

AD-A114 029

ROYAL AIRCRAFT ESTABLISHMENT FARNBOROUGH (ENGLAND) F/6 9/2
AN IMPLEMENTATION OF MIL-STD-1750 AIRBORNE COMPUTER INSTRUCTION--ETC(U)
MAY 81 S J SHRIMPSON
RAE-FS-403

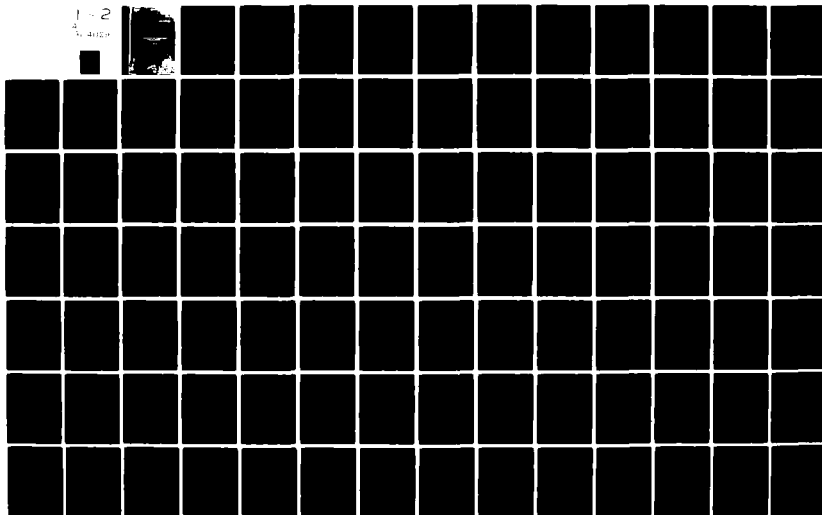
UNCLASSIFIED

DRIC-BR-80502

NL

1-2

4-4000



AD A114029

R O Y A L A I R C R A F T E S T A B L I S H M E N T

Technical Memorandum FS 403

Received for printing 1 May 1981

AN IMPLEMENTATION OF MIL-STD-1750 AIRBORNE COMPUTER
INSTRUCTION SET ARCHITECTURE

by

S. J. Shrimpton

SUMMARY

This Memorandum describes the design of a processor implementing the Mil-Std-1750 Airborne Computer Instruction Set Architecture, using Advanced Micro Devices 2901 bit-slice microprocessor devices. The aspects of the hardware design and microcode specific to Mil-Std-1750 are discussed and reviewed in the light of the experience gained. A full listing of the AMD 'AMDASM' micro assembler definition file and microcode source text is included, together with full hardware documentation.



Copyright
©
Controller HMSO London
1981

Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A	

LIST OF CONTENTS

	<u>Page</u>
1 INTRODUCTION	3
1.1 The Mil-Std-1750 standardisation exercise	3
1.2 RAE/FS5 implementation programme	3
1.3 Current status	4
2 SYSTEM DESIGN PHILOSOPHY	4
2.1 Hardware design	4
2.2 Microcode word definition	8
2.3 Firmware design	8
3 DESIGN REVIEW	9
3.1 Instruction decode features	9
3.2 Arithmetic operations	10
3.3 Microcode word formatting	11
3.4 Interrupt system	11
4 CONCLUDING REMARKS	12
Appendix A Central processor hardware description	13
Appendix B Hardware/microcode interface definition	29
Appendix C AMD 'AMDASM' definition file	37
Appendix D Microcode source text	45
References	120
Illustrations	Figures 1-29
Report documentation page	inside back cover

1 INTRODUCTION

1.1 The Mil-Std-1750 standardisation exercise

Mil-Std-1750, the current version of which is Mil-Std-1750A¹, is an Airborne Computer Instruction Set Architecture standard published by USAF and intended to be applicable to all avionics applications with a few exceptions. Recognised examples of such exceptions are those cases where a very specialised architecture is required such as, for example, in signal processing, and those cases where a single chip microprocessor can fulfil the computing requirement. The need for computer standardisation is well established and need not be set out here.

The approach to standardisation implied by Mil-Std-1750 carries with it several advantages in that it is an instruction set standard rather than a computer standard. This makes it technology independent since it may be implemented using currently available devices to performance specifications defined by the project. The instruction set itself is not proprietary to any private company, being freely available for general use and is thus vendor independent.

The approach taken by USAF in publishing 1750 seems to be similarly enlightened in that the participation of all defence contractors has been actively sought and a mechanism in the form of a User Group has been established to enable input from these companies to materially effect the development of Mil-Std-1750. Indeed, changes to Mil-Std-1750 put forward by participants in the User Group and accepted by the group in a democratic manner have been agreed by the USAF Instruction Set Architecture Control Board and represent the differences between the original 1750 and 1750A.

RAE together with British Industry have been invited to participate in the User Group and RAE has also been invited to sit on the USAF Control Board. This participation is described in an RAE Technical Memorandum² which discussed the potential impact 1750 could have on domestic standardisation policies. It is also felt that 1750 represents a useful starting point for standardisation debates in NATO and ASCC.

1.2 RAE/FS5 implementation programme

The practical work on implementing 1750, described in this Memorandum, has been undertaken by RAE in view of the importance of 1750 both as a standard for possible adoption in the UK and also because of its relevance to British Avionic Companies wishing to bid into the American defence market. The objectives of RAE's work can be summarised as follows:

(i) To gain an appreciation of the features of 1750 from the point of view of implementation and thus to establish a base of specific expertise which will make subsequent implementation by private industry a lower risk enterprise.

(ii) To investigate the areas of computer architecture that remain undefined by 1750, such as internal bus structures and input/output mechanisms, and to assess the relevance of such features in standardisation policy.

(iii) To investigate airborne computer architecture in general and look towards the development of advanced system architectures and bus systems for application to more distant projects.

(iv) To gain general experience in microprogrammed systems and assess their applicability within the wider avionic system to fulfil such intelligent functions as, for example, Mil-Std-1553B data bus control.

This Memorandum describes the development of what is hopefully the first in a range of Mil-Std-1750 computers and which is referred to as the Mk 1 processor. This processor is constructed using AMD 2900 series bit-slice devices³ and exists at present in the form of a development rig constructed around a 'System 29'⁴ development system marketed by Advanced Micro Computers specifically for microprogram development. This is referred to as the Mk 1A rig.

1.3 Current status

The publication of this Memorandum has been unavoidably delayed by circumstances beyond the control of the author who has since left RAE. The continued monitoring of Mil-Std-1750A has led to the development of a single-card Mk 2 processor with considerable enhancement to the Mk 1A design described herein. This work will be reported in due course by Mr D.K. Marshall, Flight Systems Division 5.

2 SYSTEM DESIGN PHILOSOPHY

2.1 Hardware design

The design of an efficient mechanism for implementing an instruction set such as 1750 is almost certain to be an iterative process, starting with the design of a system based on an 'Educated Guess' as to the necessary hardware facilities. Having defined a starting point in terms of hardware and microcode field definitions, the exercise would proceed with the writing of some of the critical areas of microcode, such as the machine instruction fetch cycle and the more commonly used machine instruction implementation sequences. This would result in an understanding of where inadequacies are present and where the hardware design needed to be modified to enable the performance goals to be reached.

In the case of the implementation described in this document however, it was felt that the achievement of a successful, although perhaps inefficient, implementation at an early date was highly desirable in order to provide a facility for the testing of software written for 1750.

In view of this, it was decided to proceed with construction of the hardware and commissioning of a laboratory rig at an early stage and also develop microcode to a full 1750 implementation. Both of these phases of the project represent considerable effort although the use of the 'System 29' development system greatly reduced the manpower investment in the microcode development. For example, at the present time microcode exists to implement all but the double precision and floating point sections of the instruction set. The manpower investment to write this code, implementing actually about 60% of the instruction set, was probably little more than two man months.

The rest of this section will discuss the decisions that were made during the design of the system. The next section will critically examine the design in the light

of further experience, particularly after writing microcode, and identify areas of possible improvement.

At an early stage in the design of the system, decisions were necessary regarding which of the defined 1750 operations would be implemented mainly in microcode and which in hardware. In order to reduce the hardware development time it was decided that the system functions would be implemented mainly by microcode operating in hardware designed to be as versatile as possible. This was felt to be a wise decision in view of the partially undefined state of 1750 at that time. For example, no dedicated floating point hardware was designed into the system, a coded approach being preferred in view of the particular uncertainty of floating point format. For similar reasons, the input/output (I/O) instructions were all code implemented including the interpretation by the machine of the I/O command.

On the other hand, it was decided that it would be impractical for the majority of instructions to implement the selection of registers by code. Dedicated logic was therefore included to unpack the appropriate fields from the instruction register and present them to the 2901 array to address the registers. Also, for efficiency reasons, it was decided to include a mapping PROM to decode the basic 8-bit 1750 opcode and generate a microcode start address. Such an approach combines both versatility and speed.

The handling of opcode extensions (including the I/O command) represented a slight problem since the mapping PROM would have to have been unacceptably deep to accommodate them and the uncertainty in the definition of the I/O command made dedicated logic undesirable. Eventually, the problem was resolved in a way which resulted in reduced performance in these areas of the instruction set but maintained versatility. The method was to use the 'OR' inputs of the 2909 microprogram sequencer to modify a jump address according to a 4-bit field taken from the same unpack logic that selects registers. It was therefore possible to select either of the two defined opcode extension fields of the instruction and also, by shifting and loading of the I/O command into the instruction register to select any of its 4-bit fields.

Turning now to the arithmetic facilities of the machine, it was decided to implement the 16 defined registers in 1750 as the 16 registers internal to the 2901. This meant that register to register operations could be carried out in one micro-instruction. It also meant that those instructions involving a derived operand and a register, with the result to be left in a register, could be accommodated in one cycle provided the derived operand was available at the 'D' input to the 2901.

The need for additional registers was recognised early in the design to fulfill such functions as holding derived operands and providing microcode working space for complex operations. An array of four 29705 16×4 register file devices was therefore included which provided ample space for such purposes. These are referred to as external registers and numbered ERO to ER15. The registers internal to the 2901 chips are referred to as internal registers and are numbered IRO to IR15.

The availability now of spare registers prompted the placing of the machine instruction counter in the external register file as ERO. Also, ER1 and ER2 were

allocated to holding the first and second words of the current machine instruction, ER3 was allocated as the fault register and ER4 was specified as a 'State' register holding bits available only to the microcode and indicating various hardware conditions. ER5, 6 and 7 were specified as holding the first, second and third word respectively of a one, two or three word derived operand. ER8, 9, 10 and 11 were designated as microprogram work space and ER15 was designated for use as a holding register for the interrupt mask.

The bus structure within the Central Processor came about as a result of the desirability of being able to take data from a register within the external file, operate on it within the arithmetic logic unit (ALU) and return it to the same register. This function enables an operation between a 1750 register and a derived operand with the result left in the derived operand, to be carried out in one micro-instruction. Two buses were therefore defined: the A-bus and the D-bus. Only the external register file can drive the A-bus which is used to carry the address when accessing a memory-like device external to the CPU. The D-bus can be driven by a number of devices including the external register file, the 2901 array and the memory data register and is used to carry data when accessing a device on the memory bus.

This brings us to a discussion of the memory/IO bus itself which was specified as a standard interface between the 1750 processor and the rest of the computer system. It was recognised that the 1750 processor should be a versatile device for inclusion within a larger processing system, involving perhaps, other processors, memory systems and intelligent I/O controllers. It was felt, however that it would be inappropriate to choose a multiplex, multi-source, multi-sink parallel bus as the basic interface to the 1750 processor. It was felt that such an approach would make the processor both complicated and inflexible.

Rather, a bus was specified with separate address and data lines, with a simple handshake protocol intended for the exclusive use of the processor. Such a bus, it was felt, would act as a standard interface allowing processor development to go ahead in the absence of any definite multiplex bus policy, whilst leaving the way clear for the processor to be coupled to a multiplex bus via a specialised linker unit.

The memory for the processor was defined with multiple access ports, each conforming to the specification of the processor bus and prioritised by hardware within the memory. This gave another mechanism by which multiple processors could be linked, via common memory block.

When considering the implementation of the status register defined within 1750, it was felt that the AMD 2904, while not ideal for the job, did handle the shift linkage of the ALU neatly and offered the usual benefits of large-scale integration (LSI). Both the 4-bit micro-status register and the 4-bit machine status register are loadable from the D-bus, each other and the 2901 status outputs. Some logic was included on the 2901 outputs to provide the carry, negative, positive and zero status bits defined within 1750. Additional micro-status bits (*viz* most sig byte = 0, least sig byte = 0, overflow and carry) are held within an auxilliary micro-status register. The carry-in multiplexer

within the 2904 is used to select the carry input to the 2901 array. The signals logic 0, logic 1, aux, carry, machine carry and micro carry are available for selection.

The outputs of the auxiliary status register and the output of the test condition multiplexer within the 2904 are available as test conditions for the microprogram sequencer.

The rest of the bits of the status register defined in 1750 are implemented by three AM 2918 devices which are loadable from and may drive onto the D-bus.

Other devices which may drive the D-bus are the fault flag register, the data insertion driver which allows constants from microcode to be inserted as data, the status registers within the machine and micro-interrupt units and the mask registers within the same units.

The interrupt system is arranged on two levels, the micro-interrupt system and the machine interrupt system. When a micro-interrupt is generated, a test condition available to the microprogram sequencer is taken low. This can be tested at a suitable point within the microcode and a conditional jump to a service sequence executed. The micro-interrupts produce a vector that is used as the next microprogram address in a jump vector operation. A vector jump is thus used in the microprogram to go to the appropriate microprogram sequence for each micro-interrupt. At present, micro-interrupts are allocated as follows:

- Level 0 Spare
- Level 1 Machine Interrupt Request (lowest priority group)
- Level 2 Machine Interrupt Request (highest priority group)
- Level 3 Fault Flag Interrupt
- Level 4 Spare
- Level 5 CPU Control Panel Service Request
- Level 6 Spare
- Level 7 Spare.

It can thus be seen that the machine interrupts are divided into two groups each of which provide a micro-interrupt. The 16 machine interrupts are implemented by two 2914 devices, each handling a group of 8. The vectors from these units are available to drive onto the D-bus so that the different actions for each are implemented by firmware rather than hardware. This decision was made because the action for each of the machine interrupts is the same, with the exception of the address at which the present status is stored and the address from which the new status is fetched. It would thus have been extravagant in code to have hardware vectoring to a different sequence in each case.

An additional hardware vectoring mechanism is provided to differentiate between CPU control panel service requests from the 12 push buttons.

The fault interrupt is the OR of all the bits in the fault flag register and enables a code routine to set the appropriate bit in the fault register (ER3) and generate a machine interrupt at the level defined for machine error. Because the fault register is firmware maintained, faults can be set by the microcode which do not involve the fault micro-interrupt.

To generate machine interrupts internally, there is a facility for the data on the D-bus to drive the machine interrupt inputs via an open collector 'Machine Interrupt Bus'.

2.2 Microcode word definition

The microcode word length was chosen as 64 bits which was the longest that could be accommodated using the writable control store that was available. The fields within the word are described in detail in Appendix B and it can be seen that there is considerable sharing of bits between fields. In Appendix B there is a qualifying condition specified for each field interpretation for a group of bits. The qualifying condition must be true before the field contents will have the effects described, thus enabling the bit positions to be shared. For example, the data insertion field is shared with the branch address field and thus a jump to the branch address cannot be carried out at the same time as an arithmetic operation involving the use of a constant from microcode (unless the constant happens to be the same as the branch address, which is very unlikely).

The decisions as to which fields share bits are perhaps some of the most important that have to be made, since the speed of the code is very dependent on which operations can be done in parallel. In the processor described here, no iterations in design were performed in this respect. In other words, experience in writing sections of code was not used to redesign the formatting of the microcode word. As a result, there are a number of fairly glaring deficiencies which give rise to longer code than might perhaps be obtainable with reformatting. It is perhaps doubtful however, if much improvement can be obtained without redesigning parts of the hardware to make control fields shorter or increasing the length of the microcode word. This will be discussed more fully in section 3.

2.3 Firmware design

The aims in writing the firmware (used in this Memorandum to mean microcode) were both to achieve minimum depth (total number of micro-instructions) and minimum number of micro-instructions per machine instruction. The second of these criteria is clearly the most important as far as performance is concerned and the first is important to reduce chip count. In addition, it was thought to be important to make the firmware as structured as possible, in order to make it readily understandable and easily changeable. It should be realised however, that along with these requirements came the constraints imposed by limited programmer resources and the desire for early operation of the system.

Once the hardware has been designed, the microprogrammer is working within constraints. Any deficiencies in the design may cause gross inefficiency in the final complete product in spite of the efforts of the microprogrammer. In the hardware described here there turned out to be a number of defects, but programming technique was still important to achieve the best possible performance within the hardware constraints.

One technique that the microprogrammer has at his disposal is the use of sub-routines. These are instrumental, as in machine code, in reducing depth of code and programmer time. The 2909 microprogram sequencer does not allow nesting to greater than four deep which is a definite constraint on the use of subroutines. Also, the subroutine

call may involve an additional micro-instruction if the call cannot be paralleled with another essential operation. In a sequence involving a small number of micro-instructions the extra one has a significant effect on performance.

Each case must therefore be considered on its merits, taking into account such factors as how much code can be saved, the time penalty involved by the call in the particular instruction and the weighting of that instruction in the mix used to calculate performance. In no way, for example, would one consider putting a subroutine call in an instruction fetch sequence if that call resulted in an additional micro-instruction.

In the 1750 processor the whole of the instruction implementation routines formed a single subroutine. This had the advantage that conditional return statements could be used extensively which did not require the use of the branch address field in the micro-instruction, thus freeing it for use by other fields. Also, the implementation subroutine (execute, as it is called) could be called either from the running state of the machine, or from the halted state via the control panel initiated 'Single Step' function.

The other area in which subroutines were used extensively is the operand fetch sections of the code. Here there was a very distinct advantage since these operations are common to so many instruction implementation sequences.

Turning to structuring, it may be said that to some extent the microcode has intrinsic structure since the route will always be from instruction fetch to one of the instruction execute routines, then back to fetch with perhaps a branch to an interrupt service sequence. The code for the 1750 Mk 1A processor divides naturally into two parts, the instruction implementation sequences and the machine service functions. The machine service functions comprise those functions concerned with the basic machine cycle such as instruction fetch, interrupt service, control panel service, stop, start, single step, etc. The implementation sequences, on the other hand comprise those areas of code that are specific to the implementation of each instruction. A third group perhaps may be defined as the operand fetch routines, although these are called only from the implementation sequences.

3 DESIGN REVIEW

3.1 Instruction decode features

On the whole it is felt that the method of decoding machine level instructions chosen, has worked fairly well and has resulted in a reasonably quick instruction decode. Of course, the method used to accommodate opcode extensions results in an extra two micro-instructions which reduces the speed of these machine instructions. It is mainly base relative indexed and immediate operand instructions that fall into this group and if higher performance for these were required, additional hardware would need to be added to accomplish the decode in a more efficient manner.

The most blatant example of inefficient decoding in the system is clearly the I/O operation commands, where each 4-bit field in the 16-bit command has to be handled separately. Clearly some additional thought is needed to tidy up this situation now that the I/O commands are standardised.

Other parts of the instruction that require handling are the register select fields, the bit select fields, the shift place number fields and the relative branch addresses. These will now be discussed.

On the whole, the register select method was successful although it would have been useful to be able to address all of the registers directly from microcode rather than just the bottom three. In particular it was often required that the stack pointer (R15) be addressed directly from microcode and it was necessary to load the appropriate bit pattern into the instruction register each time this was required. Often it was then necessary to reload the instruction, the whole procedure requiring two extra micro-instructions.

The decoding of the bit position indicator field for single bit operations was particularly inefficient since it required the generation of a bit mask from the 4-bit field. This was done by loading the field into a register, then shifting a mask and decrementing the value of the field until zero was reached. An obvious improvement would be to add a hardware 4 to 16 decoder which would enable this operation to be performed in one micro-instruction.

In order to accomplish multiple shifts using the 4-bit shift place number field, it was necessary to load the field into a register and carry out a shift and decrement sequence, each operation taking two micro-instructions. If the architecture had allowed the loading of the counter in the micro-sequence section from the D-bus, the shift and decrement operation could have been accomplished in one micro-instruction. A further improvement might be to include a multiple shifter programmable from a 4-bit field.

Some difficulty was experienced in sign extending the 8-bit relative address in branch instructions and this turned out to be rather inefficient; some further hardware might make an improvement.

3.2 Arithmetic operations

The single precision Add and Subtract functions were quite straightforward to implement but Multiply involved two micro-instructions for each shift and Add operation. This was because an extra micro-instruction was required to test the least significant bit (LSB). The amount of logic required to carry out the conditional shift or shift and add dependent on the value of the LSB would be quite small, requiring simply an ALU function defined in the appropriate field of the microcode word that would cause the 2901 function control to be dependent on the LSB in the correct way. Similarly logic could be included to reduce the number of micro-instructions required on each cycle of the divide routine.

The AM 2903 presents an attractive way of achieving these aims while also introducing some further useful features such as dual input ports. This latter feature would enable the internal bus system to be configured somewhat differently to enable operations to be performed between two external registers directly. This, in effect gives the external registers the same status as those internal to the bit-slice chip. This would save such time consuming adjustments as placing words from scratch registers temporarily in internal registers in order to use the ALU shifter.

The setting of the status bits and the determination of overflow status required extra micro-instructions in the case of double length operations. Again some micro-programmable hardware to determine how the lines from the ALU generate the status bits would possibly solve the problem.

Although at the present time, the double precision and floating point operations in 1750 have not been implemented, it is apparent that the code to carry them out will be long. This is because of the need to handle the two halves of the operands separately. In the case of the floating point operands, the exponent will need to be masked off and treated separately. The extended precision floating point will require the mantissa to be split into three, and each section, plus the exponent, handled separately. Clearly to make these operations fast, dedicated floating point hardware is required although some improvement would be apparent if the ALU were extended to 32-bits.

