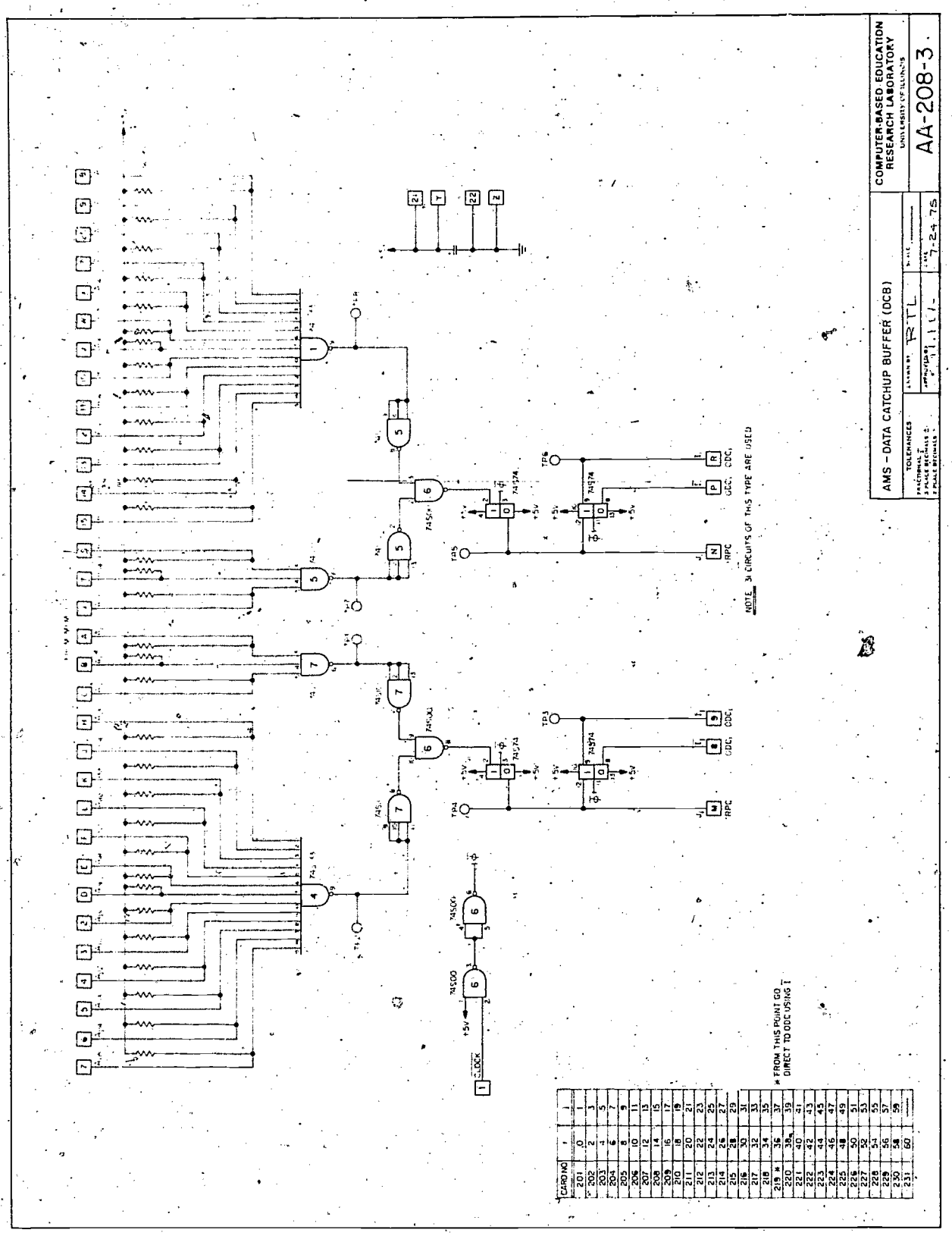

AMS - DATA CATCHUP BUFFER (DCB)
COMPUTER-BASED EDUCATION RESEARCH LABORATORY
AA-208-3
RTL
7-24-75
NOTE: 34 CIRCUITS OF THIS TYPE ARE USED
* FROM THIS POINT GO DIRECT TO DDC USING T
CLOCK

Figure 7.41

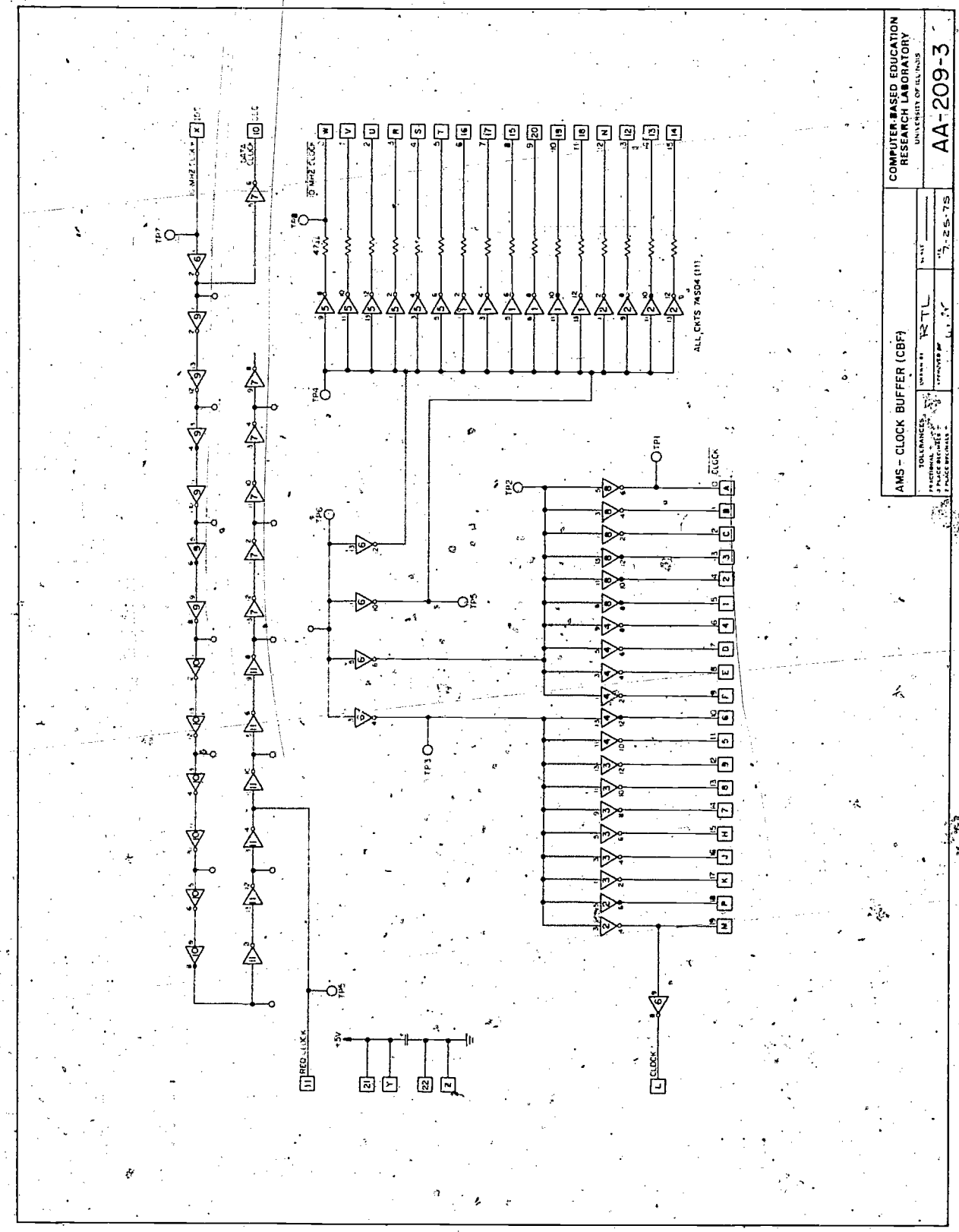

COMPUTER-BASED EDUCATION RESEARCH LABORATORY
UNIVERSITY OF ILLINOIS
AA-209-3
AMS - CLOCK BUFFER (CBF)
TOLERANCES
DRAWN BY RTL
7-25-75
ALL CKTS 74S04 (11)
TP1
TP2
TP3
TP4
TP5
TP6
TP7
TP8
CLOCK
+5V