3.3 Microcode word formatting

As stated, the choice of a microcode word length of 64 bits was dictated initially by the width of the writable control store available, and it was soon discovered that considerable sharing of bit fields within the word would have to take place in order to accommodate all the control fields required. The problems that subsequently arose were mainly because of the necessity to share various status register control fields (required by the 2904) with the condition select field and the branch address field. This resulted in extra micro-instructions in many places throughout the code. The problem was further compounded by the necessity to share the same field with the data insertion field. This meant that one could not do arithmetic operations with a constant from code at the same time as a branch instruction involving a pipeline address. Neither was it possible to control the status register and shift linkage when the data insertion field was used.

The sharing of the branch address with the 2904 control field also meant that the condition multiplexer within the 2904 could not be used to select the condition for a conditional branch involving a pipeline address. Conditional return and conditional test end-of-loop operations, defined by the 29811 could, however, be carried out. This state of affairs led to the use of the auxiliary status register bits to determine many conditional branches.

This problem can only really be overcome either by simplifying the status register control, perhaps not using the 2904, or by increasing the width of the microcode word. Perhaps the 2904 is not very suitable for the implementation of the 1750 status register.

3.4 Interrupt system

The machine interrupt system, being implemented using 2914 has several shortcomings that have arisen as a result of changes to the 1750 standard due to User Group activity.

The first is the need to be able to load and read the pending interrupt register. Originally, the interrupt lines were connected directly to the 2914 interrupt inputs. However when the requirement to load from code arose, an open collector bus was added at these inputs which could be driven either from devices requesting interrupts or from the D-bus via open collector buffers which would be enabled when a particular bit in the auxiliary clock field was set.

There is no clear way of reading the pending interrupt register from software and this currently remains unimplemented.

Another problem arose because of the requirement not to be able to disable the power down interrupt. 2914 provides interrupt disable facilities but these could not be used since they disable all interrupts. Instead, a dummy interrupt mask was loaded which masked all but power down, the real interrupt mask being stored in one of the 29705 registers. It was necessary therefore, in all instructions referencing the interrupt mask, and in the servicing of interrupts, to test whether the machine was in the enabled or disabled state before carrying out changes to the interrupt mask.

The interrupt system thus became more complicated both in terms of hardware and microcode and the use of the 2914 in the circuit should, perhaps now be questioned.

4 CONCLUDING REMARKS

On the whole it is felt that the work described here has achieved what was planned, that is, to arrive at a working 1750 processor and gain experience on the way. The comments made in section 3 indicate how the use of the various bit slice devices and support chips in the 2900 series has led to a greater understanding of their capabilities and most suitable areas of application.

Of course, bit slice technology is a developing field and new devices are constantly appearing in manufacturers' literature. It is felt that great benefit would accrue from undertaking a complete redesign of the system in the light of experience and making use of the latest developments.

For example, the comments regarding AM 2903, made in the previous section make this a likely contender for inclusion in the design. Also the use of PLA devices to implement some of the MSI logic should be considered along with the use of higher density PROMs for the storage of microcode.

Clearly this represents a major piece of development, a simpler task is the re-partitioning of the present hardware onto perhaps 5U circuit boards to form a self-contained unit, independent of the System 29 Development Rig. It is hoped that this will be carried out in the near future.

Appendix A

CENTRAL PROCESSOR HARDWARE DESCRIPTION

A.1 General description

The Mil-Std-1750 Mk 1A processor occupies two System 29 prototyping boards which slot into the 'User Prototype' area of the System 29 development system.

The two prototype boards are designated CPU board 1 and CPU board 2. CPU board 1 contains the arithmetic unit and general registers, the memory and I/O control logic, the interrupt system and the CPU control box push button implementation logic.

CPU board 2 contains the computer control unit that is responsible for providing during each micro-cycle, the microcode word whose individual bits control all the other sections of the processor. The computer control unit incorporates a microcode sequencer that provides, on each micro-instruction a 12-bit address which is applied to the microcode PROM to derive the 64-bit wide micro-instruction. On CPU board 2 is accommodated 2K words of RAM, 64-bits wide, but the micro-address is available at a connector on this board so that additional PROM or WCS may be connected and control the machine. This facility may be used either to extend the facilities of the machine by writing additional code or during development of the basic 1750 implementation code.

The Mk 1A rig actually uses an AMD writable control store for this purpose, the microcode word passing through an additional board, the pipeline board, before driving onto the micro-instruction bus.

In addition to the micro-sequencer and PROM, CPU board 2 also contains some driver and latches which interface with the CPU control box.

A.2 Central processor board 1

A.2.1 Arithmetic section

A.2.1.1 Bus structure

The arithmetic section is built around a two bus architecture, one being used as the address bus in memory operations and the other being used to carry data. Within the CPU these buses will be referred to as the A-bus and D-bus respectively. Many of the ICs within the CPU can supply data onto the D-bus but only the AM 29705 register file can supply data onto the A-bus, this device being permanently enabled onto this bus.

The control of the D-bus is vested in a 4-bit field in the microcode word which may have 16 possible bit patterns. Each of these patterns causes a particular part of the system to have its output enabled, and thus supply data onto the D-bus. The 4-bit field is decoded on each CPU board and causes an output enable line to the selected part of the system to become active (low) if that part is to supply data. The D-bus runs between boards on the system motherboard and, in order to preserve signal integrity, transceivers are provided (74LS245) at the interface between the section of D-bus on each board and on the motherboard. The 74LS245 has two control lines: pin 1, the direction control determines the direction in which the transceiver will operate; pin 19, the

enable line determines whether the buffers within the transceiver are enabled at all. In the Mk 1A rig, the enable lines of all transceivers are driven via an inverter from the bus control field decoder and are inactive (high) when the bus control field in the micro-code word contains all zeros. The direction line of the transceivers are driven from Nand gates on each board that have as inputs the enable lines to all ICs on that board that are capable of driving the bus. This means that when any IC on a particular board is enabled onto the D-bus, the transceiver on that board will have its direction line held so that it conducts data outwards from that board onto the section of the D-bus on the backplane.

On CPU board 1 the bus control field is decoded by IC5 which supplies enable lines to all the chips on the board capable of driving the bus. IC6 supplies the direction line to the transceivers IC56 and IC70. Since only pit patterns up to 1000 are used on CPU board 1, only a 3-8 line decoder is used, the output for the 1000 pattern being produced from IC3 and IC4.

A.2.1.2 AM 2901 Array

Central to the arithmetic section is the array of four AM 2901 chips which contain an eight function ALU, a 16 deep register file, a general purpose register called the Q register and ALU source and destination data routing logic including single bit left and right shifting facilities for the RAM and Q register inputs.

The 'D' input to the 2901 array comes from a 2-input multiplexer made up from four 748157 chips, the 'select' input to these chips coming from a single bit field in the microprogram word. The inputs to this 2-1 multiplexer come from the D-bus and the A-bus so that the data from either of these two buses may be selected as the 'D' input to the 2901 array.

The 'Y' outputs of the 2901 array are connected directly to the D-bus and can act as source to this bus.

The carry connections between the 2901 chips are handled by a 2902 wired to the 2901s in the conventional manner as outlined in AMD literature. The carry input at the least significant end of the array (IC39 pin 13 and IC58 pin 29) comes from a multiplexer in the AM 2904 (IC43). This multiplexer is controlled by part of the status and shift control field in the microprogram word and can select as its output either logic '0', logic '1', the auxiliary carry latch in IC8 (Cx), the micro-status carry bit in the 2904, the machine status carry bit or the inverse of the latter two alternatives. Control of this multiplexer is described in the 2904 data sheet. The carry output at the most significant end of the 2901 array is one of eight inputs to a multiplexer (IC42) the output of which goes to the auxiliary carry latch in IC8 and to the IC input of the 2904. This input will be loaded into either or both the machine and micro-status registers within the 2904 as determined by the status and shift control field of the microprogram word. Other inputs to this carry select multiplexer are logic '1', logic '0', the RAM shifter MSB output, the Q shifter MSB output, the RAM shifter LSB output and the RAM and Q shifter midpoint outputs. The latter two are provided to facilitate byte

length operation. This carry select multiplexer is controlled by a 3-bit field of the microprogram word.

The shift linkage for the 2901 array is carried out by the 2904 wired in conventional manner as described in the relevant data sheet and thus provides the variety of shift linkages described in the 2904 data sheet by application of the appropriate bit patterns to the status and shift control field in the microprogram word.

Control of the 2901 array is carried out by the application of bit patterns to four separate sets of inputs:

(a) Source control. This is a 3-bit field taken directly from a field in the microprogram word that selects the sources for the two inputs (R and S) to the ALU internal to the 2901 chip. The pairs of sources available are described in the 2901 data sheet and need not be repeated here.

(b) Function control. Again this is a 3-bit field taken directly from the microprogram word that selects one of eight functions to be performed by the ALU internal to the 2901 chips. This is also described in the 2901 data. There are in fact two fields in the microprogram word, each of 3-bits, one supplying function control to the most significant two 2901 chips, the other supplying control to the least significant pair. This division of control facilitates byte operations within the ALU.

(c) Destination control. Again this is supplied directly from a field in the microprogram word and controls the destination within the 2901 chips of the data coming from the ALU output. Details of this can also be found in 2901 data sheets.

(d) Register file output select. These are 4-bit fields which select one of the 16 registers within the 2901 register file for each output of the register file. The outputs of the register file can be selected as ALU sources within the chip by the source control. The register file select fields are not supplied directly from the microprogram word but come from the Mil-Std-1750 instruction stored in the instruction register (IC21 and IC34).

Eight 4-input multiplexers (IC36, IC 35, IC23, IC22) are used to select each 4-bit group from various fields of the 1750 instruction. IC37 and IC38 are used to add a 2-bit modifier to the output of the multiplexers before applying them to the 2901 array. The register select multiplexer inputs are (i) a zero field, (ii) a field made from bits 6 and 7 of the 1750 instruction placed in the least significant 2-bits of the field and 0 and 1 respectively in the most significant and next most significant bit of the field, (iii) the GR1 field - bits 8 through 11 and (iv) the GR2 field - bits 12 through 15.

This logic enables the 2901 internal registers to be selected in a number of different ways to suit the various 1750 instruction formats. The adders enable 0, 1, 2 or 3 to be added to the register address so that 1750 operations referring to RA, RA + 1, RA + 2 and RA + 3 can be accommodated. This facility is necessary for the implementation of double length and floating point instructions.

The register select multiplexers are controlled from a pair of 2-bit fields in the microprogram word, one for register file port A and one for port B. A and B modifiers are supplied from another pair of 2-bit fields in the microprogram word.

A.2.1.3 Scratch registers

In the Mk 1A CPU, the 16 registers defined by Mil-Std-1750 are implemented as the 16 registers in the file internal to the 2901 chips. In the design stage of the CPU, it became clear that extra space would be required by the microcode to hold data that would be invisible at machine level. It was therefore decided to include an array of four AM 29705 chips to provide a 16 deep file of 16-bit wide registers to fulfil this requirement. The 29705 has two ports for which any of the 16 registers may be selected by applying a control field to each of two 4-bit register select inputs. In the Mk 1A CPU, these are connected together and connected directly to a 4-bit field in the microprogram word. The same register is thus always selected to both outputs.

One of the outputs is permanently enabled onto the A-bus and the other may be enabled onto the D-bus.

A.2.1.4 The status register

The machine status register defined in 1750 is implemented partly by the 2904 machine status register and partly by three AM 2918s. The four most significant bits (stored in the 2904) may be loaded, under microprogram control either from the micro-status register, from the D-bus or from four direct inputs from the 2901 array. These inputs are designated carry, positive, zero and negative.

The zero bit is derived simply by adding the MSB = 0 and LSB = 0 lines from the 2901 array. The negative bit comes directly from the F3 output of the most significant 2901 of the array. The carry bit comes from the carry out line of the most significant ALU and the positive bit comes from NORing the negative and zero lines. The other 12 status bits are loadable only from the D-bus.

The 2904 is controlled by a 13-bit field which is described fully in the 2904 data sheet. Twelve of these bits are supplied directly from a field in the microcode word, the other bit, called I10, being taken from a bit in the 2901 destination control field. I10 in fact, determines the direction of shift and is thus determined by 17 of the ALU destination control field.

Writing into either the machine or the micro-status register within the 2904 only occurs when CEm and/or CEmicro respectively are held low. These lines are taken directly from bits in the microprogram word. In addition, writing into the machine status register is also controlled by the individual bit enable lines Ez, Ec, Em, Eovr which are, again, taken from bits in the microcode word. It should be noted that the bit designated as overflow in the 2904 data sheet is, in fact, used as 'positive' bit. Pin 37 of the 2904, called SE (Shift Enable), when taken low, allows the shift linkage outputs of the 2904 to become active and also allows the machine carry bit to be loaded when the appropriate shift linkages are selected. This input to the 2904 is taken from 18 of the 2901

destination control field, which is taken high during shift operations. The signal is inverted by IC50 before being applied to pin 37 of the 2904.

The other 12-bits of the 1750 machine status register are loaded from the D-bus when the bit in the microcode word called 'ENAUxCK' is taken high at the same time as the appropriate bit of the auxilliary clock in the microcode word is taken high (called CKSTATUS).

It can thus be seen that the implementation of the 1750 machine status register is somewhat complicated by the use of the 2904 chip although the arrangement has proved workable.

A.2.1.5 Data insertion from microcode

A 16-bit field in the microcode word is designated as a data insertion field and is enabled onto the D-bus via IC11 and IC12 when the appropriate bus control bit pattern (1000) is present in the microprogram word.

A.2.1.6 Memory/IO interface

In the Mk 1A implementation, IO operations and memory accesses are carried out on the same bus, being differentiated by the holding of the line 'IOreq' low for IO operations. The memory/IO bus has separate sets of lines for address and data together with two command lines which specify the type of bus cycle required and two handshake lines, J and K, that establish the protocol. In the Mk 1A rig there is no provision for devices other than the CPU to request cycles on the bus, DMA from peripherals being handled by the multiport architecture of the memory.

Three-bits in the microcode word are designated to control of the memory/IO bus. These are 'CKPORT' a bit which, when high, causes a bus cycle to take place, C0 and C1 which are bits in the microcode word that drive the command lines of the bus directly.

The protocol and the way in which the CPU circuitry implements this protocol is described fully in other sections. At the appropriate point in any cycle in which the CPU is to write data to the memory or an IO device, the D-bus is enabled onto the memory data bus via IC20 and 33. The address is placed onto the memory address bus via IC24 and 25 from the A-bus. When the cycle is one in which data from the memory or IO device is being delivered to the CPU, the data from the memory data bus is clocked, at the appropriate point in the cycle, into the memory data register made up of IC7 and 32.

A.2.1.7 Fault flag register

IC55, 69, 67 and 53 make up a 16-bit register intended to hold bits supplied by hardware devices indicating the presence of faults as defined in 1750. In the Mk 1A rig only four of the fault bits are implemented by hardware and these lines go to the 'set' inputs of IC29. IC29 therefore acts as a negative pulse catcher, responding to any pulses on the four fault lines long enough to set the latches. The outputs from the latches go to the appropriate bit positions in the fault flag register which is clocked on each system clock pulse. The outputs of the fault flag register may be sourced onto the D-bus when the appropriate bit pattern is present in the bus control field of the

microcode word. The outputs are also ORed together and generate the machine error micro-interrupt.

The fault register defined within 1750 is actually one of the registers within the 29705 register file. Bits may therefore be set or cleared by microcode operations. Some of these operations are initiated by exception conditions arising in the execution of microcode for a particular machine instruction (such as illegal IO command). Others are initiated by the machine error micro-interrupt already mentioned and involve interrogation of the fault flag register to determine which bit is to be set.

A.2.2 Memory/IO bus control section

A.2.2.1 General description

This section comprises the sequencer that controls the protocol that takes place on the memory/IO bus when the microcode requests a bus cycle by setting the CKPORT bit in the microcode word. Because the system clock is affected by the carrying out of a memory/IO bus cycle (in future referred to as simply a memory bus cycle), the circuitry of the memory bus sequencer and the clock generation are somewhat entwined. It will therefore be sensible to describe these areas of the circuitry together.

As is described in the section on bus protocol, there are defined four types of bus cycle:

Read cycle (CO = 1, CI = 0)
Write cycle (CO = 0, CI = 1)
Read-modify-write cycle (CO = 1, CI = 1)
Refresh cycle (CO = 0, CI = 0).

In the Mk 1A rig, neither the read-modify-write cycle nor the refresh cycle are used. The RMW cycle should, in fact, be used for certain operations where DMA must not be allowed between reading from a memory location to the CPU and writing data back to the location.

At the present time, the read-modify-write cycle is not fully implemented.

We now proceed to discuss the operation of the memory access sequencer in detail.

Master oscillator signal, a square wave, enters CPU board 1 at P2/55 at a frequency four times the system clock frequency. The signal is divided by four by IC2 and IC27 produces a pulse which is low for a quarter of a cycle and high for the rest of the time. As long as CKPORT remains low, IC17 remains in the clear state with pin 6 high. This means that the signals at IC27 pin 11 and IC63 pin 3 are inverted versions of the signal at IC27 pin 8. This is the inverted system clock that is distributed throughout the system, along the backplane to each board that requires it. On a given board, the signal must pass through one Schottky gate delay before being used to clock any register. This rule ensures that all set-up and hold times are observed by causing all registers to be clocked at the same time (probably within a couple of nanoseconds, i.e. the spread of delays in two Schottky gates in series). By distributing an inverted system clock and allowing one gate delay before clocking a register, we make it possible to gate the system clock with any given clock enable bit of the microcode word.

If a memory access cycle is to be carried out, CKPORT will be high well before the signal at IC27 pin 8 goes low and thus, when this signal does go low, IC17 will toggle causing the level at pin 6 to go low. Thus although the signal at IC27 pin 8 will go high again after a quarter of a system clock cycle, the inverted system clock distributed around the system will remain high.

If the present input to IC17 were to remain high, IC17 would toggle back on the next negative going edge of the signal at its pin 1 and the system clock would continue as normal having produced one double length cycle. However, with CKPORT high, the positive going edge of inverted system clock at IC27 pin 11 causes a negative pulse to occur at IC65 pin 2 causing the counter, IC1 to be loaded with a value determined by the logical levels of C0 and C1. The counter, IC1 and the 4-16 line decoder, IC19A together form a 16 step sequencer.

The value loaded into the counter is the start address for the sequence required for the type of cycle specified by C0 and C1 and is produced by the logic made up of IC50 and IC28.

Once the sequencer has left step 0, IC19A pin 1 goes high so that the memory address buffers, IC24 and 25 are enabled. These remain enabled until the end of the access cycle although protocol only demands that the address remain valid until K has been taken low by the memory.

Another action that begins whenever the sequencer leaves state 0 is the time out mechanism. When IC19A pin 1 goes high, IC51 pin 12 goes low so that the signal on the cathode of the diode is taken high. During the period when the cathode was low, the 0.01 micro-farad capacitor will have charged. This now begins to discharge through the input resistance of the next inverter and after a few microseconds, the Schmitt inverters change state and the signal at IC64 pin 6 goes high. The latch made up of the two NOR gates in IC64 changes state so that IC30 pin 1 goes low and thus a clear pulse is generated at IC30 pin 3. Once the sequencer has returned to state 0, IC64 pin 3 goes high and the latch resets, ending the clear pulse. Also, the cathode of the diode is taken low so that the capacitor charges to its initial state once again.

Because this time-out circuit begins to operate whenever the sequencer is not on state 0, it also acts as a power-on reset whenever the system powers up on a state other than state 0. The circuitry implementing the time out function just described must be said to be of a temporary nature. It is clear that some redesign work can be carried out to reduce the number of integrated circuits needed to implement the function.

The start addresses for the various types of cycle are as follows:

Read	1001	9
Write	0010	2
RMW	0110	6 (not currently implemented)
RFRSH	0000	0 (not implemented).

Having been loaded with a start address, the sequencer proceeds through a series of steps appropriate to the type of memory access cycle requested, movement from one

point in the sequence to the next being initiated by either a transition of the 'K' line from the memory or by a transition of the system clock coincident with a high value of CKPORT. Which of these events is acceptable as the event initiating stepping is determined by which step the sequencer is moving from. The table gives the characteristics of each step in the sequence including the event needed to trigger stepping on and the value of 'J' and 'K' during the period in which the sequencer waits on each step.

On a step where the memory access sequencer is waiting for a response from the memory, the system clock is held, that is, the signal at IC27 pin 11 and IC63 pin 3 remains high. This is accomplished by the generation of a high at IC19 pin 8 when any of its inputs are low (*ie* when states 1, 2, 4, 6, 8 or 9 are active) which causes the 'preset' input to IC17 to be low so that this flip flop is prevented from toggling back to its normal state (*ie* with pin 6 high) on the next negative going edge at IC17 pin 1. Thus on any state of the sequencer when the CPU is waiting for memory response, the CPU remains on the same micro-instruction until the memory has responded.

The sequencer made up of IC1 and IC19A can be made to change its state in one of three ways:

- (a) By application of a negative pulse to the 'load' input (pin 11), an action that only takes place at the beginning of a memory access cycle.
- (b) By application of a positive edge to the clock input of IC1 (pin 4), an action that is normally used to step the sequencer on.
- (c) By the application of a positive pulse to the clear input of IC1 (pin 14). This is used at the end of the read sequence.

These signals are generated when appropriate by combining signals derived from the K handshake line or the system clock with the outputs of the 4-16 line decoder IC19A. These signals are therefore derived with a knowledge of sequencer state as well as external circumstances.

Let us now examine, in detail, the actions taking place during the execution of the three types of memory access cycle.

A.2.2.2 Detailed description of read cycle

The read cycle begins when, after the new microcode word is loaded into the pipeline register, a high appears on P3/46 the CKPORT line, a low appears on P4/23, the C1 line and a high appears on P4/31, the C0 line. These logic levels are established soon after the upward edge of SYSCK that marks the end of the previous cycle. At a point three-quarters of the way through the cycle, the signal at IC27 pin 8 goes low causing IC17 to toggle into the 'clock hold' state. Very soon after, the edge from IC27 propagates through IC27 and IC50 and, since IC30 pin 5 is high, and IC52 pin 2 is low (due to the sequencer being in state 0), a negative pulse appears at IC65 pin 2 whose length is determined by the delay from IC51 pin 1 to IC30 pin 6. This negative pulse causes the counter (IC1) to be loaded as already described. After the delay through IC19A, its pin 1 goes high and its pin 10 goes low since it will have been loaded with binary value

1001. Since IC19A pin 10 is now low, IC27 pin 3 will be held low so that the preset input of IC17 is held low so that the 'hold clock' state remains as long as the sequencer is at this position.