Figure 7.42

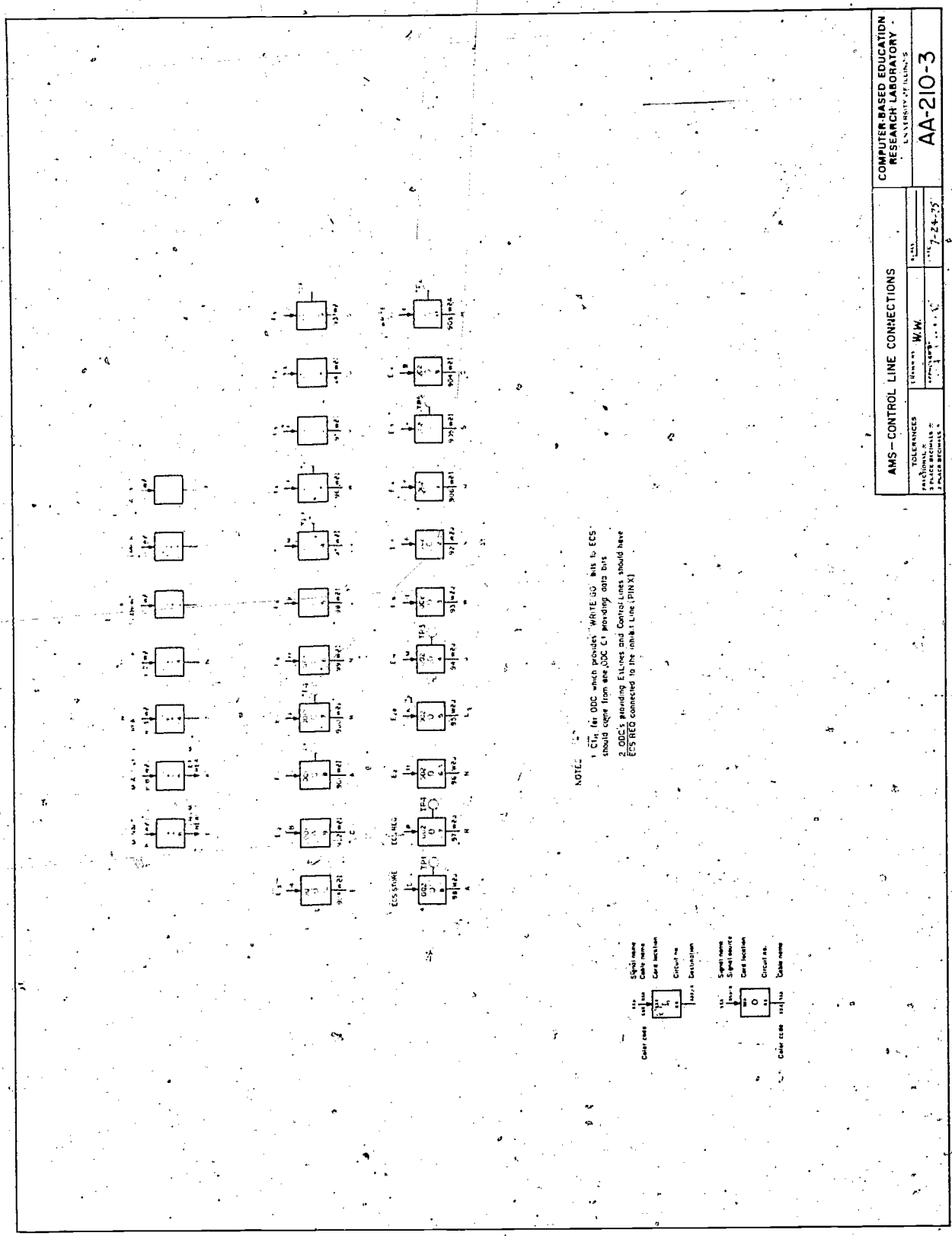

AMS—CONTROL LINE CONNECTIONS
COMPUTER-BASED EDUCATION RESEARCH LABORATORY
AA-210-3
W.W.
7-24-75
NOTES:
1. $\overline{CI_{11}}$ for ODC which provides WRITE GO bits to ECS should come from the ODC C1 providing data bits
2. ODC's providing Enables and Control Lines should have ECS REQ connected to the mark Line (PIN X)
ECS STORE
ECL HOLD
TP1
TP2
TP3
TP4
Signal name
Cable name
Card location
Circuit no.
Destination
Color code
Signal source

Figure 7.43

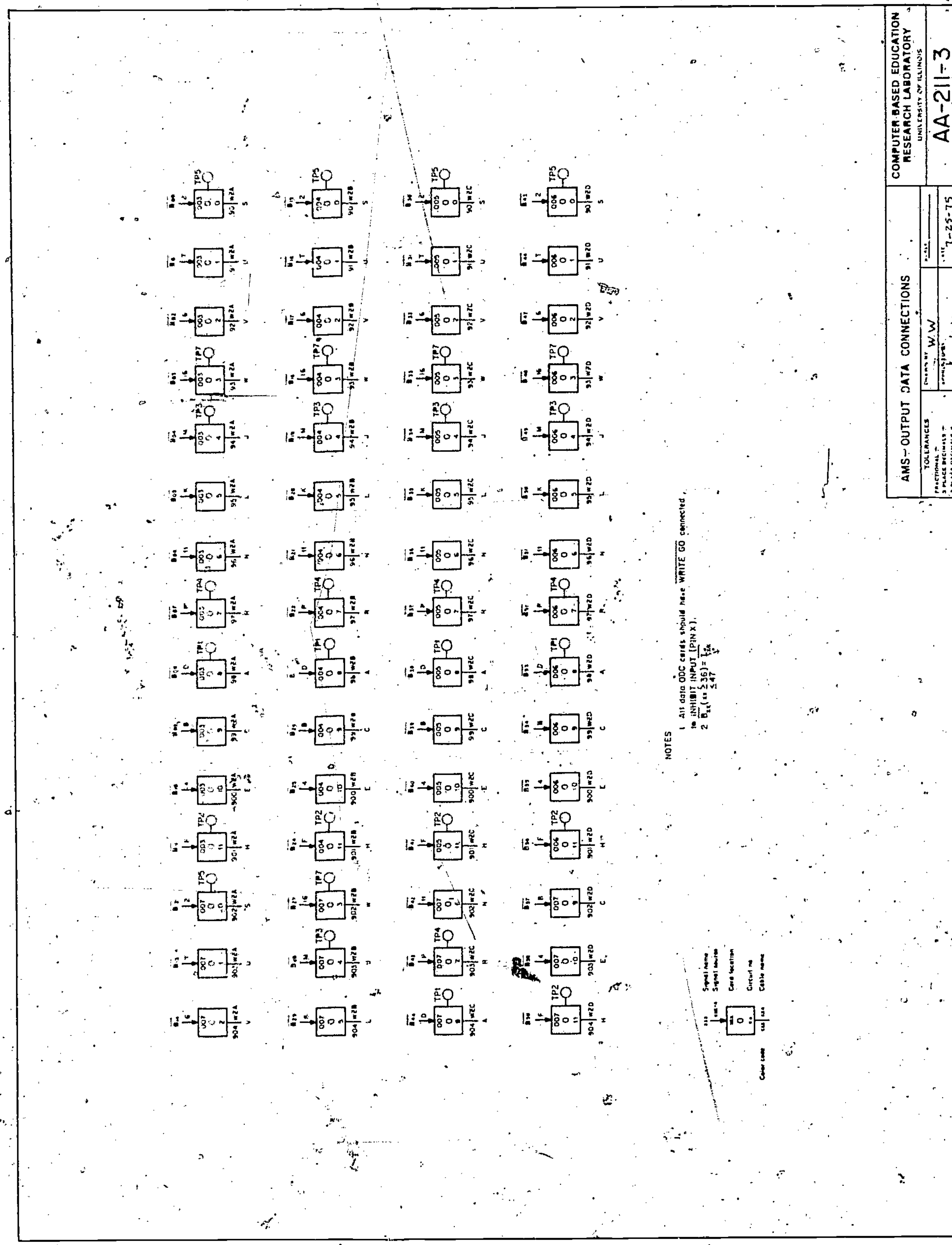

COMPUTER-BASED EDUCATION RESEARCH LABORATORY
UNIVERSITY OF ILLINOIS
AMS—OUTPUT DATA CONNECTIONS
AA-211-3
NOTES
1. All data ODC cards should have WRITE GO connected to INHIBIT INPUT (PIN X).

Figure 7.44

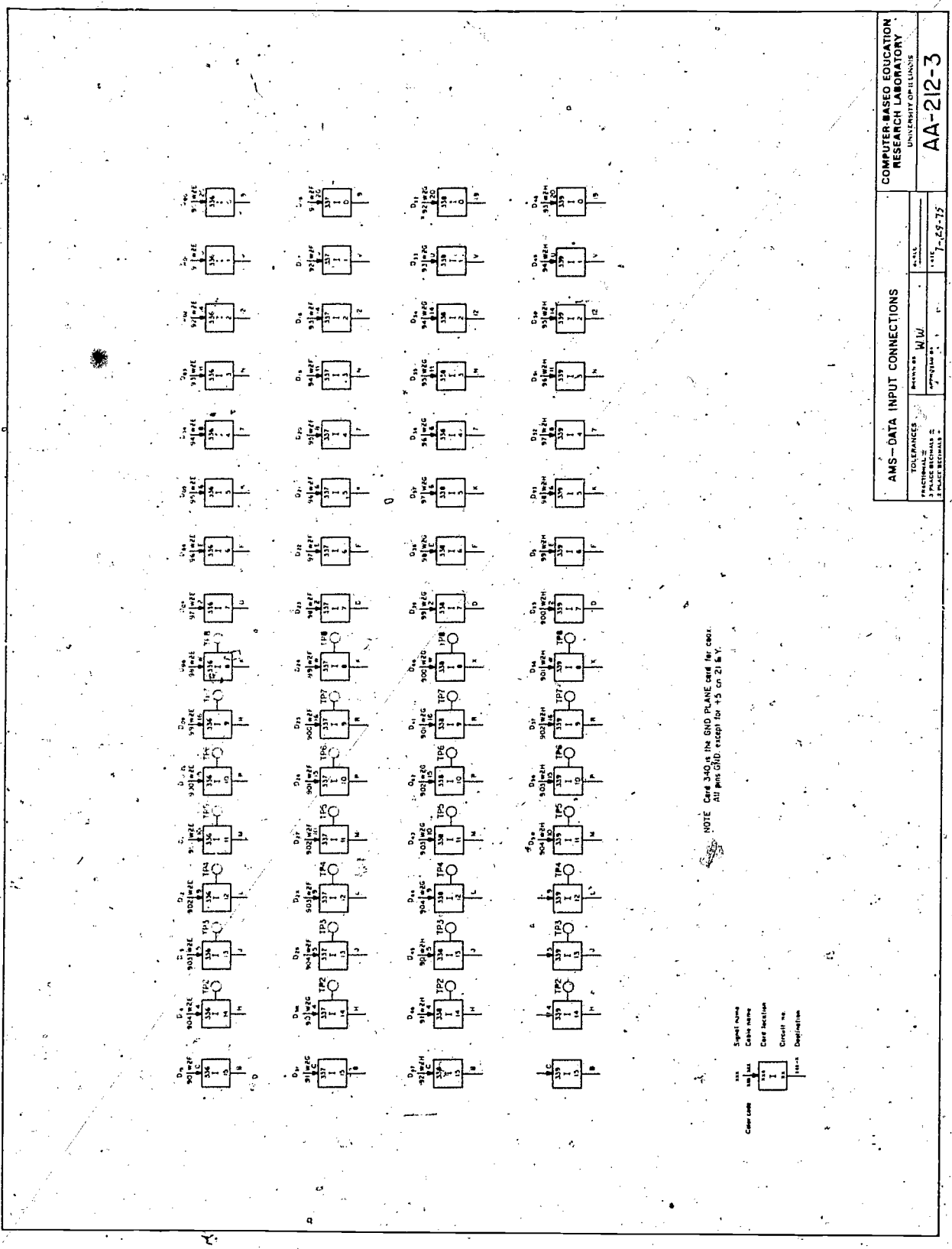

AMS-DATA INPUT CONNECTIONS
COMPUTER-BASED EDUCATION RESEARCH LABORATORY
UNIVERSITY OF ILLINOIS
AA-212-3
TOLERANCES
FRACTIONAL ±
2 PLACE DECIMALS ±
3 PLACE DECIMALS ±
Drawn by W.W.
Approved by
Date 7-29-75
NOTE Card 340 is the GND PLANE card for cable.
All pins GND except for +5 on 21 & Y.
Signal name
Cable name
Card location
Circuit no.
Destination
Color code