It can be seen that whenever a memory access cycle takes place, the system clock period is doubled and then extended by the signal from IC27 pin 3 for as long as is necessary. The initial doubling of the system clock period is to allow for the delay from IC50 pin 1 to IC27 pin 3 (ie through IC1 and 19A plus the other gates) which otherwise would be too long for the hold clock state to be established before the next clock edge.

To continue the description of the read cycle, the fact that IC19A pin 1 is now high as are all the other inputs to IC18, results in the J handshake line to the memory going low. The low on IC19A pin 10 causes IC28A pin 12 to go high and thus, when the memory responds by taking its K line low, so that IC30 pin 10 is also high, IC30 pin 8 goes low so that a high appears on IC28A pin 8 causing the sequencer to be counted down to the next state. The high on IC28A pin 8 lasts until the sequencer has moved to its next state (state 8) but the delay in the various gates is sufficient to give a long enough clock pulse.

On state 8, IC19A pin 9 is low so that IC19 pin 5 and 12 are low and the clock hold state continues. As IC19A pin 10 goes high, the data on the memory data bus which was declared by the memory to be stable when J was taken low, is clocked into the memory data register. Also, coincident with the transition from state 9 to state 8, J is taken high as a result of IC18 pin 5 and 12 going low. This is the signal to the memory that the data has been taken and may now be removed.

Since IC19A pin 9 is now low and therefore IC30 pin 13 is high, when the memory takes K high to indicate the completion of the cycle and thus causes IC30 pin 12 to go high, IC30 pin 11 goes low causing IC30 pin 3 to go high. A clear pulse therefore is presented to the counter IC1 and the sequencer returns to state 0. The clear pulse only lasts until the sequencer has left state 8, but the delays are sufficient to ensure that the pulse is long enough to cause the transition. Once the sequencer is back on state 0, the signal at IC27 pin 3 goes high and so the preset input to IC17 is removed. On the next negative edge of the signal at IC27 pin 8 IC17 toggles and the held clock state is relinquished.

A.2.2.3 Detailed description of write cycle

Initially, the only difference between the write cycle and the read cycle is that C0 is low and C1 is high rather than the other way around. The cycle therefore begins in the same manner but the counter is loaded with the binary number 0010 so that the sequencer begins on state 2.

Once in this state, IC19A pin 3 is low so that IC19 pin 2 is low and the same 'clock held' state is established. The fact that all the inputs to IC18 are now high, cause J to go low and indicate to the memory that the cycle has begun. The low on IC19A pin 3 also causes IC20 and 33 to be enabled so that the data on the D-bus appears on the memory data bus.

The stepping of the sequence to the next state takes place when K goes low indicating that the memory has accepted the data. This is because the low on IC19A pin 3 causes a high on IC28A pin 12 and thus on IC30 pin 9 so that the transition is accomplished in the same way as from step 9 to step 8 in the read cycle.

Once on step 1, the low on IC19A pin 2 results in a low on IC18 pin 2 and thus causes J to go high. The memory responds by taking its K line high to indicate the completion of its cycle. Since on step 1 IC31 pin 4 is low, IC31 pin 6 is high placing a high on IC31 pin 13, thus when K goes high taking IC31 pin 12 high, IC31 pin 11 goes low taking IC28A pin 8 high and so clocking the counter. The cycle therefore finishes and the 'clock held' state is relinquished.

A.2.2.4 Detailed description of read-modify-write cycle

The read-modify-write circuitry currently needs some modification and cannot therefore be described here.

Table A1

Table of sequencer states

Step number	Description	Transition event	J value	K value
0	Inactive	Positive edge of system CK when CKPORT is high	High	High
1	Awaiting end of cycle signal from memory	Positive transition of K	High	Low
2	Write cycle has begun. Data has been placed on memory bus by CPU. CPU waits for data taken signal from memory	Negative transition of K	Low	High
3	The read section of the read-modify-write cycle is complete. The memory sequencer is waiting for microcode initiation of step on	Positive transition of system clock with CKPORT high	High	High
4	CPU has signalled that it has finished with the data, CPU is waiting for end of cycle signal from memory	Positive transition of K	High	Low
5	The data from the memory is on bus, memory access sequence is waiting for microcode initiation of step on	Positive transition of system clock with CKPORT high	Low	Low
6	A read-modify-write cycle has started. CPU is waiting for memory to signal data ready by taking K low	Negative transition of K	Low	High
7	Not used			
8	CPU is awaiting end of cycle signal from memory	Positive transition of K	High	Low

Table A1 (concluded)

Step number	Description	Transition event	J value	K value
9	Read cycle has begun. CPU is waiting for memory to indicate data available on bus. When this occurs data is clocked into memory data register and sequencer steps on	Negative transition of K	Low	High
10-15	Not used			

NOTE: At the present time the read-modify-write cycle defined above does not operate correctly as constructed in the Mk 1A rig. This is because it was intended that the data from memory should remain on the memory bus until the microcode requested its removal by using CKPORT. In the Mk 1A rig, this is not required since the data from the memory is clocked into the memory data register at the appropriate time and the read cycle completed before the CPU continues with its microcode.

A.2.3 Interrupt control section

A.2.3.1 General description

The interrupt structure of the Mk 1A rig can be divided into two sections, the micro-interrupt system and the machine interrupt system. The micro-interrupt system is invisible to the machine language programmer and is incorporated to enable such occurrences such as engineering control panel service requests and hardware error situations to be dealt with. In fact, the machine interrupt is implemented as a special kind of micro-interrupt.

The machine interrupt is that interrupt system defined in Mil-Std-1750 and has 16 levels, some of which are defined for machine functions such as error conditions and overflow occurrences.

A.2.3.2 The micro-interrupt system

The micro-interrupt system is constructed around IC44, an AM 2914 vectored priority interrupt encoder. Full details of this device are given in the appropriate AMD data. The control field for the chip is taken directly from the microcode word but does not become effective unless the instruction enable line on the 2914 is taken low. This occurs if the CKMICINT bit in the microcode word is a 1 (this signal occurs at P3/44).

The 3-bit vector output of IC44 is connected to the least significant 3-bits of a 'vector bus' which supplies the address input to the mapping PROM on the AMD computer control unit card. The most significant bit of this 5-bit bus is connected so that it carries the inverse of CKMICINT so that this bit is always zero when IC44 is enabled onto the bus in a RDVC operation. The next most significant bit, bit 3 comes from IC13 pin 4 and is always zero when CKMICINT is high. This means that whenever the vector from IC44 is read, the vector bus carries the pattern 00XXX, where XXX is the vector supplied by the 2914.

IN 15, 100

The interrupt inputs to IC44 come from:

- (a) IC47 which handles the eight top priority machine interrupts.
- (b) IC48 which handles the eight lower priority machine interrupts.
- (c) IC61 pin 8 which is a signal which indicates that a fault flag has been set.
- (d) IC60 pin 8 which is a negative pulse produced when a button is pressed on the engineering control panel of the machine.

The buttons on the engineering control panel are debounced by IC14, 15 and 16 which are each quad latches. The switches on the buttons are two-way and are wired so that when a button is pressed the reset input of the latch is connected to ground and when the button is released, the set input is connected to ground. A long negative pulse is therefore produced from the output of each latch when its button is pressed. IC76 and 77 are used to produce a 4-bit encoded vector indicating which button was pressed. The outputs are tristate and either IC76 or 77 is enabled when IC44 vector output is not enabled. IC76 and 77 are wired so that only one can have its vector output enabled at any one time, IC76 taking precedence if two buttons are inadvertently pressed together. Bit 3 of the vector bus indicates which of IC76 and 77 the vector is, in fact, coming from. The vector on the bus is thus 11YYY, where YYY is a vector produced by IC76 or 10ZZZ where ZZZ is a vector produced by IC77.

When handling service requests from the engineering control panel, two vector jumps are used: the first is to jump to the sequence that handles the micro-interrupt from the control panel, the second is to jump to the appropriate sequence for the button that has been pressed.

When any button on the control panel is depressed, a high is produced at either IC76 pin 19 or IC77 pin 19. A low is therefore produced at IC13 pin 10 for as long as the button remains depressed. This signal is ORed with a delayed and inverted version in IC13 so that a short positive pulse is produced at IC13 pin 13. This is inverted and applied to IC44 as a level 5 micro-interrupt. When IC44 subsequently executes a read vector (RDVC) command, the vector placed on the vector bus will be 00101.

The 'interrupt request' output of IC44 is treated as a test condition for the microprogram sequencer and is examined by the microcode during each machine instruction fetch cycle, a conditional vector jump being executed to the appropriate micro-interrupt service sequence if the 'interrupt request' line is found to be active, i.e. low.

A.2.3.3 Machine interrupt system

The machine interrupt system defined in Mil-Std-1750 has 16 levels of interrupt each of which may be masked or disabled by specialised machine instructions. The response to an interrupt that is not masked or disabled consists of storing the machine status at an area in memory defined by a pointer in the interrupt pointer table and then loading the new machine status from another area of memory defined by a second pointer. The machine status, for this purpose, is defined to be the interrupt mask, the machine status word and the program counter.

The machine interrupt system is implemented by the two AM 2914 ICs, IC47 and 48, the interrupt process itself and the various support instructions being implemented by microcode. IC48 handles the eight highest priority machine interrupts and IC47 the eight lowest. The interrupt request outputs from each of these chips go to separate micro-interrupt inputs on IC44 and hence are handled by separate pieces of microcode. IC48, of course produces a higher priority micro-interrupt than IC47 so that its machine interrupts are serviced first.

The instruction fields for IC47 and 48 come directly from the microcode word and share a field within the word. The instruction does not affect either chip unless its instruction enable line is low, this occurs when the microcode word bit called ENAUXCK is high and the bit in the auxiliary clock field allocated to that IC is high.

The interrupt inputs to both IC47 and IC48 come directly from P5 and have pull up resistors.

The M-bus input/outputs on IC47 and 48 are connected directly to the D-bus such that IC47 occupies the least significant 8-bit positions and IC48 the most significant. The vector output of both these chips is connected to the three least significant bit positions of the D-bus as are the status input/output connections. It is thus important to remember that if either of these ICs is clocked (by enabling via the auxiliary clock field) and the instruction is such as to cause either of these input/output connections to output data, there is an implicit sourcing of either, or both, of these chips onto the D-bus. This action is independent of the bit pattern present in the D-bus control field of the microcode word and it is thus possible to drive the D-bus with two outputs at the same time. To avoid this, and also to be sure that the D-bus transceiver on CPU board 1 does not drive the section of the bus on that board, it is important that the bit pattern 0000 is applied to the bus control field of the microcode word whenever the outputs from IC47 and 48 are active.

A.3 CPU board 2

A.3.1 Control panel latches and drivers

A.3.1.1 Data display latches

The data display latches drive data to the LEDs on the engineering control panel via buffers within the control box. The latches on CPU board 2 are IC5 and IC6 which are AM 2920 chips. The inputs to these come directly from the D-bus section on this board. The latches are clocked simultaneously when the microcode bit ENAUXCK is high together with CKDISP.

A.3.1.2 State display latch

This is IC7 which is a 74S175 and is loaded directly from the D-bus four most significant bits when ENAUXCK and CKSTDISP are both high. The loading actually occurs, of course, on the positive going edge of system clock.

A.3.1.3 Data drivers

These are IC4 and 3 which are octal tristate drivers and enable the output from the data switch field in the control box onto the D-bus when the appropriate bit pattern

is present in the bus control field of the microcode word. IC9 decodes the 4-bit pattern and provides active low outputs which enable devices to be sourced onto the D-bus on this board. The outputs relevant to devices on this board are also ANDed and used to define the direction input to the D-bus transceivers (IC1 and 2), at pin 1, so that they send data outwards from the board when a device on the board is enabled.

A.3.1.4 Register select driver

This is IC8 which simply enables the output from the register select switch field in the control box onto the most significant 4-bits of the D-bus when the appropriate bit pattern is present in the D-bus control field of the microcode word.

A.3.2 Microcode sequencer

The microcode sequencer is built around two AM 2911 and one AM 2909 chips providing a micro-address 12 bits wide. The 2909/11 chips provide the micro-address from the output of a four input multiplexer which can select its output from four micro-address sources. The first is a microprogram counter register which can be loaded on each micro-cycle with the output of the multiplexer incremented by one. A straight sequencing through the code can therefore be accomplished by selecting the output of the program counter at the address multiplexer.

The second input to the multiplexer is the output of a LIFO stack which can be loaded from the microprogram counter output. Selection of the stack output at the address multiplexer causes a return from subroutine.

The third input is the output of an auxiliary register and the fourth is a direct input to the 2909/11 chip. The control lines to the 2909/11 chips come from an AM 29811 which has a 4-bit control field. The bit pattern on this field selects a particular next micro-address generation mechanism and comes directly from the 4-bit next address control field in the microprogram word.

The direct address input to the 2909/11 chips is a tristate bus that has several potential sources, the source of the direct address is controlled by outputs from the AM 29811 which are dependent on the next address control field.

The first source of a direct address is the least significant 12-bits, or branch address field of the microcode word. This may be sourced onto the direct address bus via the drivers IC18 and 1/2 IC20. The second possibility is the output of the mapping PROM (IC19, 36, 38) which generates a microcode address directly from the eight most significant bits of the machine instruction word in the instruction register IC35. IC35 is loaded from the D-bus when P4/41 is taken low, this is connected directly to a bit in the micro-instruction. The third, and last source of the direct address is the 5-bit micro-interrupt vector which may be enabled onto the direct address bus via IC37 and 1/2 IC20.

The control lines coming from IC14 (AM 29811), as well as being dependent on the next address control field, are also dependent on a test condition which is selected by the 16 input multiplexer IC34. The polarity of the test condition may be reversed by the 74S86 so that either the true or false state of the condition may be selected as the requirement for the conditional operation to proceed. The polarity is controlled by a single-bit field in the micro-instruction.

The condition selected by IC34 is determined by a 4-bit field P4/32 through P4/35 and can be one of the following list although others could be added:

- (a) The carry bit stored in the auxilliary status register.
- (b) The MS byte = 0 bit in the same register.
- (c) The LS byte = 0 bit in the same register.
- (d) The overflow bit in the same register.
- (e) Condition multiplexer output of the 2914.
- (f) The interrupt request output of IC44 the micro-interrupt controller.
- (g) The output of IC21 which flags the zero state of a counter.

The counter mentioned above consists of IC22, 23, 24 forming a 12-bit counter that can be loaded from the branch address field of the microcode word. The counter is loaded and decremented under control of lines from IC14 (29811) so that the 29811 instructions which reference a counter are implemented.

When IC14 pins 6 and 7 are low and high respectively, IC13 pin 5 will go low which will cause pin 11 on IC22, 23, 24 to be taken low during the last quarter of system clock cycle. This causes the counter to be loaded from the microcode and also, since the PRESET input of IC21 is taken low, will impose a high on the Q output from this device (the counter zero test line).

Each time pins 6 and 7 on IC14 are held high and low respectively during a micro-cycle, pin 6 of IC13 will be held low. This causes the system clock waveform to appear at pin 4 of IC22 (the count down clock input) so that the counter is decremented on the rising edge of system clock. When the count in IC22 reaches zero, the borrow output (pin 13) is taken low so that when, on the next decrement, the count goes to 1111 and borrow goes high, IC23 will be counted down. Thus the three counters are cascaded. When all the counters are at zero and a further counter enable action takes place, a low appears at IC24 pin 13 so that IC21 is loaded with zero. This then appears as a test condition at the test condition multiplexer. Thus a low on this test condition indicates that in the last micro-instruction the counter was decremented below its zero state. It should be noted that if the repeat file counter equals zero instruction is used with the 29811, the instruction pointed to by the branch address will be repeated twice more than the value previously loaded into the counter.

A.3.3 Microcode PROM

On the CPU board 2, 2K words of PROM are provided, 64-bits wide. This is made up of an array of 32 AM 29775 512 $\times$ 8 PROMS with output registers. These PROMS are arranged in groups of eight, each group implementing 512 words of the microcode memory. The appropriate group is enabled by a low level on a line from a 74S139 2-4 line decoder, its inputs being taken from bits 9 and 10 of the micro-address. Bit 11 of the micro-address is applied to the 'G' input of the 74S139 so that none of the four groups of PROMS are enabled if the micro-address is above the 2K word range.

Since the PROMS have registers on their outputs to pipeline the micro-instruction from the PROM, it is necessary that the output enable signals to the PROMS should also be pipelined. This is accomplished by using the D-type flip flop within the 29775, that pipelines the E2 input.

Appendix B

HARDWARE/MICROCODE INTERFACE DEFINITION

B.1 Introduction

This Appendix described what is effectively the interface between the microcode (or firmware) and the hardware, the interface through which control of all the hardware takes place. In subsection 2 all the fields in the microcode word will be described in detail and the exact operations defined by the various bit patterns will be defined.

As well as defining the bit position within the microcode word for each field, the next section also describes the function of each field and a feature called the qualifying condition. This is the condition that must be satisfied by certain other bits within the microcode word before the field will perform the function described. It is necessary that fields should be controlled by such qualifying conditions in order that bit positions within the microcode word can be used for several fields, only one usually being qualified during any particular micro-cycle.

In section 2, in order to avoid confusion, bit positions within fields and within the microcode word are numbered 0, 1, 2, etc from the least significant end, fields being placed within the microcode word with their least significant end pointing towards the least significant end of the microcode word. Bit position numbers within the microcode word as opposed to within a field are distinguished by prefixing with the letter M.

B.2 Detail definition of field within microcode word

B.2.1 Branch address field

Bit position: M0 through M11.

Function: this field is used to supply a 12-bit branch address (sometimes called the 'pipeline' address) which is used as the next microcode address when either a CJS, CJP, JSRP, JRP, RPCT, CJPP or JP instruction is obeyed by the 2909, 2911 array. It is also used as the data to be loaded into the counter within the AMD computer control unit board in either a PUSH or LDCT instruction is obeyed by the 2909, 2911 array.

Qualifying condition: CJS, CJP, JSRP, JRP, RPCT, CJPP, JP, PUSH or LDCT instruction applied to 'next address control' field.

B.2.2 Status and shift control field

Bit position: M0 through M11.

Function: this field controls the AM 2904 status and shift unit by supplying directly the inputs 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, I11, I12 from the bits within the field.

- 10 = bit 0
- 11 = bit 1
- 12 = bit 2
- 13 = bit 3
- 14 = bit 4

15 = bit 5
 16 = bit 6
 17 = bit 7
 18 = bit 8
 19 = bit 9
 I11 = bit 10
 I12 = bit 11.

Qualifying condition: this field is effective is any of the following conditions are true:

- (a) Bit M53 is low meaning that the micro-status register within the 2904 will be loaded.
- (b) Bit M52 is low and any of bits M20 through M23 are low meaning that some or all of the section of the machine status register within the 2904 will be loaded.
- (c) A shift instruction is being executed by the 2901 array.
- (d) An arithmetic operation is being executed by the 2901 array so that the carry in selected by the multiplexer in the 2904 is applicable.
- (e) The output of the test condition multiplexer in the 2904 has been selected as the test condition for a conditional jump by applying the appropriate bit pattern to the condition test select field.
- (f) The tristate output of the 2904 has been enabled onto the D-bus by applying the appropriate pattern to the bus control field.

B.2.3 Data insertion field

Bit position: M0 through M15.

Function: this field is sourced directly onto the D-bus so that constants can be supplied from microcode. Bit M0 becomes the least significant bit of the data and bit M15 the most significant bit.

Qualifying condition: the bit pattern in the bus control field is that required to source the data insertion field onto the D-bus.

B.2.4 Auxilliary clock control field

Bit position: M0 through M11.

Function: ones or zeros in each position of this field enable or disable the clocking of various registers or subsystems within the CPU as defined below.

Bit 0 - If set to one causes control panel data display to be clocked on the current micro-cycle.

Bit 1 - If set to one causes the control panel state display and any other state latches to be clocked on the current micro-cycle.

Bit 2 - If set to one causes bits 4 through 15 of the machine status register to be loaded from the D-bus.

Bit 3 - If set to one causes the instruction applied to the 2914 dealing with the least significant eight machine interrupts to be obeyed.

Bit 4 - If set to one causes the instruction applied to the 2914 dealing with the most significant eight machine interrupts to be obeyed.

Bit 5 - If set to one causes the fault flag register to be cleared on the current micro-cycle.

Bit 6 - If set to one causes Timer A to be loaded from the D-bus.

Bit 7 - If set to one causes Timer B to be loaded from the D-bus.

Bit 8 - If set to one causes any memory cycle occurring on this micro-cycle to be regarded as an IO cycle.

Qualifying condition: bit M61 (ENAUxCK) is set to one.

B.2.5 Carry select field

Bit position: M12 through M14.

Function: this 3-bit field selects the signal that will be loaded into the machine status word carry bit, the micro-status word carry bit and the auxiliary carry latch (IC8).

The bit patterns select signals as follows:

- 000 - logical one
- 001 - logical zero
- 010 - LSB output of a Q register during shift operation
- 011 - MSB output of RAM during shift operations
- 100 - LSB output of RAM during shift operations
- 101 - carry output from 2901 array
- 110 - selects midpoint on RAM in 2901 array
- 111 - selects midpoint of Q register in 2901 array.

Qualifying condition: this field always affects the loading of the auxiliary carry latch since this is loaded on all micro-cycles irrespective of other control fields. The carry bits of the machine and micro-status registers are only affected if they are enabled for loading. The condition for the machine status is if bit M52 is zero and any of bits M21 through M23 are zero. The condition for loading of the micro-status register is that bit M53 is zero.

B.2.6 Memory/IO bus control

Bit position: M15 and M51.

Function: this 2-bit field controls which type of Memory/IO cycle is initiated during a micro-cycle when CKPORT is high. These 2-bits are separated within the microcode word only for historic reasons. The type of cycle initiated is as follows:

<u>M51</u>	<u>M15</u>	<u>Type of cycle</u>
0	1	read
1	0	write
1	1	read-modify-write
0	0	designated as refresh but not implemented.

Qualifying condition: a one must be present in bit M62 if this field is to have an effect.

B.2.7 Next address control

Bit position: M16 through M19.

Function: this field controls the generation of the next microcode address. Because the system is pipelined, the next address is being generated and the next micro-instruction is being fetched during the execution of the current micro-instruction. Any test condition specified refers therefore to the value that was loaded into the various status registers at the end of the previous micro-instruction. The next address control options are exactly as specified in the data on the computer control unit card included in the system 29 manual.

Qualifying condition: this field is always active, there is no qualifying condition.

B.2.8 Condition test select

Bit position: M20 through M23.

Function: this field controls the test input selected for deciding the result of conditional operations selected by the next address control field. The test conditions that may be selected are as follows:

- 0000 - selects logical zero as test condition
- 0001 - selects the bit in IC8 that is set to one when the previous ALU operation resulted in the most significant byte being zero
- 0010 - as 0001 but applies to least significant byte
- 0011 - selects the bit in IC8 that is set to one when the previous ALU operation resulted in ALU overflow
- 0100 - selects the output of the 2904 multiplexer
- 0101 - selects the micro-interrupt request line, i.e. the interrupt request output from IC44
- 0110 - selects the bit in IC8 that acts as the auxiliary carry latch. This bit is always loaded with the output of the carry select multiplexer
- 1111 - selects the signal that indicates that the counter in the AMD computer control unit has reached zero. This line goes low at the end of the micro-cycle AFTER the one in which the counter actually reached zero.

Qualifying condition: the field contents are only applicable if a conditional operation has been selected by the contents of the next address control field.

B.2.9 Condition polarity select

Bit position: M24.

Function: this 1-bit field selects whether the 'true' state of the above condition or the 'false' state is the one that will result in the conditional operation being carried out.

Bit value 0 - FALSE state tested

Bit value 1 - TRUE state tested.

Qualifying condition: this field will be effective only if a conditional operation has been selected by the contents of the next address control field.

B.2.10 Clock instruction register field

Bit position: M25.

Function: this single-bit field causes the instruction register, that is IC21 and 34 on CPU board 1 and the instruction register on the AMD computer control unit board to be loaded at the end of the current micro-cycle if its value is ZERO.

Qualifying condition: none, this field is always active.

B.2.11 ALU source control

Bit position: M26 through M28.

Function: this controls the selection of the sources to the ALU R and S inputs within the 2901 chips. The effect of the various bit patterns is described fully in AM 2901 data.

Qualifying condition: none, always active.

B.2.12 ALU most significant byte function

Bit position: M29 through M31.

Function: controls the function carried out by the ALU in the most significant 2901 pair in the array. The detail of this is specified in AMD data on 2901.

Qualifying condition: none, always applicable.

B.2.13 ALU least significant byte function

Bit position: M32 through M34.

Function: as section 2.11 but applies to the least significant pair of AM 2901 chips in the array.

B.2.14 ALU destination control

Bit position: M35 through M37.

Function: controls the destination of the data from the ALU in the 2901 chips. This is fully specified in AM 2901 data.

Qualifying condition: none, always active.

B.2.15 External register select

Bit position: M38 through M41.

Function: this 4-bit field, selects which external register (*ie* register in the file external to the 2901 array) is to be either read or written to or both. The binary number placed within the field is the number of the register that is to be selected.

B.2.16 Internal register a select control field

Bit position: M42 and M43.

Function: this field selects the source of the information that will be applied to the 2901 array A input and selects which of the 16 internal registers will appear at port A of the internal register file.

Sources for this 'A register' address are as follows:

<u>Bit pattern in field</u>	<u>Source for A register address</u>
00	Bits 8 through 11 of the instruction stored in the instruction register (<i>ie</i> the GR1 field of the 1750 instruction).
01	Bits 12 through 15 of the instruction stored in the instruction register (<i>ie</i> the GR2 field of the 1750 instruction).
10	Bit 3 of the address (MSB) is set to zero, Bit 2 is set to one and bits 1 and 0 come from bits 5 and 6 of the 1750 instruction. This means that the base register is correctly selected as register 4, 5, 6 or 7 using the base register select field of the 1750 instruction.

Qualifying condition: this field is always active, and is applicable if the register file within the 2901 chips is being used.

B.2.17 Internal register B select control

Bit position: M44 and M45.

Function: identical to 2.15 except that it applies to the 'B' port of the 2901 internal register file.

B.2.18 Internal register A select modifier

Bit position: M46 and M47.

Function: the contents of this field are added to the register A address derived in the manner described in section 2.15 before the address is applied to the 'A' inputs of the 2901 array.

Qualifying condition: none, always active.

B.2.19 Internal register B select modifier

Bit position: M48 and M49.

Function: as section 2.17 but applies to the 'B' port of the internal register file within the 2901 array.

B.2.20 2901 'D' input source select

Bit position: M50.

Function: this single-bit field selects the source of the data applied to the 'D' input of the 2901 array. A zero in this field selects the A-bus as the source of this data (i.e. the output of the external register file). A one in this field selects the D-bus as the source of this data.

Qualifying condition: none, always active.

B.2.21 Status register enable field

Bit position: M52 and M53.

Function: bit M52, if low enables the part of the machine status register in the 2904, subject to which individual bits are enabled by M20 through M23, and as specified by the status and shift control field. Bit M53, if low, enables the micro-status register within the 2904 for loading as specified by the status and shift control field.

Qualifying conditions: none, except as mentioned above.

B.2.22 Main clock control field

Bit position: M58 through M62.

Function: the bits in this field, when set to one, enable the clocking of various parts of the system as specified below:

Bit 0 (M58) causes the 2901 array to be clocked

Bit 1 (M59) causes the external register file to be clocked

Bit 2 (M60) causes the micro-interrupt controller to be clocked

Bit 3 (M61) causes the auxiliary clock field to be enabled

Bit 4 (M62) causes a memory/IO cycle to occur as specified by C0 and C1.

B.2.23 AM 29803 control field

Bit position: M63.

Function: this single-bit field, when set to one, causes the 2901 'A' register address, derived in the way specified by the field described in section 2.15, to be ORed with the least significant 4-bits of the next microcode address. This allows a jump to be performed modified according to the contents of some part of the instruction register.

B.2.24 D-bus control field

Bit position: M54 through M57.

Function: this field controls the sourcing of data onto the D-bus as follows:

<u>Bit pattern</u>	<u>D-bus source</u>
0000	None - transceivers are also disabled.
0001	2901 array sourced onto D-bus.
0010	Sources memory data register onto D-bus.
0011	Sources external register file onto D-bus.
0100	Sources fault flag register onto D-bus.
0101	Sources machine status register.
0110	Sources manual data switch field.
0111	Sources register select switch field.
1000	Sources microprogram data insertion field.
1001	Sources timer A onto D-bus.
1010	Sources timer B onto D-bus.

Qualifying conditions: none, always active.

B.2.25 Micro-interrupt control field

Bit position: M42 through M45.

Function: this field controls the micro-interrupt unit - IC44. The bits of the field are connected directly to the instruction inputs of the 2904. Bit 0 of the field is connected to I0 and Bit 3 to I3.

Qualifying condition: bit M60 must be high before the instruction will have any effect on the 2914.

B.2.26 Machine interrupt control field

Bit position: M46 through M49.

Function: this field controls the operation carried out by either or both of the machine interrupt controllers (IC47 and 48). Bit 0 of the field is connected to I0 of the 2914 instruction and bit 3 to I3.

Qualifying condition: the field affects either or both of IC47 and 48 only if either of bits M3 are high and the ENAUXCK bit M61.

B.2.27 Machine status bit enable

Bit position: M20 through M23.

Function: this field controls the enabling of individual bit loading of the section of the machine status register in IC41. Bits are allocated as follows:

Bit 0 - not enable Z

Bit 1 - not enable C

Bit 2 - not enable N

Bit 3 - not enable P (this bit is allocated to OVR in 2904 data).

The status bit loading is enabled when the appropriate bit is LOW.

Qualifying conditions: bits in the status register are only enabled by this field if the machine status enable - bit M52 is also LOW.

Appendix CAMD 'AMDASM' DEFINITION FILE

TITLE MIL-STD-1750 PHASE 1 FORMAT DEFINITION FIELD MIL-STD-1750 PHASE 1 RIG FORMAT DEFINITION WORD 04		
MICROPROGRAM FIELD FORMAT		
BIT FUNCTION	CONDITION FOR ACTION	
M0		
M1	CE MAC STATUS OR CK MICRO STAT. ACTIVE (LUN) OR SHIFT FUNCTION SELECTED BY ALU.	
M2	(2) 12 BIT BRANCH ADDRESS	
M3	(2) BITS 0 TO 11 OF 16 BIT DATA FIELD	
M4	(4) AUX CLOCK CONTROL FIELD	
M5	M0 - CE DISPLAY	
M6	M1 - CE STATUS	
M7	M2 - CE MACB. INT.	
M8		
M9		
M10		
M11		
M12	(1) CARRY SELECT FIELD	ALWAYS AFFECT. BY CAPM LATCH
M13		MACB. CARR. OR MIC. CARR.
M14	(2) BITS 12 THROUGH 14 OF DATA FIELD	IF CORRESPONDING ENABLE BITS EFFECTIVE IF DATA FIELD IS ENABLED
M15	(1) 1/0 PORT CONTROL BIT 1 (2) BIT 15 OF DATA FIELD	ARE ACTIVE. EFFECTIVE IF CE PORT HIGH AFFECTIVE IF DATA FIELD IS SOURCED ONTO D BUS
M16	NEXT ADDRESS CONTROL	ALWAYS APPLICABLE
M17		
M18		
M19	(1) CONDITION TEST SELECT (2) NOT ENABLE Z - M20	TEST COND. REQUIRED BY AM9611
M20	NOT ENABLE C - M21	MACHINE STATUS ENABLED
M21	NOT ENABLE N - M22	
M22	NOT ENABLE P - M23	
M23	CONDITION POLARITY TEST	APPLICABLE IF TEST CONDITION REQUIRED BY AM 2011
M24	CE INSTRUCTION REG	ALWAYS APPLICABLE
M25		
M26	ALU SOURCE CONTROL	ALWAYS APPLICABLE
M27		
M28		
M29	ALU MS BYTE FUNCTION	ALWAYS APPLICABLE
M30		
M31		
M32	ALU LS BYTE FUNCTION	ALWAYS APPLICABLE
M33		
M34		
M35	ALU DESTINATION CONTROL	ALWAYS APPLICABLE
M36		
M37		
M38	EXTERNAL REGISTER SELECT	APPLICABLE IF EX-REG IS SOURCED
M39		ONTO D BUS OR 2961 D INPUT
M40		ON IF RAG IS Clocked
M41		
M42	(1) INTERNAL REGISTER A SELECT CONTROL	APPLICABLE IF REG A IS USED
M43	(2) MICRO INTERRUPT CONTROL BITS 1 AND 2	APPLICABLE IF CK MIC. INT. IS HIGH.
M44	(1) INTERNAL REGISTER F SELECT CONTROL	APPLICABLE IF REG F IS USED
M45	(2) MICRO INT. CONTROL BITS 3 AND 4	IS HIGH
M46	(1) INTERNAL REGISTER A SELECT MOD.	APPLICABLE IF REG A IS USED
M47	(2) MAC-INT-CONTROL BITS 1 AND 2	IS HIGH
M48		
M49	(1) INTERNAL REGISTER B SELECT MOD.	APPLICABLE IF REG B IS USED
M50	(2) MAC-INT-CONTROL BITS 3 AND 4	IS HIGH
M51	MAC D INPUT SOURCE SELECT	APPLICABLE IF D INPUT IS SELECTED AS R OR S SOURCE FOR ALU
M52	PORT CONTROL BIT 2	APPLICABLE IF CK PORT IS HIGH
M53	MACB. STAT. NOT ENABLE	ALWAYS APPLICABLE
M54	MICRO STATUS NOT ENABLE	ALWAYS APPLICABLE
M55	D BUS SOURCE CONTROL FIELD	ALWAYS APPLICABLE
M56		
M57		
M58	CE ALU	ALWAYS APPLICABLE
M59	CE REG	ALWAYS APPLICABLE

TM FS 400

```

IM30 CK MIC. INTERRUPT          ALWAYS APPLICABLE
IM31 ENABLK AUI CLOCK FIELD     ALWAYS APPLICABLE
IM32 CK PORT                     ALWAYS APPLICABLE
IM33 AM2000S CONTROL           ALWAYS APPLICABLE

TITLE TMR AM2000 FAMILY M M M O N I C S
*****

13 DECEMBER 1976 JBM
UPDATED SYPT 26, 1977

IM2001 INSTRUCTION SET

; REGISTER DEFINITIONS
R0: EQUH#0
R1: EQUH#1
R2: EQUH#2
R3: EQUH#3
R4: EQUH#4
R5: EQUH#5
R6: EQUH#6
R7: EQUH#7
R8: EQUH#8
R9: EQUH#9
R10: EQUH#A
R11: EQUH#B
R12: EQUH#C
R13: EQUH#D
R14: EQUH#E
R15: EQUH#F

; AM2001 SOURCE OPERANDS (R S)
A0: EQUH#0
A1: EQUH#1
A2: EQUH#2
A3: EQUH#3
A4: EQUH#4
A5: EQUH#5
A6: EQUH#6
A7: EQUH#7

; AM2001 ALU FUNCTIONS (R FUNCTION S)
ADD: EQUH#0
SUB: EQUH#1
SUS: EQUH#2
OR: EQUH#3

; ALWAYS APPLICABLE
IM2001 DESTINATION CONTROL
O0: EQUH#0
O1: EQUH#1
O2: EQUH#2
O3: EQUH#3
O4: EQUH#4
O5: EQUH#5
O6: EQUH#6
O7: EQUH#7

; AM2001 INSTRUCTION SET
JZ: EQUH#0; JUMP TO ADDRESS ZERO
CJS: EQUH#1; CONDITIONAL JUMP TO SUBROUTINE WITH JUMP
ADDRESS IN THE PIPELINE REGISTER
JMP: EQUH#2; JUMP TO ADDRESS AT MAPPING FROM OUTPUT
CJP: EQUH#3; CONDITIONAL JUMP TO ADDRESS IN PIPELINE
REGISTER
PUSH: EQUH#4; PUSH STACK AND CONDITIONALLY LOAD COUNTER
JSP: EQUH#5; JUMP TO SUBROUTINE WITH STARTING ADDRESS
CONDITIONALLY SELECTED FROM THE AM2011
IN-REGISTER OR PIPELINE ADDRESS
CJF: EQUH#6; CONDITIONAL JUMP TO VECTOR ADDRESS
JRF: EQUH#7; JUMP TO ADDRESS CONDITIONALLY SELECTED FROM
M-REGISTER OR PIPELINE REGISTER
MPC: EQUH#8; REPEAT LOOP IF COUNTER IS NOT EQUAL TO ZERO
MPC: EQUH#9; REPEAT PIPELINE ADDRESS IF COUNTER IS NOT
EQUAL TO ZERO
CRTM: EQUH#A; CONDITIONAL RETURN FROM SUBROUTINE
CJFP: EQUH#B; CONDITIONAL JUMP TO PIPELINE ADDRESS AND POP
STACK
LCT: EQUH#C; LOAD COUNTER AND CONTINUE
LOOP: EQUH#D; END OF LOOP
CONT: EQUH#E; CONTINUE TO NEXT ADDRESS
JP: EQUH#F; JUMP TO PIPELINE REGISTER ADDRESS

; AM2014 INSTRUCTION SET
MCLR: EQUH#0; MASTER CLEAR
CLRIM: EQUH#1; CLEAR ALL INTERRUPTS
CLMB: EQUH#2; CLEAR INTERRUPTS FROM M-BUS
CLMR: EQUH#3; CLEAR INTERRUPTS FROM MASK REGISTER
CLVC: EQUH#4; CLEAR INTERRUPT FROM LAST VECTOR READ
RST: EQUH#5; RESTART STATUS REGISTER
RDM: EQUH#6; READ MASK REGISTER
SETH: EQUH#7; SET MASK REGISTER

```

```

LSTA: EQU 00000000; LOAD STATUS REGISTER
BCLM: EQU 00000001; BIT CLEAR MASK REGISTER
RSTW: EQU 00000002; RESET MASK REGISTER
CLR: EQU 00000003; CLEAR MASK REGISTER
DISW: EQU 00000004; DISABLE INTERRUPT REQUEST
LPM: EQU 00000005; LOAD MASK REGISTER
RPM: EQU 00000006; ENABLE INTERRUPT REQUEST

:INTERNAL REGISTER SELECT CONTROL PATTERNS
:-----
GR1: EQU 00000000; SELECTS BITS 0 THROUGH 11 OF INSTRUCTION WORD AS
GR2: EQU 00000001; REGISTER SELECT
BASE: EQU 00000002; SELECTS BASE REGISTER USING BITS 5 AND 6
RZERO: EQU 00000003; SELECTS REGISTER ZERO

: CARRY SELECT CONTROL PATTERNS
:-----
CONE: EQU 00000000; SELECTS LOGICAL ONE AS CARRY TO BE STORED
CZERO: EQU 00000001; SELECTS LOGICAL ZERO AS CARRY TO BE STORED
COLSB: EQU 00000002; SELECTS LSB OUTPUT OF Q REGISTER AS CARRY
CMCSB: EQU 00000003; SELECTS MSB OUTPUT OF Q REGISTER AS CARRY
CLLSB: EQU 00000004; SELECTS LSB OUTPUT OF RAM AS CARRY
CCAR: EQU 00000005; SELECTS CARRY OUTPUT FROM ALU
CRMD: EQU 00000006; SELECTS RAM MIDPOINT
CQMD: EQU 00000007; SELECTS Q MIDPOINT

:NB. THERE ARE THREE CARRY LATCHES IN THE MK ONE IMPLEMENTATION.
:ONE IS INTERNAL TO THE 2094 AND IS CLOKED ON EVERY MICROCYCLE.
:THE OTHERS ARE THE CARRY BITS IN THE MICRO AND MACHINE STATUS REGISTERS
:WHICH ARE WITHIN THE 2094 CHIP AND ARE ONLY CLOKED WHEN THE APPROPRIATE
:ENABLE LINE IS HELD LOW AND THE CORRECT INSTRUCTION IS PLACED ON THE
:STATUS AND SHIFT CONTROL FIELD OF THE MICROINSTRUCTION WORD.

:BUS CONTROL PATTERNS
:-----
:THESE CONTROL WHICH DEVICE HAS ITS OUTPUT SOURCED ONTO THE INTERNAL
:TRISTATE DATA BUS DURING THE CURRENT MICROCYCLE.
:ONLY ONE DEVICE CAN BE SOURCED AT A TIME.

:SOURCE: EQU 00000000; DISABLES ALL CPU OUTPUTS TO D BUS
:STATUS: EQU 00000001; SOURCES ALU ONTO D BUS
:DATA: EQU 00000002; SOURCES MEMORY DATA REGISTER ONTO D BUS
:SPC: EQU 00000003; SOURCES REGISTER FILE ONTO D BUS
:FAULT: EQU 00000004; SOURCES FAULT REGISTER

:SOURCE: EQU 00000005; SOURCES MACHINE STATUS REGISTER
:STATUS: EQU 00000006; SOURCES MANUAL DATA INPUT
:DATA: EQU 00000007; SOURCES MANUAL REGISTER SELECT FIELD
:SPC: EQU 00000008; SOURCES MICROPROGRAM DATA FIELD
:FAULT: EQU 00000009; SOURCES CPU STATE DISPLAY
:SOURCE: EQU 0000000A; SOURCES TIMER A
:STATUS: EQU 0000000B; SOURCES TIMER B

:TEST CONDITION SELECT PATTERNS
:-----
:THESE CONTROL THE SOURCE OF THE TEST INPUT TO THE 20811 NEXT ADDRESS
:CONTROL UNIT.

TZERO: EQU 00000000; INPUTS LOGICAL ZERO ON TEST INPUT
ZMSB: EQU 00000001; TESTS FOR ZERO MS BYTE
ZLSB: EQU 00000002; TESTS FOR ZERO LS BYTE
ONE: EQU 00000003; TESTS FOR OVERFLOW SET
MUL: EQU 00000004; TESTS FOR OUTPUT OF 402084 MULTIPLIER
INTPT: EQU 00000005; TESTS MICROINSTRUCTION REQUEST LINE
CARR: EQU 00000006; TESTS OUTPUT OF EXTERNAL CARRY LATCH
COUNT: EQU 00000007; TESTS COUNTER ZERO

:TEST POLARITY SELECT PATTERNS
:-----
TRUE: EQU 00000000
FALSE: EQU 00000001

:ID INPUT TO 2081 SELECT CONTROL PATTERNS
:-----
ABUS: EQU 00000000; SELECTS THE ADDRESS BUS (OUTPUT OF EXTERNAL REGISTER FILE)
DBUS: EQU 00000001; SELECTS THE INTERNAL TRISTATE DATA BUS

:FORMAT DEFINITIONS
:-----
ADDRESS
*****

:THIS FORMAT IS USED TO PLACE A BRANCH ADDRESS IN BITS 0 THRU 11
:OF THE MICROINSTRUCTION FOR USE IN DETERMINING THE NEXT ADDRESS IN
:A JP, CJP, CJS, JSRP, JRP, RPT, CJP, OR LDC INSTRUCTION.
:VARIABLE FIELD SUBSTITUTIONS:
:(1) A BINARY NUMBER NOT GREATER THAN 12 DIGITS OR A LABEL.
:(THE NUMBER WILL BE RIGHT JUSTIFIED AND TRUNCATED)