Figure 7.45

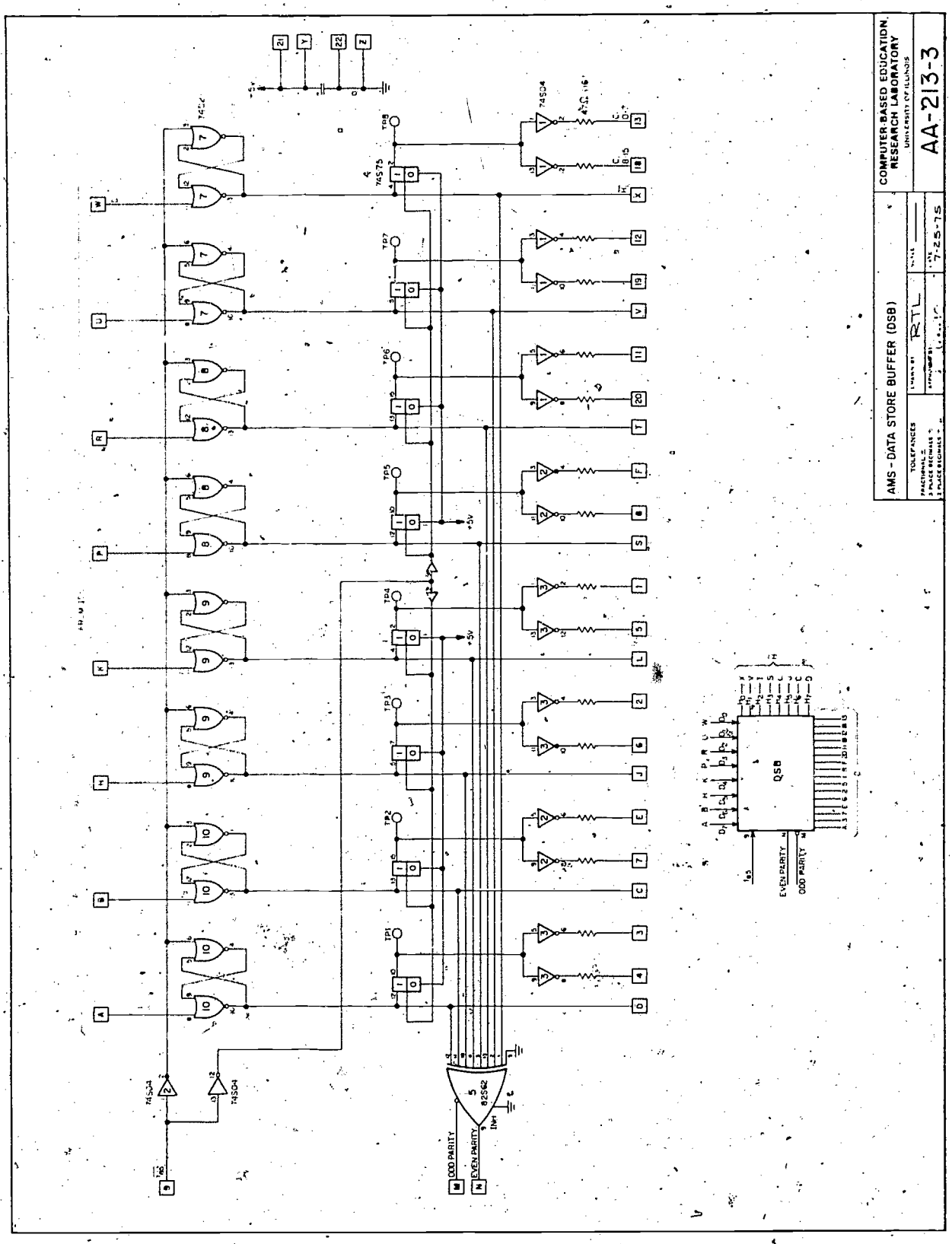

COMPUTER-BASED EDUCATION RESEARCH LABORATORY
UNIVERSITY OF ILLINOIS
AA-213-3
AMS - DATA STORE BUFFER (DSB)
RTL
7-25-75
TOLERANCES
QSB
EVEN PARITY
ODD PARITY
74S04
74S74
74S75
82S62

Figure 7.46

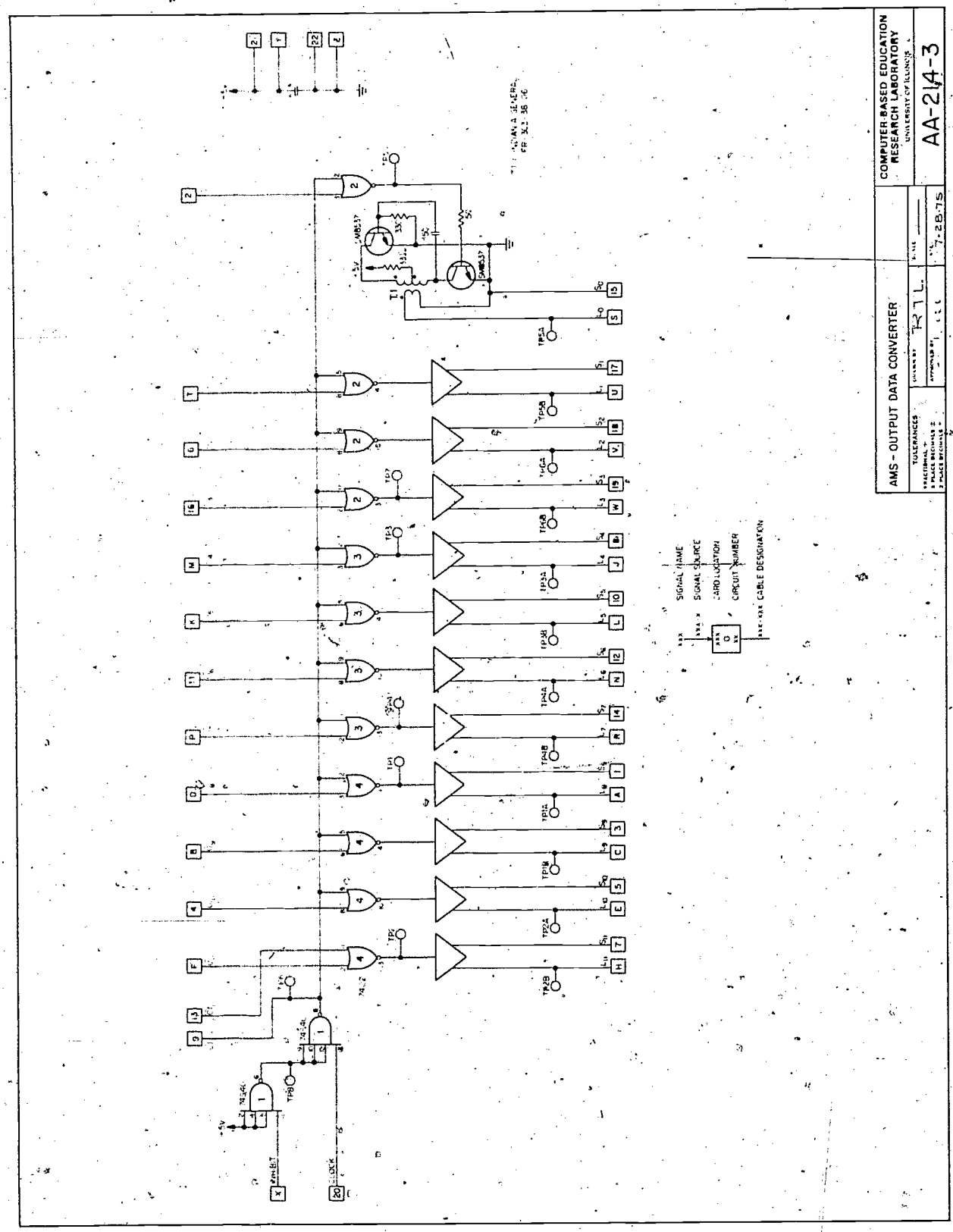

AMS - OUTPUT DATA CONVERTER
COMPUTER-BASED EDUCATION RESEARCH LABORATORY
UNIVERSITY OF ILLINOIS
AA-214-3
7-28-75
SIGNAL NAME
SIGNAL SOURCE
CARD LOCATION
CIRCUIT NUMBER
XXX-XXX CABLE DESIGNATION

Figure 7.47

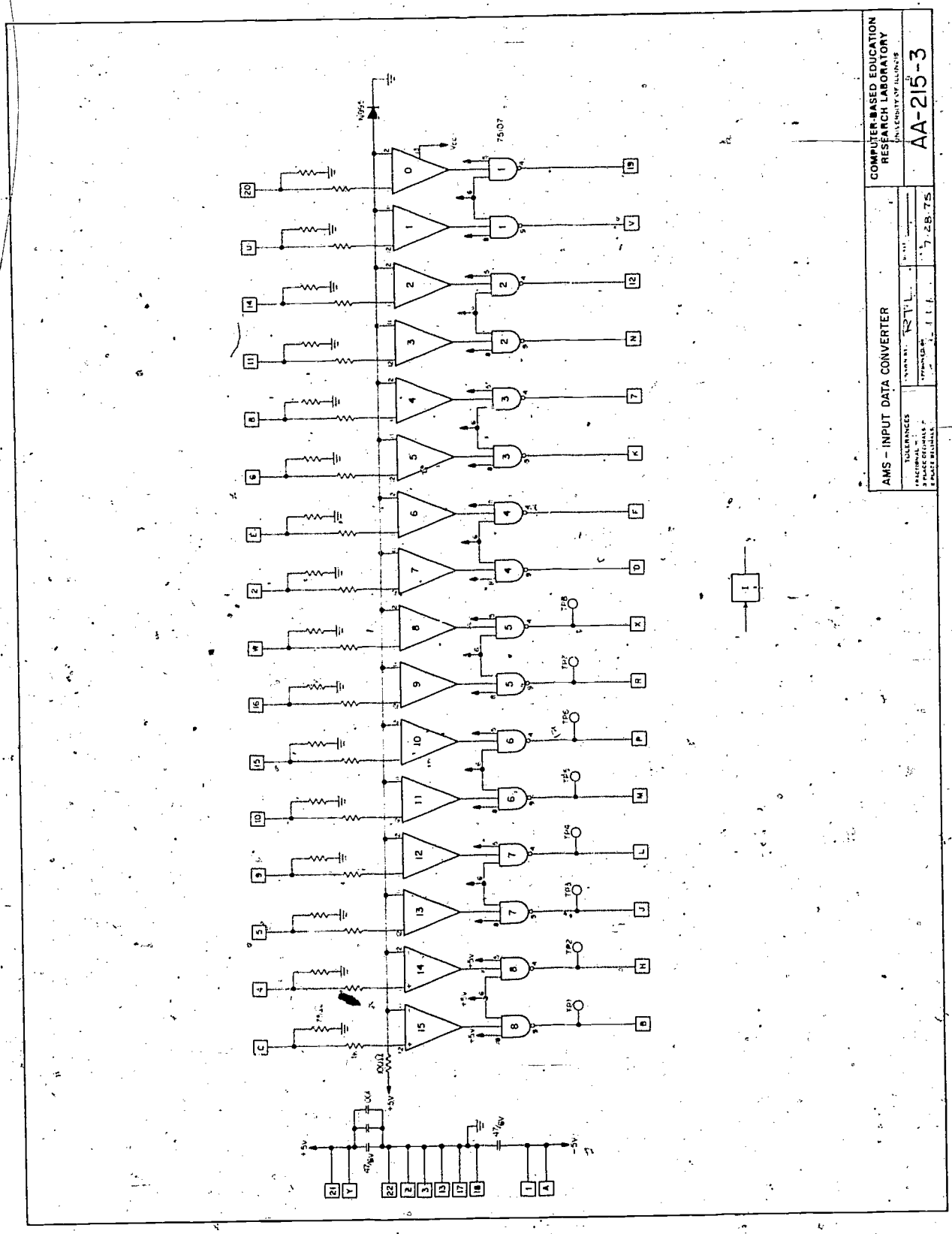

COMPUTER-BASED EDUCATION RESEARCH LABORATORY
UNIVERSITY OF ILLINOIS
AA-215-3
AMS - INPUT DATA CONVERTER
7.28.75
75107
+5V
-5V