```

TM ES 400

```

OVERLAY REQUIREMENTS:
THIS FORMAT MAY NOT BE OVERLAID WITH
STATSHT, DATMSHT, ENAUICK, AUICK
ADDRESS: DEF 521,12V;

STATSHT
*****

THIS FORMAT IS USED TO ENABLE THE BINARY PATTERN CONTROLLING THE AM2004
CHIP TO BE INSERTED INTO THE MICROPROGRAM WORD.

VARIABLE FIELD SUBSTITUTIONS:
(1) A 12 BIT BINARY NUMBER FORMING THE CONTROL
WORD FOR THE 2004 CHIP.
    Bit 11 (MSB) - 112
    10 - 111
    9 - 10
    8 - 18
    ....
    0 - 19

SEE AM2004 DATA SHEET FOR DEFINITION OF 10 TO 112.

OVERLAY RESTRICTIONS:
CANNOT BE OVERLAID WITH:
ADDRESS, DATMSHT, ENAUICK, AUICK.

STATSHT: DEF 521,12V

CARRYSEL
*****

THIS FORMAT IS USED TO INSERT INTO THE MICROPROGRAM WORD THE BIT PATTERN
THAT SELECTS THE SOURCE OF THE CARRY BIT THAT IS STORED IN THE
CARRY BITS OF THE MACHINE AND MICROSTATUS REGISTERS IF THEY ARE ENABLED.
IT ALWAYS DETERMINES WHAT IS STORED IN THE CARRY LATCH EXTERNAL
TO THE 2004 ON EVERY CLOCK CYCLE.

VARIABLE FIELD SUBSTITUTIONS:
(1) A THREE BIT BINARY PATTERN APPROPRIATE TO SELECT THE REQUIRED
INPUT TO THE CARRY SELECT MULTIPLIER (IC 3 CPU BOARD II)
THIS WILL NORMALLY BE SUPPLIED BY USING ONE OF THE CONSTANTS
DEFINED UNDER "CARRY SELECT CONTROL PATTERNS".

OVERLAYING RESTRICTIONS:
MUST NOT BE OVERLAID WITH
DATMSHT

```

```

CARRYSEL: DEF 401,3V,12X
DATMSHT: DEF 401,10V,100

RFSH
****

THIS FORMAT PRODUCES THE APPROPRIATE PATTERN IN BITS 15 AND 51 OF THE
MICROPROGRAM WORD SO THAT A REFRESH CYCLE IS GENERATED THROUGH
THE I/O PORT IF BIT 62 (REFRESH) IS HELD HIGH.

THERE ARE NO VARIABLE FIELDS.

OVERLAY RESTRICTIONS:
MAY NOT BE OVERLAID WITH DATMSHT, READ, WRITE, RWRT.

RFSH: DEF 121,800,351,800,15X

READ
****

THIS FORMAT IS THE SAME AS RFSH EXCEPT THAT A READ CYCLE IS INITIATED.

OVERLAY RESTRICTIONS:
MAY NOT BE OVERLAID WITH
DATMSHT, RFSH, WRITE, RWRT.

READ: DEF 121,801,35X,800,15X

WRITE
****

THIS FORMAT IS THE SAME AS RFSH EXCEPT THAT A WRITE CYCLE IS PRODUCED.

OVERLAY RESTRICTIONS:
MAY NOT BE OVERLAID WITH
RFSH, READ, RWRT.

WRITE: DEF 121,800,35X,801,15X

RWRT
****

THIS FORMAT IS THE SAME AS RFSH EXCEPT THAT A READ/MODIFY/WRITE
CYCLE IS PRODUCED.

OVERLAY RESTRICTIONS:
MUST NOT BE OVERLAID WITH

```

DAYINSTR, RPNR, READ, WHITE.	
DEF 121,201,351,301,151	
MACP	

THIS FORMAT ALLOWS THE NEXT ADDRESS CONTROL SECTION OF THE MICROCODE WORD TO BE SET UP.	
VARIABLE FIELD SUBSTITUTIONS:	
(1) A 1 BIT BINARY FIELD. THIS SHOULD BE SET TO A ONE IF THE 2901 BRANCH ADDRESS MODIFICATION FACILITY IS REQUIRED.	
(2) A ONE BIT BINARY FIELD. THIS SHOULD BE SET TO A ONE IF A CONDITIONAL JUMP IS BEING PERFORMED AND THE CONDITION TO BE TESTED IS THE TRUE STATE OF SELECTED CONDITION INPUT. SHOULD BE SET TO ZERO. THIS CAN BE ACHIEVED BY USING THE PREVIOUSLY DEFINED CONSTANTS 'TRUE' AND 'FALSE'.	
(3) A FOUR BIT BINARY FIELD WHICH SPECIFIES WHICH OF SIXTEEN TEST CONDITION INPUTS ARE TO BE USED TO DETERMINE THE ACTION WHEN A CONDITIONAL JUMP IS REQUIRED.	
THE CONSTANTS DEFINED IN THE 'TEST CONDITION SELECT PATTERNS' SECTION CAN BE USED HERE.	
(7240, 2MSB, 2LSB, OVR, MUL, INTRPT, CARR, COUNT)	
(4) A FOUR BIT BINARY FIELD THAT SPECIFIES THE INSTRUCTION TO THE AM2901 NEXT ADDRESS CONTROL UNIT.	
THE CONSTANTS DEFINED IN THE AM2901 INSTRUCTION SET CAN BE USED HERE.	
OVERLAY RESTRICTIONS:	
MUST NOT BE OVERLAPED WITH MACSTEM IF A CONDITION TEST IS SPECIFIED.	
DEF 17000,371,801,141,471,470,1110,101	
MACP	

THIS IS THE SAME AS MAC EXCEPT THAT BIT 25 IS SET TO 0 RATHER THAN 1. THIS CAUSES THE INSTRUCTION REGISTER TO BE CLOKED EITHER ON THE CURRENT CYCLE IN THE PIPELINED ARCHITECTURE OR ON THE NEXT CYCLE IN THE NON PIPELINED HARDWARE.	
DEF 17000,371,800,141,471,470,1110,101	
ALU	

THIS FORMAT ENABLES THE BIT PATTERNS THAT CONTROL THE ALU TO BE INSERTED INTO THE MICROPROGRAM WORD.	
VARIABLE FIELD SUBSTITUTIONS:	
(1) A SINGLE BINARY BIT. THIS BIT SHOULD BE SET TO 0 IF THE MICROSTATUS REGISTER IS TO BE CLOKED AND ONE IF NOT.	
THIS FIELD DEFAULTS TO BINARY ONE.	
(2) A SINGLE BINARY BIT. THIS BIT SHOULD BE SET TO 0 IF THE MACHINE STATUS REGISTER IS TO BE CLOKED AND ONE IF IT IS NOT.	
THIS FIELD DEFAULTS TO BINARY ONE.	
(3) A SINGLE BINARY BIT. THE CONSTANTS DBUS OR ABUS SHOULD BE INSERTED HERE TO SPECIFY WHETHER THE DATA BUS OR THE OUTPUT OF THE INTERNAL REGISTER FILE IS SELECTED AS THE INPUT TO THE AM2901 DATA INPUT.	
(4) A THREE BIT BINARY FIELD SPECIFYING THE DESTINATION OF THE DATA LEAVING THE 2901 ALU, WITHIN THE 2901 CHIP.	
ONE OF THE CONSTANTS DEFINED IN THE SECTION CALLED AM2901 DESTINATION CONTROL SHOULD BE USED HERE.	
(5) A THREE BIT BINARY FIELD SPECIFYING THE ALU LS BYTE FUNCTION.	
ONE OF THE CONSTANTS DEFINED IN THE SECTION CALLED AM2901 ALU FUNCTIONS SHOULD BE USED HERE.	
(6) A THREE BIT BINARY FIELD SPECIFYING THE ALU MS BYTE FUNCTION. THE CHOICE OF SUBSTITUTIONS IS AS IN (5).	
(7) A THREE BIT BINARY FIELD SPECIFYING THE ALU SOURCE CONTROL. THE SUBSTITUTION SHOULD BE CHOSEN FROM THE CONSTANTS DEFINED IN THE SECTION CALLED AM2901 SOURCE OPERANDS.	
OVERLAY RESTRICTIONS:	
CAN BE OVERLAPED WITH ANY OTHER FORMAT.	
DEF 101,17001,17001,11,111,121,351,351,201	
ALU	

THIS FORMAT ALLOWS THE BIT PATTERNS THAT CONTROL THE SELECTION OF REGISTERS TO BE INSERTED INTO THE MICROPROGRAM WORD.	
VARIABLE FIELD SUBSTITUTIONS:	
(1) A TWO BIT FIELD THE CONTENTS OF WHICH IS ADDED TO THE BIT PATTERN BEING USED TO SELECT 2901 INTERNAL REGISTER 8.	
(2) A TWO BIT FIELD THE CONTENTS OF WHICH ARE ADDED TO THE BIT PATTERN	

TM FS 403

```

(3) A TWO BIT FIELD USED TO CONTROL THE SELECTION OF 2081 INTERNAL
REGISTER A. THE SUBSTITUTION MAY BE CHOSEN FROM THE
CONSTANTS DEFINED IN THE SECTION CALLED "INTERNAL REGISTER SELECT
CONTROL PATTERNS".

(4) A TWO BIT FIELD USED TO CONTROL THE SELECTION OF 2081 INTERNAL
REGISTER A. THE SUBSTITUTION MAY BE CHOSEN FROM THE CONSTANTS
DEFINED IN THE SECTION CALLED "INTERNAL REGISTER
SELECT CONTROL PATTERNS".

(5) A FOUR BIT BINARY PATTERN USED TO SELECT THE SCRATCH (OR EXTERNAL)
REGISTER. THE CONSTANTS DEFINED IN THE SECTION CALLED REGISTER
DEFINITIONS (REG.1 ETC.) CAN BE USED HERE.

OVERLAY RESTRICTIONS:
MUST NOT BE OVERLAPED WITH
MICINT, MACINT.

REG: DEF 163.2H,2H,2H,2H,2H,4H,30H

BUS:
=====
THIS FORMAT ENABLES THE BIT PATTERN THAT CONTROLS THE SOURCING OF OUTPUTS
ONTO THE BUS TO BE INSERTED INTO THE MICROPROGRAM WORD.

VARIABLE FIELD SUBSTITUTIONS:
(1) A FOUR BIT BINARY PATTERN THAT SELECTS ONE OF THE DEVICES CAPABLE
OF DRIVING THE BUS.
THE CONSTANTS DEFINED IN THE SECTION CALLED "BUS CONTROL PATTERNS"
SHOULD BE USED AS SUBSTITUTIONS HERE.

OVERLAY RESTRICTIONS:
MAY BE OVERLAPED WITH ANY OTHER FORMAT.

BUS: DEF 6X,4F,50X

MAIN CLOCK CONTROL FORMATS
=====
THESE FORMATS HAVE NO VARIABLE FIELD SUBSTITUTIONS AND CAN BE USED
TO SET THE APPROPRIATE BIT IN THE MICROINSTRUCTION WORD THAT CAUSES
A PARTICULAR PART OF THE SYSTEM TO BE Clocked ON THE CURRENT CYCLE.

OVERLAY RESTRICTIONS:
THERE ARE NO OVERLAY RESTRICTIONS ALL OF THE MAIN CLOCK CONTROL
FORMATS CAN BE USED WITH EACH OTHER AND WITH OTHER FORMATS.

```

```

CKINST: CAUSES THE INSTRUCTION REGISTER TO BE LOADED ON THE CURRENT
CYCLE (IF PIPELINED) OR THE NEXT CYCLE (IF NOW PIPELINED).
CRINST: DEF 30X,301,25X
CVALU: CAUSES THE 2081 CHIPS TO BE Clocked ON THE CURRENT CYCLE.
REGISTERS ARE LOADED ACCORDING TO THE VARIABLE FIELD
SUBSTITUTIONS IN THE ALU FORMAT.
CVALU: DEF 5X,301,50X
CEREG: CAUSES THE EXTERNAL (OR SCRATCH REGISTER FILE TO BE
Clocked.
CEREG: DEF 4X,301,50X
CAUSE: CAUSES THE MICRO INTERRUPT CONTROL UNIT TO BE Clocked (OR
IN OTHER WORDS, THE INSTRUCTION TO THE 2016 IS ENABLED).
CMICINT: DEF 3X,301,60X
CPORT: CAUSES THE LOGIC CONTROLLING THE I/O PORTS OF THE CPU
TO BE Clocked ON THE CURRENT CYCLE SO THAT AN I/O CYCLE
AS DETERMINED BY THE BIT PATTERN SET UP BY THE R/ASH,READ,WRITE
AND R/WT,FORMATS IS CARRIED OUT.
SOME OF THESE CYCLES ONLY GO THROUGH PART OF THEIR
SEQUENCE AFTER ONE CLOCK PORT AND NEED FURTHER CLOCKS TO
COMPLETE. WHEN A CYCLE IS PART WAY THROUGH ITS
SEQUENCE WHEN IT IS Clocked, THE TYPE OF CYCLE DOES
NOT NEED TO BE SPECIFIED.
CPORT: DEF 1X,301,62X
ENAUCE: THIS CAUSES THE AUXILIARY CLOCK FIELD TO BE ENABLED
NORMALLY THE FORMAT AUCE WILL BE USED TO SPECIFY THE
AUXILIARY CLOCK FUNCTION.
ENAUCE: DEF 2X,301,61X

AUCE
=====
THIS FORMAT ENABLES AN AUXILIARY Clocking FIELD TO BE INSERTED INTO THE
MICROPROGRAM WORD.
VARIABLE FIELD SUBSTITUTIONS:
(1) A FOUR BIT BINARY FIELD THAT SPECIFIES THE EXACT Clocking FUNCTION
BIT 0 - SET TO BINARY 1 - CAUSES STATE DISPLAY TO BE Clocked
BIT 1 - CAUSES STATUS DISPLAY TO BE Clocked
BIT 2 - CAUSES STATUS REGISTER TO BE Clocked
BIT 3 - CAUSES MACHINE INTERRUPT UNIT TO

```

(SEE 2004 DATA SHEET)

OVERLAY RESTRICTIONS:
MUST NOT BE OVERLAID WITH MAC OR MACP UNLESS THE CONDITION SELECT
VARIABLE FIELD IS LEFT TO DEFAULT IN MAC OR MACP.

MACSTEN: DEF 401.4120000.201

END

BE Clocked.

OVERLAY RESTRICTIONS:
MUST NOT BE OVERLAID WITH
ADDRESS, STATUS, MACHINT.

MACSTEN: DEF 401.4V

MACINT
000000

THIS FORMAT ENABLES THE MICROINTERRUPT CONTROL FIELD TO BE PLACED IN
THE MICROPROGRAM WORD.

VARIABLE FIELD SUBSTITUTIONS:
(1) A FOUR BIT BINARY FIELD WHICH IS AN INSTRUCTION FOR THE AM 2014
RECORD PRIORITY INTERRUPT CONTROL UNIT.
THIS INSTRUCTION CAN BE PROVIDED USING ONE OF THE CONSTANTS DEFINED
IN THE SECTION CALLED "AM 2014 INSTRUCTION SET".

OVERLAY RESTRICTIONS:
MUST NOT BE OVERLAID WITH REG UNLESS VARIABLE FIELD SUBSTITUTIONS
(3) AND (4) ARE LEFT TO DEFAULT IN REG (IE DONT CARE).

MACINT: DEF 101.4V.421

MACINT
000000

THIS FORMAT IS THE SAME AS MACINT EXCEPT THAT IT APPLIES TO THE 2014
CONTROLLING THE MACHINE INTERRUPT STRUCTURE.

MACINT: DEF 141.4V.401

MACSTEN
000000

THIS FORMAT ENABLES A BIT PATTERN CONTROLLING EXACTLY WHICH OF THE FIRST
FOUR BITS OF THE MACHINE STATUS REGISTER ARE ENABLED WHEN THE MACHINE
STATUS REGISTER, WITHIN THE 2004, IS ITSELF ENABLED.

VARIABLE FIELD SUBSTITUTIONS:
(1) A FOUR BIT BINARY PATTERN AS FOLLOWS:
(1) BIT 0 NOT ENABLE
1 NOT ENABLE C
2 NOT ENABLE B
3 NOT ENABLE P
(THIS IS USED AS ONE IN THE SPEC. FOR
AM 2004 BUT IS USED AS P IN THIS MACHINE)

Appendix D

MICROCODE SOURCE TEXT

1 MAY 1981

AMDOS/29 ANDASH MICRO ASSEMBLER, V1.2
MIL STD 1750 PHASE 1 BIT FORMAT DEFINITION

WORD 64

MICROPROGRAM FIELD FORMAT

BIT FUNCTION

CONDITION FOR ACTION

NO. ALTERNATIVES:

M1 (1) STATUS AND SHIFT INSTRUCTION
CE MAC STAT OR CE MICRO
STAT. ACTIVE (LOW) OR SHIFT
FUNCTION SELECTED BY ALU.
BRANCH ADDRESS REQUIRED BY
AM 29811

M2 (2) 12 BIT BRANCH ADDRESS
DATA FIELD SOURCED ONTO
D BUS

M3 (3) BITS 9 TO 11 OF 16 BIT DATA FIELD
AOI CLK ENABLE ACTIVE.

M4 (4) ADE CLOCK CONTROL FIELD

M5 M6 - CE DISPLAY

M7 M8 - CE STATUS

M9 M10 - CE MACH. INT.

M11

M12 (1) CARRY SELECT FIELD
ALWAYS AFFECTS EXT. CARR LATCH
MACH. CARR. OR MIC. CARR.
IF CORRESPONDING ENABLE BITS
EFFECTIVE IF DATA FIELD IS
ENABLED

M14 (2) BITS 12 THROUGH 16 OF DATA FIELD
ARE ACTIVE.

M15 (1) I/O PORT CONTROL BIT 1
EFFECTIVE IF CE PORT HIGH
(2) BIT 15 OF DATA FIELD IS
SOURCED ONTO D BUS

M16 M17 M18 M19
NEXT ADDRESS CONTROL
ALWAYS APPLICABLE

M20 (1) CONDITION TEST SELECT
TEST COND. REQUIRED BY AM29811

M21 (2) NOT ENABLE Z - M20
MACHINE STATUS ENABLED

M22 NOT ENABLE C - M21

M23 NOT ENABLE M - M22
NOT ENABLE P - M23

1 MAY 1981

AMDOS/29 ANDASH MICRO ASSEMBLER, V1.2
MIL STD 1750 PHASE 1 BIT FORMAT DEFINITION

PAGE 3

M24 CONDITION POLARITY TEST
APPLICABLE IF TEST CONDITION
REQUIRED BY AM 29811

M25 CE INSTRUCTION REG
ALWAYS APPLICABLE

M26 ALU SOURCE CONTROL
ALWAYS APPLICABLE

M27 M28
ALU MS BYTE FUNCTION
ALWAYS APPLICABLE

M29 M30 M31
ALU LS BYTE FUNCTION
ALWAYS APPLICABLE

M32 M33 M34
ALU DESTINATION CONTROL
ALWAYS APPLICABLE

M35 M36 M37
EXTERNAL REGISTER SELECT
APPLICABLE IF EX. REG IS SOURCED

M38 M39 M40 M41
INTERNAL REGISTER A SELECT CONTROL
APPLICABLE IF REG A IS USED
OR IF REG IS Clocked

M42 M43
(1) MICRO INTERRUPT CONTROL BITS 1 AND 2
APPLICABLE IF CE MIC. INT.
IS HIGH.

M44 M45
(1) INTERNAL REGISTER B SELECT CONTROL
APPLICABLE IF REG B IS USED
(2) MICRO INT. CONTROL BITS 3 AND 4
IS HIGH

M46 M47
(1) INTERNAL REGISTER A SELECT MOD.
APPLICABLE IF REG A IS USED
(2) MACH. INT. CONTROL BITS 1 AND 2
IS HIGH

M48 M49
(1) INTERNAL REGISTER B SELECT MOD.
APPLICABLE IF REG B IS USED
(2) MACH. INT. CONTROL BITS 3 AND 4
IS HIGH

M50 2981 D INPUT SOURCE SELECT
APPLICABLE IF D INPUT IS SELECTED

1 MAY 1981

PAGE 4

AM205/20 ANDASH MICRO ASSEMBLER, V1.2
MIL STD 1750 PHASE 1 BIG FORMAT DEFINITION

	AS R OR S SOURCE FOR ALU
PS1 PORT CONTROL BIT 2	APPLICABLE IF CK PORT IS HIGH
PS2 MACH.STAT. NOT ENABLE	ALWAYS APPLICABLE
PS3 MICRO STATUS NOT ENABLE	ALWAYS APPLICABLE
PS4 D BUS SOURCE CONTROL FIELD	ALWAYS APPLICABLE
PS5	
PS6	
PS7	
PS8 CE ALU	ALWAYS APPLICABLE
PS9 CE RD	ALWAYS APPLICABLE
PS10 CE MIC. INTERRUPT	ALWAYS APPLICABLE
PS11 ENABLE AUX CLOCK FIELD	ALWAYS APPLICABLE
PS12 CE PORT	ALWAYS APPLICABLE
PS13 AM208K3 CONTROL	ALWAYS APPLICABLE

TITLE THE AM2080 FAMILY MEMORIC S

13 DECEMBER 1976 JEM
UPDATED SEPT 28, 1977

AM205: INSTRUCTION SET

REGISTER DEFINITIONS

1 MAY 1981

PAGE 5

AM205/20 ANDASH MICRO ASSEMBLER, V1.2
THE AM2080 FAMILY MEMORIC S

R0: EQU R06
R1: EQU R01
R2: EQU R02
R3: EQU R03
R4: EQU R04
R5: EQU R05
R6: EQU R06
R7: EQU R07
R8: EQU R08
R9: EQU R09
R10: EQU R10
R11: EQU R11
R12: EQU R12
R13: EQU R13
R14: EQU R14
R15: EQU R15

AM2061 SOURCE OPERANDS (R S)

AQ: EQU Q06
AB: EQU Q01
ZQ: EQU Q02
ZB: EQU Q03
ZA: EQU Q04
DA: EQU Q05
DQ: EQU Q06
DZ: EQU Q07

AM2061 ALU FUNCTIONS (R FUNCTION S)

ADD: EQU Q06
SUB: EQU Q01
SUBS: EQU Q02
OR: EQU Q03
AND: EQU Q04
NOTES: EQU Q05
XOR: EQU Q06
XNOR: EQU Q07

AM2061 DESTINATION CONTROL

QREG: EQU Q06
WOP: EQU Q01
RAMP: EQU Q02
RAMP: EQU Q03
RAMPD: EQU Q04

TM FS 403

1 MAY 1981

PAGE 8

AMD03/29 AMDASH MICRO ASSEMBLER, V1.2
THE AM2000 FAMILY MEMORICS

```

ZALD: EQU #0001 :SOURCES ALD ONTO D BUS
ZMRG: EQU #0010 :SOURCES MEMORY DATA REGISTER ONTO DBUS
ZMRG: EQU #0011 :SOURCES REGISTER FILE ONTO D BUS
ZMRG: EQU #0012 :SOURCES FAULT REGISTER
ZSTATS: EQU #0010 :SOURCES MACHINE STATUS REGISTER
ZSTATS: EQU #0011 :SOURCES MANUAL DATA INPUT
ZMANG: EQU #0011 :SOURCES MANUAL REGISTER SELECT FIELD
ZDATA: EQU #0100 :SOURCES MICROPROGRAM DATA FIELD
ZSTATE: EQU #0101 :SOURCES CPU STATE DISPLAY
ZTINA: EQU #0101 :SOURCES TIMER A
ZTINB: EQU #0101 :SOURCES TIMER B

```

TEST CONDITION SELECT PATTERNS

```

: THESE CONTROL THE SOURCE OF THE TEST INPUT TO THE Z0011 NEXT ADDRESS
: CONTROL UNIT.

```

```

TZERO: EQU #0000 :PUTS LOGICAL ZERO ON TEST INPUT
ZM B: EQU #0001 :TESTS FOR ZERO MS BYTE
ZLSB: EQU #0010 :TESTS FOR ZERO LS BYTE
OVR: EQU #0011 :TESTS FOR OVERFLOW SET
MUI: EQU #0010 :TESTS FOR OUTPUT OF AM2004 MULTIPLIER
INTPT: EQU #0010 :TESTS MICROINTERUPT REQUEST LINE
CARR: EQU #0010 :TESTS OUTPUT OF INTERNAL CARRY LATCH
COUNT: EQU #0111 :TESTS COUNTER ZERO

```

TEST POLARITY SELECT PATTERNS

```

TRUE: EQU #01
FALSE: EQU #00

```

D INPUT TO Z001 SELECT CONTROL PATTERNS

```

ADUS: EQU #00 :SELECTS THE ADDRESS BUS (OUTPUT OF EXTERNAL REGISTER FILE)
DBUS: EQU #01 :SELECTS THE INTERNAL TRISTATE DATA BUS

```

FORMAT DEFINITIONS

1 MAY 1981

PAGE 9

AMD03/29 AMDASH MICRO ASSEMBLER, V1.2
THE AM2000 FAMILY MEMORICS

ADDRESS

```

*****
THIS FORMAT IS USED TO PLACE A BRANCH ADDRESS IN BITS 0 THRU 11
OF THE MICROINSTRUCTION FOR USE IN DETERMINING THE NEXT ADDRESS IN
A JT, CJT, CJS, JSRP, JRP, RPCT, CJPP OR LDCT INSTRUCTION.
VARIABLE FIELD SUBSTITUTIONS:
(1) A BINARY NUMBER NOT GREATER THAN 12 DIGITS OR A LABEL.
(THE NUMBER WILL BE RIGHT JUSTIFIED AND TRUNCATED)

```

OVERLAY REQUIREMENTS:

```

THIS FORMAT MAY NOT BE OVERLAID WITH
STATSHT, DATINSRT, EMAUICK, AUICK

```

```

ADDRESS: DEF 52X,12Y:

```

STATSHT

```

*****
THIS FORMAT IS USED TO ENABLY THE BINARY PATTERN CONTROLLING THE AM2004
CHIP TO BE INSERTED INTO THE MICROPROGRAM WORD.

```

VARIABLE FIELD SUBSTITUTIONS:

```

(1) A 12 BIT BINARY NUMBER FORMING THE CONTROL
WORD FOR THE 2004 CHIP.

```

```

BIT 11 (MSB) - 112
          10 - 111
           9 - 10
           8 - 18
           0 - ...
           0 - 10

```

```

SEE AM2004 DATA SHEET FOR DEFINITION OF 10 TO 112.

```

OVERLAY RESTRICTIONS:

```

CANNOT BE OVERLAID WITH:
ADDRESS, DATINSRT, EMAUICK, AUICK.

```

```

STATSHT: DEF 52X,12Y

```

1 MAY 1981

PAGE 10

ANDOS/20 ANDASH MICRO ASSEMBLER, V1.2
THE AM2000 FAMILY MEMORICS

CARRYSEL
00000000
THIS FORMAT IS USED TO INSERT INTO THE MICROPROGRAM WORD THE BIT PATTERN
THAT SELECTS THE SOURCE OF THE CARRY BIT THAT IS STORED IN THE
CARRY BITS OF THE MACHINE AND MICROSTATUS REGISTERS IF THEY ARE ENABLED.
IT ALWAYS DETERMINES WHAT IS STORED IN THE CARRY LATCH EXTERNAL
TO THE 2004 ON EVERY CLOCK CYCLE.

VARIABLE FIELD SUBSTITUTIONS:
(1) A THREE BIT BINARY PATTERN APPROPRIATE TO SELECT THE REQUIRED
INPUT TO THE CARRY SELECT MULTIPLIER (IC 3 CPU BOARD 11)
THIS WILL NORMALLY BE SUPPLIED BY USING ONE OF THE CONSTANTS
DEFINED UNDER "CARRY SELECT CONTROL PATTERNS".

OVERLAYING RESTRICTIONS:
MUST NOT BE OVERLAID WITH
DATINSTR

CARRYSEL: DFF 491.3V.12X
DATINSTR: DFF 491.10V:0V

REFRESH
00000000

THIS FORMAT PRODUCES THE APPROPRIATE PATTERN IN BITS 15 AND 31 OF THE
MICROPROGRAM WORD SO THAT A REFRESH CYCLE IS GENERATED THROUGH
THE I/O PORT 17 BIT 62 (REPORT) IS HELD HIGH.

THERE ARE NO VARIABLE FIELDS.

OVERLAY RESTRICTIONS:
MAY NOT BE OVERLAID WITH DATINSTR, READ, WRITE, RMWRT.

REFRESH: DFF 121.000.351.000.15X

READ
00000000

THIS FORMAT IS THE SAME AS REFRESH EXCEPT THAT A READ CYCLE IS INITIATED.

1 MAY 1981

PAGE 11

ANDOS/20 ANDASH MICRO ASSEMBLER, V1.2
THE AM2000 FAMILY MEMORICS

OVERLAY RESTRICTIONS:
MAY NOT BE OVERLAID WITH
DATINSTR, REFRESH, WRITE, RMWRT.
READ: DFF 121.001.351.000.15X

WRITE
00000000

THIS FORMAT IS THE SAME AS REFRESH EXCEPT THAT A WRITE CYCLE IS PRODUCED.

OVERLAY RESTRICTIONS:
MAY NOT BE OVERLAID WITH
REFRESH, READ, RMWRT.

WRITE: DFF 121.000.351.001.15X

RMWRT
00000000

THIS FORMAT IS THE SAME AS REFRESH EXCEPT THAT A READ/MODIFY/WRITE
CYCLE IS PRODUCED.

OVERLAY RESTRICTIONS:
MUST NOT BE OVERLAID WITH
DATINSTR, REFRESH, READ, WRITE.

RMWRT: DFF 121.001.351.001.15X

MAC
000

THIS FORMAT ALLOWS THE NEXT ADDRESS CONTROL SECTION OF THE MICROCODE
WORD TO BE SET UP.

VARIABLE FIELD SUBSTITUTIONS

- (1) A 1 BIT BINARY FIELD. THIS SHOULD BE SET TO A ONE IF THE
20043 BRANCH ADDRESS MODIFICATION FACILITY IS REQUIRED.
THIS FIELD DEFAULTS TO BINARY ZERO.
- (2) A ONE BIT BINARY FIELD WHICH SHOULD BE SET TO A ONE IF
A CONDITIONAL JUMP IS BEING PERFORMED AND THE CONDITION TO

TM FS 403

1 MAY 1981

PAGE 12

AMDOS/29 ANDASH MICRO ASSEMBLER, V1.2
THE AM2900 FAMILY MEMORIC S

BE TESTED IS THE TRUE STATE OF A SELECTED CONDITION INPUT.
IF THE FALSE STATE OF THE CONDITION IS TO BE TESTED THE BIT
SHOULD BE SET TO ZERO. THIS CAN BE ACHIEVED BY USING THE
PREVIOUSLY DEFINED CONSTANTS 'TRUE' AND 'FALSE'.

(3) A FOUR BIT PRIMARY FIELD WHICH SPECIFIES WHICH OF SIXTEEN TEST
CONDITION INPUTS ARE TO BE USED TO DETERMINE THE ACTION
WHEN A CONDITIONAL JUMP IS REQUIRED.

THE CONSTANTS DEFINED IN THE 'TEST CONDITION SELECT PATTERNS'
SECTION CAN BE USED HERE.

(ZERO, ZMSB, ZLSB, OVR, MUI, INTRPT, CARP, COUNT)

(4) A FOUR BIT PRIMARY FIELD THAT SPECIFIES THE INSTRUCTION 'O'

THE AM29011 NEXT ADDRESS CONTROL UNIT.
THE CONSTANTS DEFINED IN THE AM29011 INSTRUCTION SET CAN BE USED
HERE.

OVERLAY RESTRICTIONS:

MUST NOT BE OVERLAIN WITH MACSTEM IF A CONDITION TEST IS SPECIFIED.

MAC: DEF 1F000.371,001.1F1.4F1.4F01110.101

MACP
0000

THIS IS THE SAME AS MAC PCEPPT THAT BIT 25 IS SET TO 0 RATHER THAN 1.
THIS CAUSES THE INSTRUCTION REGISTER TO BE CLOKED EITHER ON THE CURRENT
CYCLE IN THE PIPELINED ARCHITECTURE OR ON THE NEXT CYCLE IN THE NON
PIPELINED HARDWARE.

MACP: DEF 1F000.371,000.1F1.4F1.4F01110.101

ALU
0000

THIS FORMAT ENABLES THE BIT PATTERNS THAT CONTROL THE ALU TO BE INSERTED
INTO THE MICROPROGRAM WORD.

VARIABLE FIELD SUBSTITUTIONS.

(1) A SINGLE BINARY BIT. THIS BIT SHOULD BE SET TO 0 IF THE MICROSTATUS
REGISTER IS TO BE CLOKED AND ONE IF NOT.
THIS FIELD DEFAULTS TO BINARY ONE.

(2) A SINGLE BINARY BIT. THIS BIT SHOULD BE SET TO 0 IF THE MACHINE

1 MAY 1981

PAGE 13

AMDOS/29 ANDASH MICRO ASSEMBLER, V1.2
THE AM2900 FAMILY MEMORIC S

STATUS REGISTER IS TO BE CLOKED AND ONE IF IT IS NOT.
THIS FIELD DEFAULTS TO BINARY ONE.

(3) A SINGLE BINARY BIT. THE CONSTANTS DBUS OR ABUS SHOULD BE
INSERTED HERE TO SPECIFY WHETHER THE DATA BUS
OR THE OUTPUT OF THE INTERNAL REGISTER FILE IS SELECTED AS THE
INPUT TO THE AM2901 DATA INPUT.

(4) A THREE BIT BINARY FIELD SPECIFYING THE DESTINATION OF THE DATA
LEAVING THE 2901 ALU, WITHIN THE 2901 CHIP.
ONE OF THE CONSTANTS DEFINED IN THE SECTION CALLED AM2901 DESTINATION
CONTROL SHOULD BE USED HERE.

(5) A THREE BIT BINARY FIELD SPECIFYING THE ALU LS BYTE FUNCTION.
ONE OF THE CONSTANTS DEFINED IN THE SECTION CALLED AM2901 ALU FUNCTIONS
SHOULD BE USED HERE.

(6) A THREE BIT BINARY FIELD SPECIFYING THE ALU MS BYTE FUNCTION. THE
CHOICE OF SUBSTITUTIONS IS AS IN (5).

(7) A THREE BIT BINARY PATTERN SPECIFYING THE ALU SOURCE CONTROL.
THE SUBSTITUTION SHOULD BE CHOSEN FROM THE CONSTANTS DEFINED IN
THE SECTION CALLED AM2901 SOURCE OPERANDS.

OVERLAY RESTRICTIONS:

CAN BE OVERLAIN WITH ANY OTHER FORMAT.

LU: DEF 101.1F001.1F001.11.1F1.121.3F1.3F1.3F1.201

REG
0000

THIS FORMAT ALLOWS THE BIT PATTERNS THAT CONTROL THE SELECTION OF REGISTERS
TO BE INSERTED INTO THE MICROPROGRAM WORD.

VARIABLE FIELD SUBSTITUTIONS:

(1) A TWO BIT FIELD THE CONTENTS OF WHICH IS ADDED TO THE PIT PATTERN
BEING USED TO SELECT 2901 INTERNAL REGISTER A.

(2) A TWO BIT FIELD THE CONTENTS OF WHICH ARE ADDED TO THE PIT PATTERN
BEING USED TO SELECT THE 2901 INTERNAL REGISTER A.

1 MAY 1981

PAGE 14

AMDOS/29 ANDASH MICRO ASSEMBLER, V1.2
THE AM2900 FAMILY MEMORICS

- (3) A TWO BIT FIELD USED TO CONTROL THE SELECTION OF 2901 INTERNAL REGISTER A. THE SUBSTITUTION MAY BE CHOSEN FROM THE CONSTANTS DEFINED IN THE SECTION CALLED 'INTERNAL REGISTER SELECT CONTROL PATTERNS'.
- (4) A TWO BIT FIELD USED TO CONTROL THE SELECTION OF 2901 INTERNAL REGISTER A. THE SUBSTITUTION MAY BE CHOSEN FROM THE CONSTANTS DEFINED IN THE SECTION CALLED 'INTERNAL REGISTER SELECT CONTROL PATTERNS'.

- (5) A FOUR BIT BINARY PATTERN USED TO SELECT THE SCRATCH (OR EXTERNAL) REGISTER. THE CONSTANTS DEFINED IN THE SECTION CALLED REGISTER DEFINITIONS (R0,R1 ETC.) CAN BE USED HERE.

OVERLAY RESTRICTIONS:

MUST NOT BE OVERLAID WITH
MICINT, MACINT.

REG: DEF 14X,29X,29X,29X,49X,38X

DBUS:

THIS FORMAT ENABLES THE BIT PATTERN THAT CONTROLS THE SOURCING OF OUTPUTS ONTO THE DBUS TO BE INSERTED INTO THE MICROPROGRAM WORD.

VARIABLE FIELD SUBSTITUTIONS:

- (1) A FOUR BIT BINARY PATTERN THAT SELECTS ONE OF THE DEVICES CAPABLE OF DRIVING THE DBUS.
THE CONSTANTS DEFINED IN THE SECTION CALLED 'BUS CONTROL PATTERNS' SHOULD BE USED AS SUBSTITUTIONS HERE.

OVERLAY RESTRICTIONS:

MAY BE OVERLAID WITH ANY OTHER FORMAT.

DBUS: DFF 6X,4X,54X

MAIN CLOCK CONTROL FORMATS

THESE FORMATS HAVE NO VARIABLE FIELD SUBSTITUTIONS AND CAN BE USED TO SET THE APPROPRIATE BIT IN THE MICROINSTRUCTION WORD THAT CAUSES

1 MAY 1981

PAGE 15

AMDOS/29 ANDASH MICRO ASSEMBLER, V1.2
THE AM2900 FAMILY MEMORICS

A PARTICULAR PART OF THE SYSTEM TO BE Clocked ON THE CURRENT CYCLE.

OVERLAY RESTRICTIONS:

THERE ARE NO OVERLAY RESTRICTIONS ALL OF THE MAIN CLOCK CONTROL FORMATS CAN BE USED WITH EACH OTHER AND WITH OTHER FORMATS.

CKINST: CAUSES THE INSTRUCTION REGISTER TO BE LOADED ON THE CURRENT CYCLE (IF PIPELINED) OR THE NEXT CYCLE (IF NON PIPELINED).

CKINST: DEF 38X,34X,25X

CKALU: CAUSES THE 2901 CRIPS TO BE Clocked ON THE CURRENT CYCLE. REGISTERS ARE LOADED ACCORDING TO THE VARIABLE FIELD SUBSTITUTIONS IN THE ALU FORMAT.

CKALU: DEF 5X,34X,59X

CKREG: CAUSES THE EXTERNAL (OR SCRATCH REGISTER FILE TO BE Clocked.

CKREG: DFF 4X,34X,59X

CKINTC: CAUSES THE MICRO INTERRUPT CONTROL UNIT TO BE Clocked (OR IN OTHER WORDS, THE INSTRUCTION TO THE 2914 IS ENABLED).

CKMICINT: DEF 3X,34X,69X

CKPORT: CAUSES THE LOGIC CONTROLLING THE I/O PORTS OF THE CPU TO BE Clocked ON THE CURRENT CYCLE SO THAT AN I/O CYCLE AS DETERMINED BY THE BIT PATTERN SET UP BY THE REFSH,READ,WRITE AND RWMT FORMATS IS CARRIED OUT.

SOME OF THESE CYCLES ONLY GO THROUGH PART OF THEIR SEQUENCE AFTER ONE CLOCK PORT AND NEED FURTHER CLOCKS TO COMPLETE. WITHIN A CYCLE IS PART WAY THROUGH ITS SEQUENCE WHEN IT IS Clocked. THE TYPE OF CYCLE DOES NOT NEED TO BE SPECIFIED.

CKPORT: DEF 1X,34X,62X

ENAUICK: THIS CAUSES THE AUXILIARY CLOCK FIELD TO BE ENABLED NORMALLY THE FORMAT AUICK WILL BE USED TO SPECIFY THE AUXILIARY CLOCK FUNCTION.

ENAUICK: DEF 2X,34X,61X

IM FS 403

1 MAY 1981

PAGE 16

AMDOS/29 AMDASH MICRO ASSEMBLER, V1.2
THE AM2900 FAMILY MNEMONICS

AUICK

THIS FORMAT ENABLES AN AUXILIARY CLOCKING FIELD TO BE INSERTED INTO THE MICROPROGRAM WORD.

VARIABLE FIELD SUBSTITUTIONS:

- (1) A FOUR BIT BINARY FIELD THAT SPECIFIES THE EXACT CLOCKING FUNCTION
 BIT 0 - SET TO BINARY 1 - CAUSES DISPLAY TO BE Clocked
 BIT 1 - CAUSES STATUS DISPLAY TO BE Clocked
 BIT 2 - CAUSES STATUS REGISTER TO BE Clocked
 BIT 3 - CAUSES MACHINE INTERRUPT UNIT TO BE Clocked.

OVERLAY RESTRICTIONS:

MUST NOT BE OVERLAID WITH
 ADDRESS, STATUSPT, DATINSRT.

AUICK: DEF 601.47

MICINT

THIS FORMAT ENABLES THE MICROINTERRUPT CONTROL FIELD TO BE PLACED IN THE MICROPROGRAM WORD.

VARIABLE FIELD SUBSTITUTIONS:

- (1) A FOUR BIT BINARY FIELD WHICH IS AN INSTRUCTION FOR THE AM 2914
 VECTORED PRIORITY INTERRUPT CONTROL UNIT.
 THIS INSTRUCTION CAN BE PROVIDED USING ONE OF THE CONSTANTS DEFINED
 IN THE SECTION CALLED 'AM 2914 INSTRUCTION SET'.

OVERLAY RESTRICTIONS:

MUST NOT BE OVERLAID WITH REG UNLESS VARIABLE FIELD SUBSTITUTIONS
 (3) AND (4) ARE LEFT TO DEFAULT IN REG (IE DONT CARE).

MICINT: DEF 101.47.42E

MICINT

1 MAY 1981

PAGE 17

AMDOS/29 AMDASH MICRO ASSEMBLER, V1.2
THE AM2900 FAMILY MNEMONICS

THIS FORMAT IS THE SAME AS MICINT EXCEPT THAT IT APPLIES TO THE 2914
 CONTROLLING THE MACHINE INTERRUPT STRUCTURE.

MICINT: DEF 141.47.40E

MAGSTEN

THIS FORMAT ENABLES A BIT PATTERN CONTROLLING EXACTLY WHICH OF THE FIRST
 FOUR BITS OF THE MACHINE STATUS REGISTER ARE ENABLED WHEN THE MACHINE
 STATUS REGISTER, WITHIN THE 2904, IS ITSELF ENABLED.

VARIABLE FIELD SUBSTITUTIONS:

- (1) A FOUR BIT BINARY PATTERN AS FOLLOWS:

- BIT 0 NOT ENABLE
 1 NOT ENABLE
 2 NOT ENABLE
 3 NOT ENABLE

(THIS IS USED AS OVR IN THE SPEC. FOR
 AM 2904 PUT IS USED AS P IN THIS MACHINE)

(SEE 2904 DATA SHEET)

OVERLAY RESTRICTIONS:

MUST NOT BE OVERLAID WITH MAC OR MACP UNLESS THE CONDITION SELECT
 VARIABLE FIELD IS LEFT TO DEFAULT IN MAC OR MACP.

MAGSTEN: DEF 401.47P#0000.47E

END

TOTAL PHASE 1 ERRORS - 0

1 MAY 1981

PAGE 1

ANDOS/20 AMDASH MICRO ASSEMBLER, VI.0
 EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EMSL)
 TITLE EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EMSL)
 RELEVANT VERSION OF STANDARD -- MARCH 1, 1980 -- MIL-STD-1750A
 TARGET HARDWARE -- MIL-STD-1750 RIG MELA
 FIRMWARE VERSION 1A.0
 DATE OF RELEASE -- JUNE 1, 1980 S.J.SHRIMPION RAE(P)/FS5

1 MAY 1981

PAGE 2

ANDOS/20 AMDASH MICRO ASSEMBLER, VI.0
 EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EMSL)

P: EQU H#FFFF

SECOND E MICROCODE ADDRESSES

FFFF PLGRIGR2: EQU H#50E
 950E REGSELBIT: EQU H#5E9
 95E1 SELBIT: EQU H#5E1
 947A SEFTMD16: EQU H#47A
 9504 SEFTMD32: EQU H#504
 9446 SEFTMN: EQU H#446
 954B TOPCD400: EQU H#54B
 957E TOPCD402: EQU H#57E
 9497 TOPCD404: EQU H#497
 9418 TOPCD405: EQU H#418
 945A TOPCD40C: EQU H#45A
 9493 TOPCD40Z: EQU H#493
 948A TOPCD40F: EQU H#48A
 940F TOPCD411: EQU H#40F
 9420 TOPCD4A2: EQU H#420
 9429 TOPCD4A4: EQU H#429
 943C TOPCD4A6: EQU H#43C
 9497 TOPCD4A7: EQU H#497
 948E TOPCD4A9: EQU H#48E
 949C TOPCD4A0: EQU H#49C
 9462 TOPCD4AA: EQU H#462
 TOPCD4AB: EQU H#4AB

THIS FIRMWARE CONTROLS ALL THE BASIC MACHINE FUNCTIONS EXCLUDING THE
 ACTUAL CARRYING OUT OF INSTRUCTIONS.