Figure 7.48

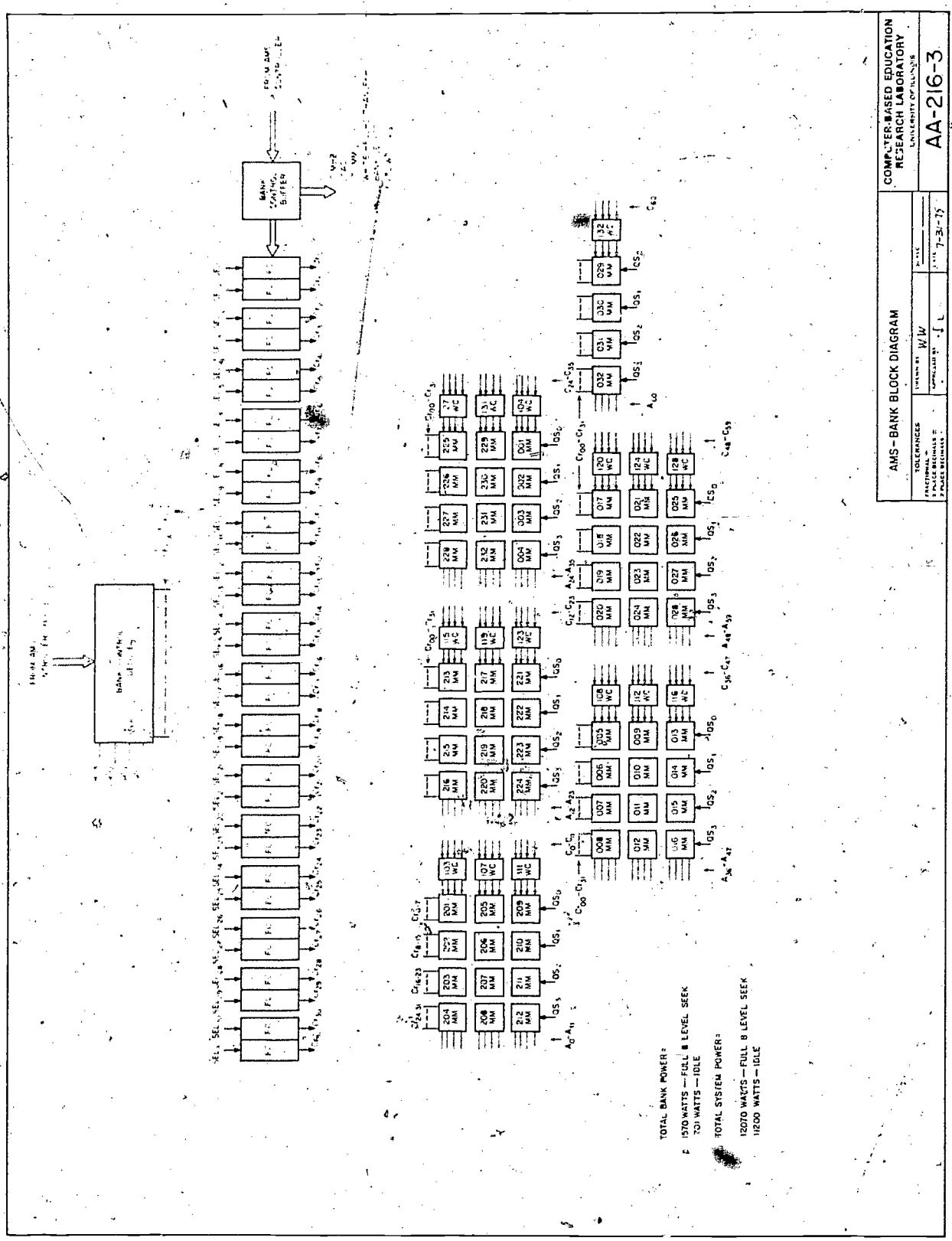

AMS-BANK BLOCK DIAGRAM
COMPUTER-BASED EDUCATION RESEARCH LABORATORY
UNIVERSITY OF ILLINOIS
AA-216-3
TOTAL BANK POWER=
1570 WATTS — FULL 8 LEVEL SEEK
701 WATTS — IDLE
TOTAL SYSTEM POWER=
12070 WATTS — FULL 8 LEVEL SEEK
11200 WATTS — IDLE

Figure 7.49

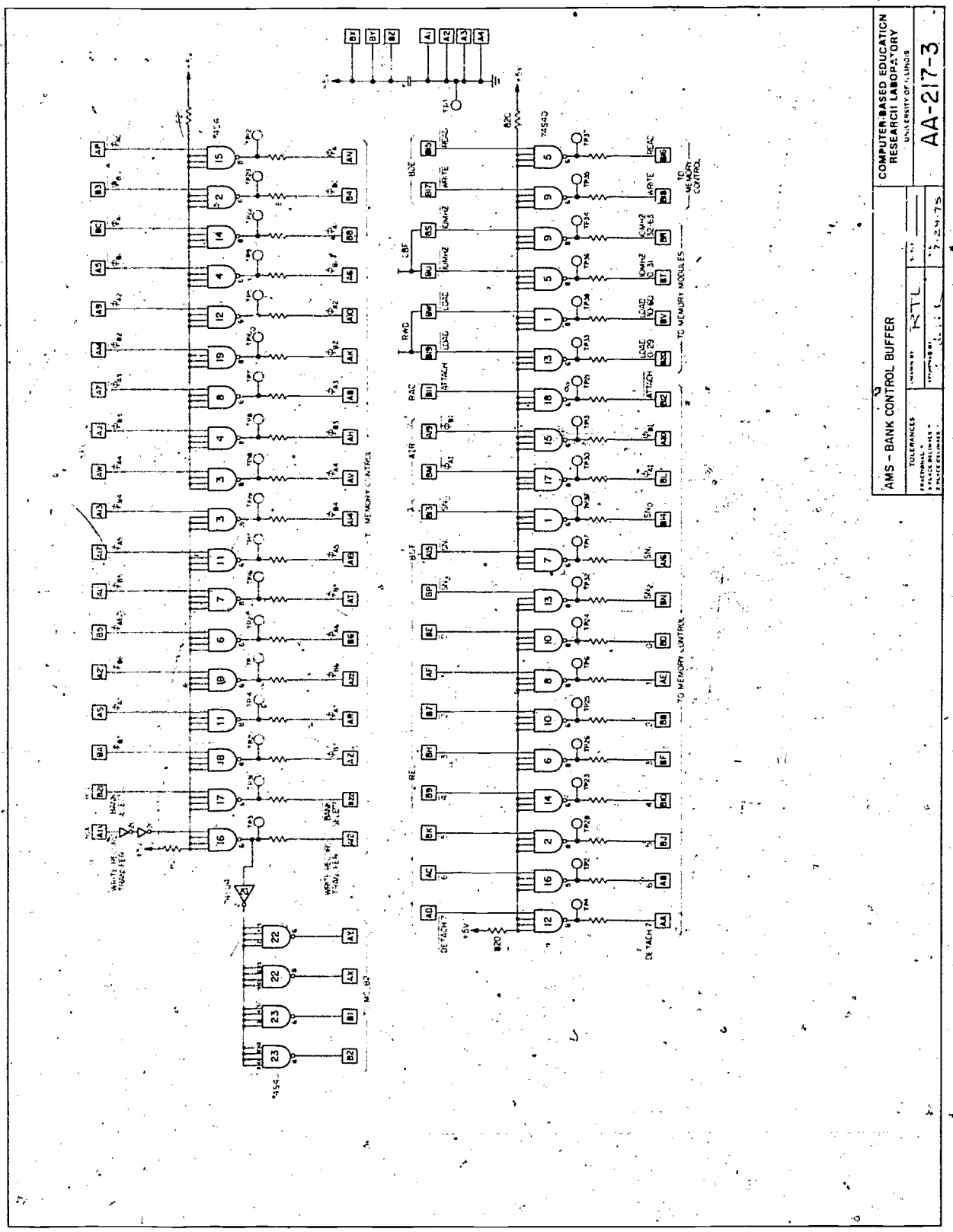

COMPUTER-BASED EDUCATION RESEARCH LABORATORY
UNIVERSITY OF ILLINOIS
AA-217-3
AMS - BANK CONTROL BUFFER
TOLERANCES
RTL
7.24.75
TO MEMORY CONTROL
TO MEMORY MODULES
TO MEMORY CONTROL
74S40
74S64
DETACH?
+5V

Figure 7.50

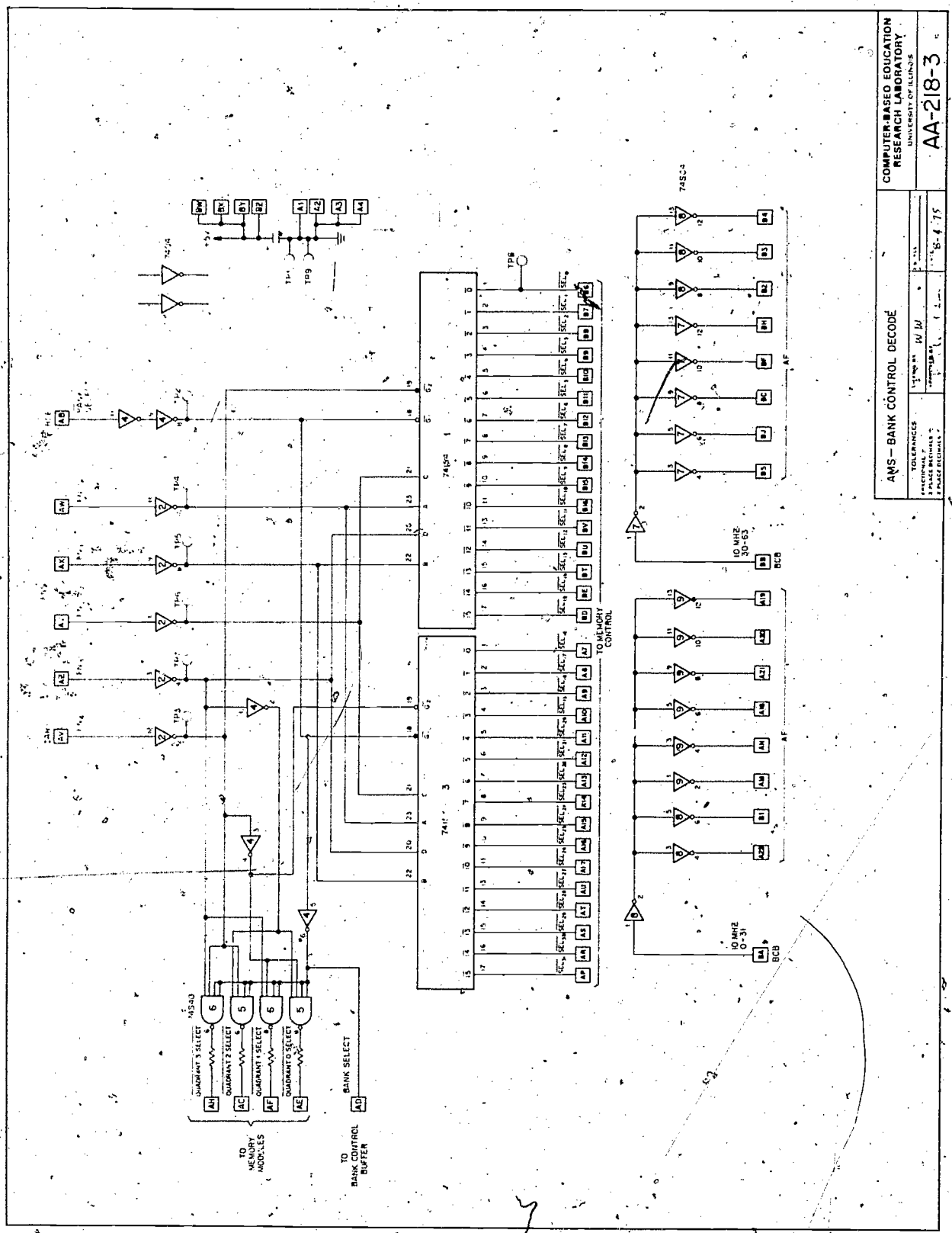

AMS – BANK CONTROL DECODE
COMPUTER-BASED EDUCATION RESEARCH LABORATORY
UNIVERSITY OF ILLINOIS
AA-218-3
TO MEMORY CONTROL
TO MEMORY MODULES
TO BANK CONTROL BUFFER
QUADRANT 3 SELECT
QUADRANT 2 SELECT
QUADRANT 1 SELECT
QUADRANT 0 SELECT
BANK SELECT
10 MHZ 30-03
10 MHZ 0-31
BCB

Figure 7.51

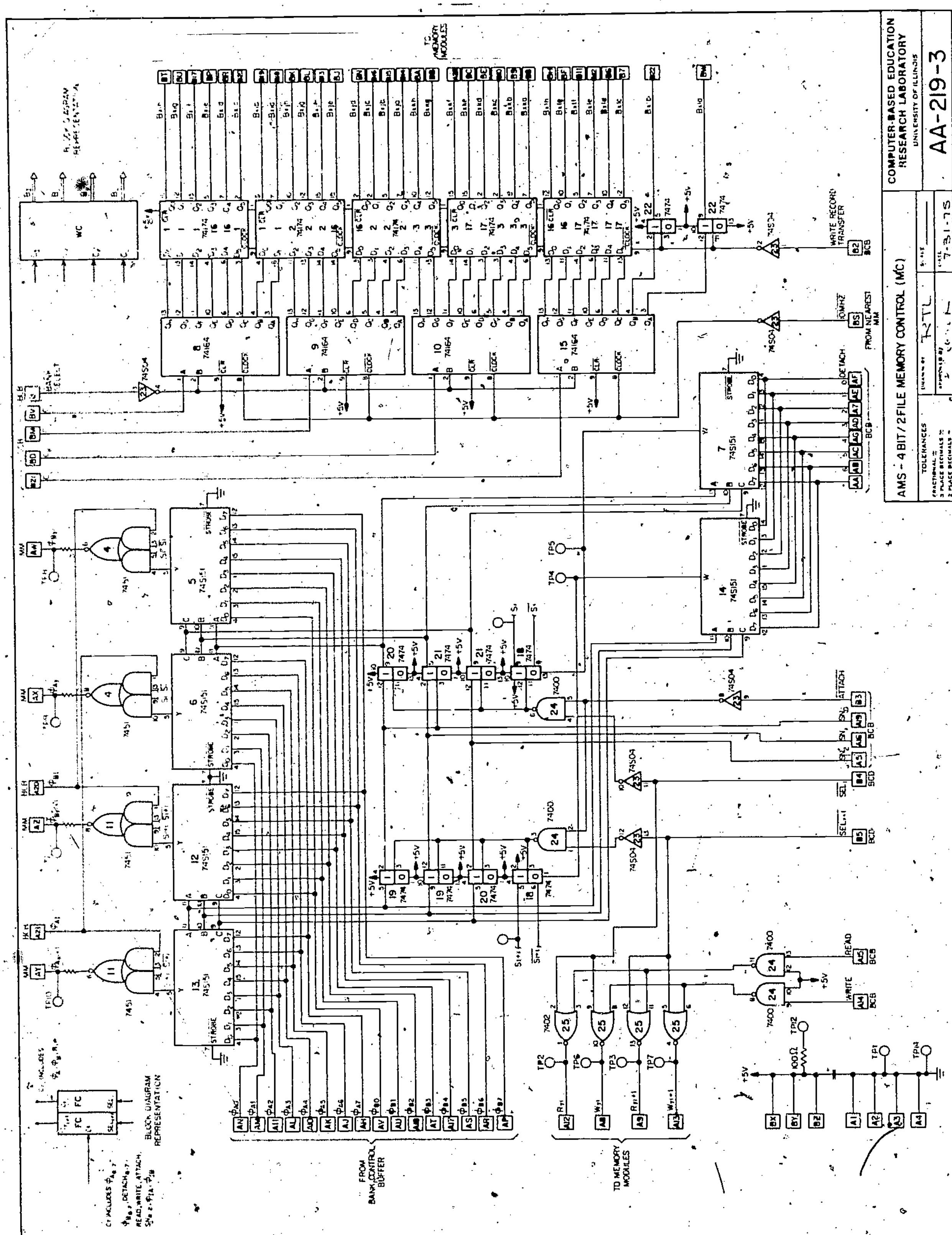

COMPUTER-BASED EDUCATION
RESEARCH LABORATORY
UNIVERSITY OF ILLINOIS
AMS - 4BIT/2FILE MEMORY CONTROL (MC)
AA-219-3
FROM BANK CONTROL BUFFER
TO MEMORY MODULES
BLOCK DIAGRAM REPRESENTATION