THE FUNCTIONS LOAD INTERNAL REGISTER, DISPLAY INTERNAL REGISTER AND
 OBTAIN TEST INSTRUCTION ARE NOT INCLUDED.

9443 ORG H#0
 9999 RESET SEQUENCE START
 9999 SRESET: MAC ...JP

TM FS 403

1 MAY 1981

PAGE 3

AMOS/20 ANDAM MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1790. (EWSL)

```

/ 6 ADDRESS RESET
/ 6 ALU
: INTERRUPT A SEQUENCE START
0001 SINTPTA: MAC ...JP
/ 6 ADDRESS INTRPTA
/ 6 ALU
: INTERRUPT B SEQUENCE START
0002 SINTPTB: MAC ...JP
/ 6 ADDRESS INTRPTB
/ 6 ALU
: SPARE SEQUENCE START
0003 SPADULTS: MAC ...JP
/ 6 ADDRESS PAULTSV
/ 6 ALU
: SPARE SEQUENCE START
0004 SPARE4: MAC ...JP
/ 6 ADDRESS HALTEST
/ 6 ALU
: FRONT PANEL SERVICE SEQUENCE START
0005 SPANEL: MAC ...JP
/ 6 ADDRESS PANEL
/ 6 ALU
: SPARE SEQUENCE START
0006 SPARES: MAC ...JP
/ 6 ADDRESS HALTEST
/ 6 ALU
: REFRESH SEQUENCE START (DUMMY)
0007 SPREFRESH: MAC ...JP
/ 6 ADDRESS HALTEST
/ 6 ALU
: FRONT PANEL SERVICE SEQUENCE STARTS
-----
ONG 0010

```

1 MAY 1981

PAGE 4

AMOS/20 ANDAM MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1790. (EWSL)

```

0010 RESET: MAC ...JP
0010 / 6 ADDRESS SRESET
/ 6 ALU
: RUN
: ---
0011 SRUM: MAC ...JP & BADDRESS RUN & ALU
: HALT
: ---
0012 SHALT: MAC ...JP & BADDRESS HALT & ALU
: SINGLE STEP
0013 SSINGSTP: MAC ...JP & BADDRESS SINGSTP & ALU
: LOAD PROGRAM COUNTER
0014 SLOADPC: MAC ...JP & BADDRESS LOADPC & ALU
: LOAD MEMORY
0015 SLOADMEM: MAC ...JP & BADDRESS LOADMEM & ALU
: SPARE FUNCTION START
0016 SSPAREP1: MAC ...JP
/ 6 ADDRESS HALTEST
: SPARE FUNCTION START
0017 SSPAREP2: MAC ...JP
/ 6 ADDRESS HALTEST
: SPARE FUNCTION START
0018 SSPAREP3: MAC ...JP
/ 6 ADDRESS HALTEST
: SPARE FUNCTION START

```

1 MAY 1981

PAGE 5

ANDOS/20 ANDASH MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EWSL)

0010 SPAREP4: MAC ...JP
/ 6
ALU ...CONT
: SPARE FUNCTION START

0011 SPAREP5: MAC ...JP
/ 6
ALU ...CONT
: SPARE FUNCTION START

0012 SPAREP6: MAC ...JP
/ 6
ALU ...CONT
: SPARE FUNCTION START

001C LOADREG: MAC ...JP & ADDRESS LOADREG & ALU
/ 6
: LOAD REGISTER

001D DISPPC: MAC ...JP & ADDRESS DISPPC & ALU
/ 6
: DISPLAY PROGRAM COUNTER

001E DISPMEM: MAC ...JP & ADDRESS DISPMEM & ALU
/ 6
: DISPLAY MEMORY

001F DISPPG: MAC ...JP & ADDRESS DISPPG & ALU
/ 6
: DISPLAY REGISTER

FRONT PANEL SERVICE FUNCTIONS

0020

RUN FUNCTION

1 MAY 1981

PAGE 6

ANDOS/20 ANDASH MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EWSL)

0020: (1) THE STATUS WORD IS LOADED INTO THE REGISTER FROM EXTERNAL REGISTER
NUMBER 4
0020 RUN: MAC ...CONT
/ 6
ALU ...ABUS,OREG,OR,OP,DZ
/ 6
RPG ...R4
/ 6
CRALU

0021: (2) BIT 15 OF THE STATUS WORD IS SET TO ZERO TO INDICATE MACHINE HALT
/ 6
MAC ...CONT
/ 6
ALU ...ABUS,OREG,OR,OP,DZ
/ 6
RPG ...R4
/ 6
CRALU

0022: (3) THE NEW STATUS WORD IS LOADED INTO THE MACHINE STATE DISPLAY LATCH
AND EXTERNAL REGISTER NUMBER 4.
/ 6
MAC ...CONT
/ 6
ALU ...OREG,OR,OP,DZ
/ 6
RPG ...R4
/ 6
CRALU

0023: (4) RETURN TO HALTEST SEQUENCE.
/ 6
MAC ...JP
/ 6
ADDRESS HALTEST
/ 6
ALU

HALT FUNCTION

0024: (1) THE STATUS WORD IS LOADED INTO THE Q REGISTER FROM
EXTERNAL REGISTER #4

0024 HALT: MAC ...CONT
/ 6
ALU ...ABUS,OREG,OR,OP,DZ
/ 6
RPG ...R4
/ 6
CRALU

0025: (2) BIT 15 OF THE STATUS WORD IS SET TO ZERO TO INDICATE MACHINE HALT.
/ 6
MAC ...CONT
/ 6
ALU ...ABUS,OREG,AND,AND,DQ

TM FS 403

1 MAY 1981

PAGE 7

AMDOS/29 AMDASH MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EWSL)

```

/ 6 DBUS SDATA
/ 6 DATINSTR D032767 ; MASK 0111111111111111
/ 6 CEALU

```

```

; (3) THE NEW MACHINE STATUS WORD IS LOADED INTO THE MACHINE STATE DISPLAY LATCH
; AND EXTERNAL REGISTER #4

```

```

0020 / 6 MAC ...CONT
/ 6 ALU ...QREG. OR. OR. ZQ
/ 6 REG ...R2
/ 6 DBUS SALU
/ 6 ENAUIKE
/ 6 AUICK B00010
/ 6 CREG

```

```

; (4) RETURN TO HALTEST SEQUENCE

```

```

0027 / 6 MAC ...JP
/ 6 ADDRESS HALTEST
/ 6 ALU

```

```

; LOAD REGISTER

```

```

; (1) REGISTER SELECT TAKES PLACE
0028 / 6 LOADREG: MAC ,PULSE.TZPRO.CJS
/ 6 ALU
/ 6 ADDRESS MANREGST

```

```

; (2) THE CONTENTS OF THE SWITCH FIELD ARE LOADED INTO THE SELECTED REGISTER.

```

```

0029 / 6 MAC ...JP
/ 6 ALU ...DBUS.RAMU. OR. OR. DZ
/ 6 REG ...R2
/ 6 DBUS SHANDT
/ 6 ADDRESS HALTEST
/ 6 CEALU

```

```

; DISPLAY REGISTER

```

```

; (1) REGISTER SELECT TAKES PLACE
002A / 6 DISPREG: MAC ,PULSE.TZPRO.CJS
/ 6 ALU
/ 6 ADDRESS MANREGST

```

1 MAY 1981

PAGE 8

AMDOS/29 AMDASH MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EWSL)

```

; (2) THE CONTENTS OF THE SELECTED REGISTER ARE PLACED IN THE DISPLAY LATCH.

```

```

002B / 6 MAC ...CONT
/ 6 ALU ...MOP. OR. OR. ZA
/ 6 REG ...R2
/ 6 DBUS SALU
/ 6 ENAUIKE
/ 6 AUICK B0001

```

```

; (3) JUMP TO HALTEST TAKES PLACE

```

```

002C / 6 MAC ...JP
/ 6 ADDRESS HALTEST
/ 6 ALU

```

```

; REGISTER SELECT ROUTINE

```

```

; (1) THE CONTENTS OF INTERNAL REGISTER ZERO ARE TEMPORARILY STORED IN THE
; Q REGISTER.

```

```

002D / 6 MANREGST: MAC ...CONT
/ 6 ALU ...QREG. OR. OR. ZA
/ 6 REG ...R2
/ 6 CEALU

```

```

; (2) THE REGISTER SELECT FIELD IS LOADED INTO IR0 AND CIRCULARLY UPSHIFTED.

```

```

002E / 6 MAC ...CONT
/ 6 ALU ...DBUS.RAMU. OR. OR. DZ
/ 6 STATSHFT B0001000000000000
/ 6 REG ...R2
/ 6 DBUS SHANDT
/ 6 CEALU

```

```

; (3) THE CONTENTS OF IR0 ARE UPSHIFTED

```

```

002F / 6 MAC ...CONT
/ 6 ALU ...RAMU. OR. OR. ZB
/ 6 REG ...R2
/ 6 STATSHFT B0001000000000000
/ 6 DBUS SALU
/ 6 CEALU

```

```

; (4) THE CONTENTS OF IR0 ARE UPSHIFTED

```

```

0030 / 6 MAC ...CONT
/ 6 ALU ...RAMU. OR. OR. ZB
/ 6 REG ...R2
/ 6 STATSHFT B0001000000000000

```


TM FS 403

1 MAY 1981

PAGE 11

ANDOS/28 ANDASH MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EMSL)

```

/ 5 DBUS SMANDT
/ 6 CEPORT & CEALU
/ 6 ENAUCK & STATSHPT B#010000000001

```

```

(2) THE INCREMENTED PC CONTENTS ARE PLACED IN ZR0 AND JUMP TAKES PLACE
TO HALTEST.

```

```

/ 6 MAC ...JP & ALU ...HOP,OR,OR,ZQ & CEREQ & REG ....R0
/ 6 DBUS SALU & ADDRESS HALTEST

```

```

DISPLAY STATUS
*****

```

```

0039 DISPTAT: MAC ...CONT & ALU & DBUS SSTATUS & ENAUCK & STATSHPT B#000000100001

```

```

003C MAC ...JP & ALU & ADDRESS HALTEST

```

```

SINGLE STEP
-----

```

```

A SINGLE MACHINE INSTRUCTION IS EXECUTED AND THE PROGRAM COUNTER IS
INCREMENTED.

```

```

(1) THE INSTRUCTION FROM MEMORY IS FETCHED

```

```

003D SINGSTP: MAC ...CONT
/ 5 ALU ...R0
/ 5 REG ....R0
/ 6 CEPORT
/ 6 READ

```

```

(2) THE INSTRUCTION IS LOADED INTO THE INSTRUCTION REGISTER AND EN1.
JUMP SUB TAKES PLACE TO EXECUTE.

```

```

003E MACP .FALSE,ZZERO,CJS
/ 5 ADDRESS EXECUTE
/ 5 ALU
/ 5 DBUS SMEM
/ 5 REG ....R1
/ 6 CEREQ

```

```

(4) JUMP TO HALTEST TAKES PLACE.

```

```

003F MAC ...JP
/ 6 ADDRESS HALTEST

```

1 MAY 1981

PAGE 12

ANDOS/28 ANDASH MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EMSL)

```

/ 5 ALU

```

```

FRONT PANEL SERVICE SEQUENCE
-----

```

```

(1) THE MICROINTERUPT THAT CAUSED ENTRY INTO THIS AREA IS CLEARED.
0040 PANEL: MAC ...CONT
/ 5 MICINT CLVYC
/ 5 MICINT CLVYC
/ 5 CERICINT

```

```

(2) THE STATUS REGISTER IN THE MICROINTERUPT UNIT IS LOADED WITH ZERO.

```

```

0041 MAC ...CONT
/ 5 ALU
/ 5 DBUS SDATA
/ 5 DATINSTR B#0
/ 5 MICINT SDATA
/ 5 CERICINT

```

```

(3) A VECTOR JUMP IS MADE TO THE APPROPRIATE FRONT PANEL SERVICE SEQUENCE.
MAC .FALSE,ZZERO,CJV & ALU

```

```

HALTEST SEQUENCE
-----

```

```

(3) A CHECK IS STARTED TO SEE IF HALT HAS BEEN SET.
THE MACHINE STATEWORD IS PASSED THROUGH THE ALU AND SHIPPED UP SO THAT THE
MOST SIGNIFICANT BIT GOES INTO THE CARRY. THE ALU IS NOT CLOCEED SO THAT NO
REGISTERS ARE LOADED.

```

```

0043 HALTEST: MAC ...CONT
/ 5 ALU ...RUS,RAMU,OR,OR,DZ
/ 5 STATSHPT B#000100000000
/ 5 CERICINT CERICR
/ 5 DBUS SALU
/ 6 REG ....R4

```

```

(4) A BRANCH OCCURS TO THE FETCH SEQUENCE IF THE CARRY LATCH HAS BEEN SET
IMPLYING THAT THERE WAS A ONE IN THE MSB OF THE MACHINE STATE WORD.
MAC .TRUE,CARRY,CJP
/ 6 ADDRESS FETCH

```


IN 33 403

1 MAY 1981

PAGE 15

ANDOS/29 ANDASM MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EMSL)

```

(11) THE MASK REGISTER OF BOTH INTERRUPT UNITS IS LOADED SO THAT ALL PUT
: THE POWER DOWN INTERRUPT AND THE MACHINE ERROR INTERRUPT ARE DISABLED.
MAC ...CONT & ALU ...NOP,OR,OR,2Q & DBUS SALU & MACINT LDM
/ 5
STATSHFT B#00000001000 & ENAUCX
(12) THE 2 REGISTER IS LOADED WITH 1000100000000000
MAC ...CONT & ALU ...DBUS,OREG,OR,OR,DZ & CEALU & DBUS SHATA
/ 5
DATINSHT D#035040

```

```

(13) THE STATE WORD IS LOADED SO THAT:
: TRP MACHINE IS RUNNING (BIT 0 SET TO 1)
: TIMER A IS RUNNING (BIT 4 SET TO 1)
: TIMER B IS RUNNING (BIT 5 SET TO ONE)
: THE STARTUP ROM IS ENABLED (BIT 6 SET TO ONE)
: MEMORY PROTECT RAM IS DISABLED (BIT 7 SET TO 0)
: DMA IS DISABLED (BIT 8 SET TO 0)
: INTERRUPTS ARE DISABLED (BIT 9 SET TO ZERO)
: ALL OTHER BITS ARE SET TO ZERO
MAC ...CONT & ALU ...NOP,OR,OR,2Q & DBUS SALU & RES ...RA & CERIC
/ 5
ENAUCX & STATSHFT B#0000000010

```

```

(14) JUMP TAKES PLACE TO A DEFINED LOCATION IN THE SECOND K OF ROM SO THAT
: ADDITIONAL RESET PROCEDURES CAN BE ADDED.
MAC ...JP & ALU & BADADDRESS HALTEST
0055 / 5
0056

```

EXECUTE SEQUENCE

```

(1) THE PROGRAM COUNTER IS INCREMENTED, JUMP MAP TAKES PLACE TO THE START
: OF THE APPROPRIATE EMULATION SEQUENCE.
EXECUTE: MAC ...JMAP
/ 5
STATSHFT B#01000000000

```

FAULT SERVICE SEQUENCE

```

THIS SEQUENCE RUNS AS A RESULT OF A HARDWARE FAULT FLAG BEING SET IN THE
: FAULT FLAG REGISTER.
: THE SEQUENCE SETS THE APPROPRIATE BIT IN THE FAULT REGISTER ERS AND CAUSES
: A MACHINE ERROR INTERRUPT -- NUMBER 1.

```

1 MAY 1981

PAGE 16

ANDOS/29 ANDASM MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EMSL)

```

0058 FAULTSV: MAC ...CONT & ALU ...DBUS,OREG,OR,OR,DZ & CEALU & DBUS STAULT
: (2) THE FAULT FLAG REGISTER IS CLEARED.
: THE CONTENTS OF THE FAULT REGISTER ARE ORED WITH THE CONTENTS OF THE
: REGISTER AND THE RESULTS PLACED IN THE FAULT REGISTER ERS.
MAC ...CONT & ALU ...ARUS,NOP,OR,OR,RA & RES ...RA & CERIC
/ 5
DBUS SALU & ENAUCX & STATSHFT B#00000010000

```

```

(3) THE MICRO INTERRUPT IS CLEARED.
MAC ...CONT & ALU & CMICINT & MICINT CLERC
005A / 5

```

```

(4) THE BINARY PATTERN 0100000000000000 IS LOADED INTO THE OREG.
005B FAULTSV: MAC ...CONT & ALU ...CEUS,OREG,OR,OR,DZ & DBUS SHATA & CEALU
/ 5
DATINSHT D#16394

```

```

(5) THE CONTENTS OF THE REGISTER ARE ORED WITH AND LOADED INTO THE
: PENDING INTERRUPT REGISTER.
MAC ...CONT & ALU ...NOP,OR,OR,2Q & DBUS SALU & ENAUCX
/ 5
STATSHFT B#01000000000

```

```

(6) JUMP TO HALTEST TAKES PLACE.
MAC ...JP & ALU & BADADDRESS HALTEST
005D / 5

```

INTERUPT 'A' SEQUENCE

```

THIS IS THE FIRST PART OF THE MACHINE INTERRUPT SERVICE SEQUENCE FOR AN
: INTERRUPT DETECTED BY THE 'A' GROUP OF INTERRUPTS. IE. THE LEAST
: SIGNIFICANT GROUP.

```

```

(1) THE VECTOR FROM THE LEAST SIGNIFICANT 2014 IS READ AND STORED IN ERS.
INTERPTA: MAC ...CONT & ALU & ENAUCX & CERIC & RES ...RB
/ 5
STATSHFT B#00000001000 & MACINT RDVC

```

```

(1A) THE CONTENTS OF ERS ARE STOPPED WITH THE CONTENTS OF IRB.
MAC ...CONT & ALU ...ABUS,RAHL,OR,OR,DZ & RES ...RZ & RZ
/ 5
DBUS SALU & CERIC & CEALU

```

```

(1B) THE CONTENTS OF IRB ARE MASKED BY (0000000000000011).
MAC ...CONT & ALU ...DBUS,RAHL,AND,AND,DA & RES ...RZ & RZ
005E / 5

```

1 MAY 1981

PAGE 17

ANDOS/20 ANDASH MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (ENSL

```

/ 6 DBUS SDATA & DATINSTR D07 & CEALU
;(1C) THE CONTENTS OF IR0 ARE SHIPPED ONE PLACE UPWARDS.
MAC ...CONT & ALU ...RAMU,OR,OR,2A & REG ...RZERO,RZERO & CEALU
STATSHFT 0000000000000000
/ 6
;(2) THE CONTENTS OF REG AND IR0 ARE SHIPPED BACK.
MAC ...CONT & ALU ...RZERO,RZERO,OR,OR,2A & REG ...RZERO,RZERO,R0
DBUS SALU & CEALU & CREG
/ 6
;(2A) THE MACHINE INTERRUPT IS CLEARED.
MAC ...CONT & ALU ...ENAUICE & MACINT CLRVC
STATSHFT 0000000000000000
/ 6
;(3) THE MACHINE INTERRUPT STATUS IS ZEROED.
MAC ...CONT & ALU ...RZERO,RZERO,AND,DZ & DBUS SALU & ENAUICE
MACINT LDSTA & STATSHFT 0000000000000000
/ 6
;(4) THE MICRO-INTERRUPT IS CLEARED. JUMP TO THE SECOND PART OF THE
INTERUPT SEQUENCE TAKES PLACE.
MAC ...JP & ALU & CEMICINT & MICINT CLRVC
/ 6
BADADDRESS INTRPT2
/ 6

```

THIS IS THE FIRST PART OF THE MACHINE INTERRUPT SERVICE SEQUENCE FOR AN
INTERUPT DETECTED BY THE 'P' GROUP OF INTERRUPTS. IE. THE MOST SIGNIFICANT
GROUP.

```

/ 6
;(1) THE VECTOR IS READ FROM THE 'B' 2014.
MAC ...CONT & ALU & ENAUICE & CREG
STATSHFT 0000000000000000 & REG ...R0 & MACINT RDVC
/ 6
;(2) THE QREG IS LOADED WITH DB0.
MAC ...CONT & ALU ...DBUS,QREG,OR,OR,DZ & DBUS SDATA & CEALU
DATINSTR D08
/ 6
;(3) THE CONTENTS OF REG ARE SHIPPED WITH THE CONTENTS OF IR0.
MAC ...CONT & ALU ...RZERO,RZERO,OR,OR,DZ & REG ...RZERO,RZERO,R0
DBUS SALU & CREG & CEALU
/ 6
;(3A) THE CONTENTS OF IR0 ARE MASKED BY (0000000000000001).
MAC ...CONT & ALU ...DBUS,NAMT,APD,AND,DA & REG ...RZERO,RZERO
DBUS SDATA & DATINSTR D07 & CEALU
/ 6
;(4) THE CONTENTS OF IR0 ARE ORED WITH DB0 AND SHIPPED UP ONE PLACE.