Figure 7.52

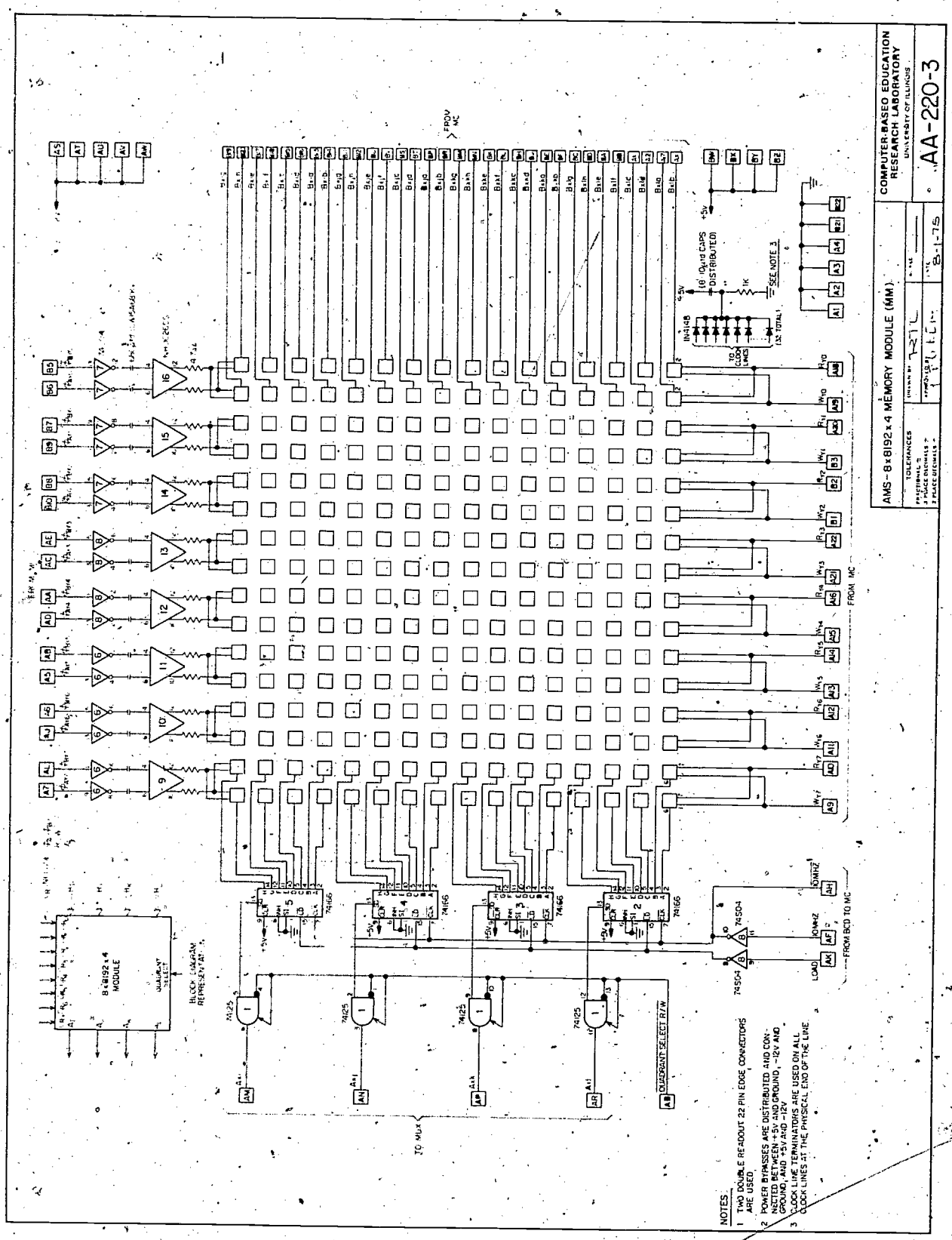

AMS-8x8192x4 MEMORY MODULE (MM)
COMPUTER-BASED EDUCATION RESEARCH LABORATORY
UNIVERSITY OF ILLINOIS
AA-220-3
8-1-75
8x8192x4 MODULE
BLOCK DIAGRAM REPRESENTATION
FROM MC
FROM BCD TO MC
10 MUX
QUADRANT SELECT R/W
NOTES
1 TWO DOUBLE READOUT 22 PIN EDGE CONNECTORS ARE USED.
2 POWER BYPASSES ARE DISTRIBUTED AND CONNECTED BETWEEN +5V AND GROUND, -12V AND GROUND, AND +5V AND -12V
3 CLOCK LINE TERMINATORS ARE USED ON ALL CLOCK LINES AT THE PHYSICAL END OF THE LINE

Figure 7.53

REFERENCES

1. Alpert, D. and D. L. Bitzer, "Advances in Computer-based Education," Science 167, pps. 1582-1590 (March 1970).

2. Bitzer, D. L. and R. L. Johnson, "PLATO--A Computer-based System Used in the Engineering of Education," IEEE Proceedings, Special Issue on Engineering Education 59-6, pps. 960-968 (June 1971).

3. Sherwood, B. A. and J. E. Stifle, "The PLATO IV Communication System," CERL X-44 (July 1975).

4. Control Data Publications, "Control Data 6400/6500/6600 Extended Core Storage Systems, Reference Manual," Pub. No. 60225100, 1968.

5. Collins, D. R., J. B. Barton, D. C. Buss, A. R. Kinetz, S. E. Schroeder, "CDD Memory Options," 1973 IEEE Solid State Circuits Conference, p. 136.

6. Amelio, G. F., "Charge-Coupled Devices for Memory Applications," 1975 National Computer Conference, pps. 515-522.

7. Ypma, J. E., "Bubble Domain Memory Systems," 1975 National Computer Conference, pps. 523-528.

8. Hughes, W. C., C. Q. Lemmond, H. G. Parks, G. W. Ellis, G. E. Possin, and R. H. Wilson, "BEAMOS-A New Electronic Digital Memory," 1975 National Computer Conference, pps. 541-548.

9. Speliotis, D. E., "Bridging the Memory Access Gap," 1975 National Computer Conference, pps. 501-508.

10. Electronic Design 4, February 15, 1975, "CCD Serial-Memory Capacity Climbs to 16-k Bits of Storage," p. 100.

11. Boyle, W. S. and G. E. Smith, "Charge-Coupled Semiconductor Devices," Bell System Technical Journal, Briefs 49, p. 587, April 1970.

12. Karp, H. R., "Magnetic Bubbles--A Technology in the Making," Electronics, September 1, 1969, pps. 83-87.

13. Lemmond, C. O., W. C. Hughes, G. E. Possin, R. H. Wilson, J. K. Fisher, "Design, Fabrication, and Evaluation of an Electron Beam Addressable High Information Density Memory Tube, Final Report," Research and Development Technical Report, United States Army Electronics Command, Fort Monmouth, N. J., No. OSD-1366, 1974.

14. Control Data Publications, "Control Data Cyber 70 Model 73 Computer System," Pub. No. 60347200, 1974.

15. Engineering Staff of Texas Instruments Incorporated, "The TTL Data Book for Design Engineers," Texas Instruments Publications, 1973.

16. Blood Jr., W. R., "MECL System Design Handbook, Second Edition," Motorola Semiconductor Products Inc., 1972.

17. Control Data Publications, "Control Data 6640 A/B Extended Core Storage Controller," Pub. No. 60186000, 1974.

18. Control Data Publications, "Control Data 6642 Distributive Data Path," Pub. No. 60376300, 1974.

19. Control Data Publications, "Control Data 6641A ECS/Mass Storage Adapter," Pub. No. 60334200, 1974.