```

1 MAY 1981

PAGE 18

ANDOS/20 ANDASH MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (ENSL

```

/ 6
MAC ...CONT & ALU ...RAMU,OR,OR,1Q & REG ...RZERO,RZERO & CEALU
STATSHFT 0000000000000000
/ 6
;(5) THE CONTENTS OF IRP AND ER0 ARE SHIPPED BACK.
MAC ...CONT & ALU ...RZERO,RZERO,OR,OR,2A & REG ...RZERO,RZERO,R0
CEALU & CREG & DBUS SALU
/ 6
;(5A) THE MACHINE INTERRUPT IS CLEARED.
MAC ...CONT & ALU & MACINT CLRVC & ENAUICE
STATSHFT 0000000000000000
/ 6
;(6) THE MACHINE INTERRUPT STATUS IS ZEROED.
MAC ...CONT & ALU ...RZERO,RZERO,AND,DZ & DBUS SALU & MACINT LDSTA
STATSHFT 0000000000000000 & ENAUICE
/ 6
;(7) THE MICRO-INTERRUPT IS CLEARED. JUMP TO THE SECOND PART OF THE INTERRUPT
SERVICE SEQUENCE TAKES PLACE.
MAC ...JP & ALU & MICINT CLRVC
/ 6
CEMICINT
/ 6
BADADDRESS INTRPT2
/ 6

```

;(1) THE MICRO-INTERRUPT STATUS IS LOADED WITH ZERO.

```

/ 6
STATSHFT 0000000000000000 & DATINSTR D08 & CEMICINT
/ 6
MICINT LDSTA

```

```

/ 6
;(2) THE ADDRESS OF WHERE TO STORE THE MACHINE STATUS IS FETCHED FROM MEMORY.
MAC ...CONT & ALU ...RZERO,RZERO,AND,DZ & REG ...R0 & CREG
CEALU & CREG & STATSHFT 0000000000000000 & DBUS SALU
/ 6

```

```

/ 6
;(2A) THE ADDRESS IS PLACED IN ER2.
MAC ...CONT & ALU & DBUS SHRM & REG ...R2 & CREG
/ 6

```

```

/ 6
;(2A) THE MACHINE INTERRUPT MASK FROM ER15 IS INVERTED AND LOADED INTO THE QREG.
MAC ...CONT & ALU ...RZERO,RZERO,AND,DZ & REG ...R0 & CREG
CEALU & CREG & STATSHFT 0000000000000000 & DBUS SALU
/ 6

```

```

/ 6
;(3) THE MACHINE INTERRUPT MASK WORD IS STORED.
MAC ...CONT & ALU ...RZERO,RZERO,AND,DZ & REG ...R2 & CREG & WHITE
DBUS SALU
/ 6

```

```

/ 6
;(4) THE CONTENTS OF ER2 ARE INCREMENTED.
MAC ...CONT & ALU ...RZERO,RZERO,AND,DZ & DBUS SALU & REG ...R2
CEALU & STATSHFT 0000000000000000
/ 6

```


22 1072

[illegible][illegible][illegible]

```
(1) MEMORY ADDRESS IN REG. #17 TAKES PLACE. THE PROGRAM COUNTER  
CONTENTS ARE INCREMENTED AND PLACED BACK IN REG.  
NAC ...CONT & ALU ...RANA,OP,RAND,DZ & REG ...RZ & CEREG & CEPOT  
READ & STATSHFT B000000000000 & DEUS SALU  
/ &  
(2) THE CONTENTS OF IR0 ARE PLACED IN ERZ. ZERO IS PLACED IN IR0.  
NAC ...CONT & ALU ...RANA,AND,ZERO & REG ...RZERO,RZERO,RZ  
/ &  
DEUS SALU & CEALU & CERZ  
(3) THE CONTENTS OF CRC ARE ADDED TO THE CONTENTS OF THE MEMORY DATA  
REGISTER AND THE RESULT PLACED IN IR0.  
NAC ...CONT & ALU ...DRU,RRM,ADD,ADD,DA & REG ...RZERO,CRC & CEALU  
/ &  
DEUS SRM & STATSHFT B00000000000  
(4) THE CONTENTS OF ERK ARE RESTORED TO IR0. CONTENTS OF IR0 ARE PLACED IN  
ERZ.  
NAC ...CONT & ALU ...ABUS,RA,CR,CR,DZ & REG ...RZERO,RZERO,RZ  
/ &  
DEUS SALU & CEALU & CERZ  
(5) THE DERIVED OPERAND IS FETCHED FROM MEMORY.
```

243E 21

1861 17M 1

AMDOS/28 AMDASSM MICRO ASSEMBLER, V1.0
 EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EMSL

APPENDIX C ROUTINES

REF ID: A66000

REF: 1. THIS A COUNTRY CODE DIRECT OR DIRECT INDEED OPERAND AND PLACES
IT IN A AND LONG. DIRECT OR DIRECT INDEED IS CHosen ACCORDING
TO THE CONTENTS OF BITS 15 THRU 16 OF THE MACHINE INSTRUCTION.
THE PROGRAM COUNTER IS AUTOMATICALLY INCREMENTED TO TAKE ACCOUNT OF
THE LONG INSTRUCTION PREFIX. THIS SUM OF THE CAN THEREFORE BE
USED AS A MACHINE INSTRUCTION COUNTER ADDRESS.

[illegible]

00001: PETCHES A SINGLE WORD BASE RELATIVE OPERAND AND PLACES IT IN R5.
-THE ADDRESS OF THE DERIVED OPERAND (IE CONTENTS OF SPECIFIED BASE
REGISTER PLUS THE DISPLACEMENT) IS LEFT IN R2.

OPB2: FETCHES A DOUBLE WORD BASE RELATIVE OPERAND AND PLACES IT IN LRS
 AND FR6.

DPB3: PETCHES A TRIPLE WORD BASE RELATIVE OPERAND AND PLACES IT IN ER5.
VR6 AND ER7.

COFFIN: PITCHES A SINGLE WORD BASE RELATIVE INDEXED OPERAND AND PLACES IT

1 MAY 1981

PAGE 23

AMDOS/29 AMDASH MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (BMSL)

0014 MAC ...CONT & ALU & READ & CEPORT & REG R2
0015 (6) THE DERIVED OPERAND IN THE MEMORY DATA REGISTER IS PLACED IN ER5
MAC .FALSE, ZERO, CPTM & ALU & DBUS SHM & REG R5 & CEREQ

OPD12 -- FETCHES TWO WORD DERIVED OPERAND AND PLACES THE RESULT IN ER 5 AND 6.

0016 (1) JUMP SUB TO OPD11 -- THIS FETCHES THE FIRST WORD.
OPD12: MAC .FALSE, ZERO, CJS & ALU & ADDRESS OPD11
0017 (2) THE CONTENTS OF ER2 ARE INCREMENTED.
MAC ...CONT & ALU ...ABUS, NOP, ADD, DZ & REG R2 & DBUS SALU
/ & CEREQ & STATSHFT B0010000000000
0018 (3) MEMORY ACCESS IN READ MODE TAKES PLACE.
MAC ...CONT & ALU & CEPORT & READ & REG R2
0019 (4) THE DATA FROM MEMORY IS PLACED IN ER6. RETURN TAKES PLACE.
MAC .FALSE, ZERO, CPTM & ALU & DBUS SHM & REG R6 & CEREQ

OPD13 FETCHES A 3WORD DIRECT OPERAND FROM MEMORY AND PLACES IT IN ER5, 6 AND 7.

0020 (1) JUMP SUB TO OPD11. THIS OBTAINS THE FIRST WORD OF THE DERIVED OPERAND
AND PLACES IT IN ER5.
OPD13: MAC .FALSE, ZERO, CJS & ALU & ADDRESS OPD11
0021 (2) THE CONTENTS OF ER2 ARE INCREMENTED.
MAC ...CONT & ALU ...ABUS, NOP, ADD, DZ & CEREQ & DBUS SALU
/ & REG R2 & STATSHFT B0010000000000
0022 (3) MEMORY ACCESS IN READ MODE TAKES PLACE. THE CONTENTS OF ER2 ARE
INCREMENTED.
MAC ...CONT & ALU ...ABUS, NOP, ADD, DZ & DBUS SALU & REG R2
/ & CEREQ & CEPORT & READ & STATSHFT B0010000000000
0023 (4) THE CONTENTS OF THE MEMORY DATA REGISTER ARE PLACED IN ER6.
MAC ...CONT & ALU & REG R6 & DBUS SHM & CEREQ
0024 (5) THE MEMORY IS ACCESSED IN READ MODE.
MAC ...CONT & ALU & CEPORT & READ & REG R2

1 MAY 1981

PAGE 24

AMDOS/29 AMDASH MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (BMSL)

0025 (6) THE MEMORY DATA IS PLACED IN ER7. RETURN TAKES PLACE.
MAC .FALSE, ZERO, CPTM & ALU & REG R7 & DBUS SHM & CEREQ

OPB1 -- FETCHES A SINGLE WORD BASE RELATIVE OPERAND.

0026 (1) THE INSTRUCTION IN B01 IS MASKED TO LEAVE ONLY THE LEAST SIGNIFICANT
8 BITS AND PLACED IN THE Q REG.
OPB1: MAC ...CONT & ALU ...ABUS, QREG, OR, AND, DZ & REG R1 & C0ALU
0027 (2) THE CONTENTS OF THE SPECIFIED BASE REGISTER ARE ADDED TO THE CONTENTS
OF THE REGISTER AND THE RESULT PLACED IN ER2.
MAC ...CONT & ALU ...NOP, ADD, ADD, AQ & REG ...BASE, R2 & DBUS SALU
/ & CEREQ & STATSHFT B000000000000000
0028 (3) THE DERIVED OPERAND IS FETCHED FROM MEMORY.
MAC ...CONT & ALU & READ & CEPORT & REG R2
0029 (4) THE DERIVED OPERAND IS PLACED IN ER5. RETURN TAKES PLACE.
MAC .FALSE, ZERO, CPTM & ALU & DBUS SHM & CEREQ & REG R5

OPB11 -- FETCHES BASE RELATIVE INDEXED OPERAND.

0030 (1) THE CONTENTS OF IR0 ARE PLACED IN ER5. ZERO IS PLACED IN IR6.
OPB11: MAC ...CONT & ALU ...NAME, AND, AND, DZ & REG ...ZERO, ZERO, R5 & C0ALU
/ & CEREQ & DBUS SALU
0031 (2) THE CONTENTS OF THE SPECIFIED BASE REGISTER ARE ADDED TO THE CONTENTS
OF THE SPECIFIED INDEX REGISTER AND THE RESULT PLACED IN ER2
MAC ...CONT & ALU ...NOP, ADD, ADD, AB & R5C ...BASE, R2 & CEREQ
/ & DBUS SALU & STATSHFT B000000000000000
0032 (3) THE MEMORY IS ACCESSED TO OBTAIN THE DERIVED OPERAND.
MAC ...CONT & ALU & R2 ...R2 & CEPORT & READ
0033 (4) THE DERIVED OPERAND IS PLACED IN ER5 AND THE CONTENTS OF ER5 RESTORED
TO IR0.
MAC .FALSE, ZERO, CPTM & ALU ...ABUS, RAMP, OR, OR, DZ & REG ...ZERO, R5
/ & DBUS SHM & C0ALU & CEREQ

1 MAY 1981 PAGE 29

AMDAS/29 AMDASM MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EMSL)

OSB11 -- THIS ROUTINE SENDS A BASE RELATIVE INDEXED OPERAND TO ITS DESTINATION

OSB11: (1) THE CONTENTS OF IR0 ARE PLACED IN ER0, ZERO IS PLACED IN IR0.
MAC ...CONT & ALU ...RAMA.AND.ZA & REG ...RZERO.RZERO.R0
CVALU & CREG & DBUS SALU

OSB11: (2) THE CONTENTS OF THE SPECIFIED BASE REGISTER ARE ADDED TO THE CONTENTS
OF THE SPECIFIED INDEX REGISTER AND THE RESULT PLACED IN ER2
MAC ...CONT & ALU ...QREG.ADD.ADD.AB & REG ...BASE.R2.R2 & CREG
DBUS SALU & STATUS R0000000000000000

OSB11: (3) THE OPERAND IN ER5 IS PLACED IN QREG.
MAC ...CONT & ALU ...ABUS.QREG.OR.OR.DZ & REG ...R5 & CVALU

OSB11: (4) THE OPERAND, NOW IN THE QREG IS SENT TO MEMORY, ER2 BEING USED TO SUPPLY
THE ADDRESS.

OSB11: (5) THE CONTENTS OF IR0 ARE RESTORED, RETURN TAKES PLACE.
MAC ...FALSE.TZERO.CRTN & ALU ...ABUS.RAMP.OR.OR.DZ & REG ...RZERO..R0
CVALU

OSB11: (6) THE OPERAND, NOW IN THE QREG IS SENT TO MEMORY USING THE CONTENTS OF
FR5 AS ADDRESS. RETURN TAKES PLACE.
MAC ...FALSE.TZERO.CRTN & ALU ...MOP.OR.OR.ZQ & DBUS SALU
CREPORT & WRITE & REG ...R5

OSB11: (7) JUMP SUB TO OFD11. THIS FETCHES THE ADDRESS OF THE INDIRECT OPERAND TO
ITS DESTINATION.

OSB11: (8) THE CONTENTS OF FR5 ARE PLACED IN QREG.
MAC ...FALSE.TZERO.CJS & ALU ...ABUS.QREG.OR.OR.DZ & REG ...R5
CVALU & BADDRESS OFD11

OSB11: (9) THE OPERAND, NOW IN THE QREG IS SENT TO MEMORY USING THE CONTENTS OF
FR5 AS ADDRESS. RETURN TAKES PLACE.
MAC ...FALSE.TZERO.CRTN & ALU ...MOP.OR.OR.ZQ & DBUS SALU
CREPORT & WRITE & REG ...R5

OSB11: (10) THE CONTENTS OF THE QREG (MULTIPLIER) ARE MASKED BY THE CONTENTS OF
FR5 SO THAT ONLY THE SIGN BIT OF THE MULTIPLIER IS LEFT.

OSB11: (11) THE CONTENTS OF THE QREG (MULTIPLIER) ARE MASKED BY THE CONTENTS OF
FR5 SO THAT ONLY THE SIGN BIT OF THE MULTIPLIER IS LEFT.

1 MAY 1981

PAGE 29

AMDAS/29 AMDASM MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EMSL)

OSB11 -- THIS ROUTINE SENDS A BASE RELATIVE INDEXED OPERAND TO ITS DESTINATION

OSB11: (1) THE CONTENTS OF IR0 ARE PLACED IN ER0, ZERO IS PLACED IN IR0.
MAC ...CONT & ALU ...RAMA.AND.ZA & REG ...RZERO.RZERO.R0
CVALU & CREG & DBUS SALU

OSB11: (2) THE CONTENTS OF THE SPECIFIED BASE REGISTER ARE ADDED TO THE CONTENTS
OF THE SPECIFIED INDEX REGISTER AND THE RESULT PLACED IN ER2
MAC ...CONT & ALU ...QREG.ADD.ADD.AB & REG ...BASE.R2.R2 & CREG
DBUS SALU & STATUS R0000000000000000

OSB11: (3) THE OPERAND IN ER5 IS PLACED IN QREG.
MAC ...CONT & ALU ...ABUS.QREG.OR.OR.DZ & REG ...R5 & CVALU

OSB11: (4) THE OPERAND, NOW IN THE QREG IS SENT TO MEMORY, ER2 BEING USED TO SUPPLY
THE ADDRESS.

OSB11: (5) THE CONTENTS OF IR0 ARE RESTORED, RETURN TAKES PLACE.
MAC ...FALSE.TZERO.CRTN & ALU ...ABUS.RAMP.OR.OR.DZ & REG ...RZERO..R0
CVALU

OSB11: (6) THE OPERAND, NOW IN THE QREG IS SENT TO MEMORY USING THE CONTENTS OF
FR5 AS ADDRESS. RETURN TAKES PLACE.
MAC ...FALSE.TZERO.CRTN & ALU ...MOP.OR.OR.ZQ & DBUS SALU
CREPORT & WRITE & REG ...R5

OSB11: (7) JUMP SUB TO OFD11. THIS FETCHES THE ADDRESS OF THE INDIRECT OPERAND TO
ITS DESTINATION.

OSB11: (8) THE CONTENTS OF FR5 ARE PLACED IN QREG.
MAC ...FALSE.TZERO.CJS & ALU ...ABUS.QREG.OR.OR.DZ & REG ...R5
CVALU & BADDRESS OFD11

OSB11: (9) THE OPERAND, NOW IN THE QREG IS SENT TO MEMORY USING THE CONTENTS OF
FR5 AS ADDRESS. RETURN TAKES PLACE.
MAC ...FALSE.TZERO.CRTN & ALU ...MOP.OR.OR.ZQ & DBUS SALU
CREPORT & WRITE & REG ...R5

OSB11: (10) THE CONTENTS OF THE QREG (MULTIPLIER) ARE MASKED BY THE CONTENTS OF
FR5 SO THAT ONLY THE SIGN BIT OF THE MULTIPLIER IS LEFT.

OSB11: (11) THE CONTENTS OF THE QREG (MULTIPLIER) ARE MASKED BY THE CONTENTS OF
FR5 SO THAT ONLY THE SIGN BIT OF THE MULTIPLIER IS LEFT.

1 MAY 1981

PAGE 30

AMDAS/29 AMDASM MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EMSL)

OSB11 -- THIS SUBROUTINE PLACES AN OPERAND IN ER5 INTO THE MEMORY
LOCATION POINTED TO BY THE CONTENTS OF ER2. NB ER2 WILL HOLD
THE ADDRESS OF THE LAST DIRECT OPERAND
AFTER THE EXECUTION OF OFD11, OFD12 OR OFD13. NB THE EXECUTION OF OSB11
DOES NOT RESULT IN A CHANGE OF THE PC VALUE IN ER0.

OSB11: (1) THE CONTENTS OF ER5 ARE PLACED IN QREG.
OSB11: (2) THE OPERAND, NOW IN THE QREG IS SENT TO THE ADDRESS IN MEMORY POINTED TO
BY ER2.
RETURN TAKES PLACE.

OSB11: (3) THE OPERAND, NOW IN THE QREG IS SENT TO THE ADDRESS IN MEMORY POINTED TO
BY ER2.
MAC ...FALSE.TZERO.CRTN & ALU ...MOP.OR.OR.ZQ & REG ...R2 & CREPORT
WRITE & DBUS SALU

OSB11: (4) THE OPERAND, NOW IN THE QREG IS SENT TO THE ADDRESS IN MEMORY POINTED TO
BY ER2.
MAC ...FALSE.TZERO.CRTN & ALU ...MOP.OR.OR.ZQ & REG ...R2 & CREPORT
WRITE & DBUS SALU

OSB11: (5) THE OPERAND, NOW IN THE QREG IS SENT TO THE ADDRESS IN MEMORY POINTED TO
BY ER2.
MAC ...FALSE.TZERO.CRTN & ALU ...MOP.OR.OR.ZQ & REG ...R2 & CREPORT
WRITE & DBUS SALU

OSB11: (6) THE OPERAND, NOW IN THE QREG IS SENT TO THE ADDRESS IN MEMORY POINTED TO
BY ER2.
MAC ...FALSE.TZERO.CRTN & ALU ...MOP.OR.OR.ZQ & REG ...R2 & CREPORT
WRITE & DBUS SALU

OSB11: (7) THE OPERAND, NOW IN THE QREG IS SENT TO THE ADDRESS IN MEMORY POINTED TO
BY ER2.
MAC ...FALSE.TZERO.CRTN & ALU ...MOP.OR.OR.ZQ & REG ...R2 & CREPORT
WRITE & DBUS SALU

OSB11: (8) THE OPERAND, NOW IN THE QREG IS SENT TO THE ADDRESS IN MEMORY POINTED TO
BY ER2.
MAC ...FALSE.TZERO.CRTN & ALU ...MOP.OR.OR.ZQ & REG ...R2 & CREPORT
WRITE & DBUS SALU

OSB11: (9) THE OPERAND, NOW IN THE QREG IS SENT TO THE ADDRESS IN MEMORY POINTED TO
BY ER2.
MAC ...FALSE.TZERO.CRTN & ALU ...MOP.OR.OR.ZQ & REG ...R2 & CREPORT
WRITE & DBUS SALU

OSB11: (10) THE OPERAND, NOW IN THE QREG IS SENT TO THE ADDRESS IN MEMORY POINTED TO
BY ER2.
MAC ...FALSE.TZERO.CRTN & ALU ...MOP.OR.OR.ZQ & REG ...R2 & CREPORT
WRITE & DBUS SALU

OSB11: (11) THE OPERAND, NOW IN THE QREG IS SENT TO THE ADDRESS IN MEMORY POINTED TO
BY ER2.
MAC ...FALSE.TZERO.CRTN & ALU ...MOP.OR.OR.ZQ & REG ...R2 & CREPORT
WRITE & DBUS SALU

OSB11: (12) THE CONTENTS OF THE QREG (MULTIPLIER) ARE MASKED BY THE CONTENTS OF
FR5 SO THAT ONLY THE SIGN BIT OF THE MULTIPLIER IS LEFT.

OSB11: (13) THE CONTENTS OF THE QREG (MULTIPLIER) ARE MASKED BY THE CONTENTS OF
FR5 SO THAT ONLY THE SIGN BIT OF THE MULTIPLIER IS LEFT.

OSB11: (14) THE CONTENTS OF THE QREG (MULTIPLIER) ARE MASKED BY THE CONTENTS OF
FR5 SO THAT ONLY THE SIGN BIT OF THE MULTIPLIER IS LEFT.

OSB11: (15) THE CONTENTS OF THE QREG (MULTIPLIER) ARE MASKED BY THE CONTENTS OF
FR5 SO THAT ONLY THE SIGN BIT OF THE MULTIPLIER IS LEFT.

OSB11: (16) THE CONTENTS OF THE QREG (MULTIPLIER) ARE MASKED BY THE CONTENTS OF
FR5 SO THAT ONLY THE SIGN BIT OF THE MULTIPLIER IS LEFT.

OSB11: (17) THE CONTENTS OF THE QREG (MULTIPLIER) ARE MASKED BY THE CONTENTS OF
FR5 SO THAT ONLY THE SIGN BIT OF THE MULTIPLIER IS LEFT.

OSB11: (18) THE CONTENTS OF THE QREG (MULTIPLIER) ARE MASKED BY THE CONTENTS OF
FR5 SO THAT ONLY THE SIGN BIT OF THE MULTIPLIER IS LEFT.

TM ES 400

1 MAY 1981

PAGE 31

PAGE 32

AMDOS/29 AMDASH MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EMSL)

```

/ 5 DBUS SALU 5 REG .....R6
: (3) A JUMP OCCURS TO ASSIGN1 IF THE MS BYTE OF THE PREVIOUS ALU OUTPUT WAS
: ZERO (IE. IF THE SIGN BIT WAS ZERO MEANING THAT THE MULTIPLIER IS POSITIVE).
00E5 / 5 MAC TRUE,ZMSB,CJP
/ 5 ADDRESS ASSIGN1 5 ALU
: (4) THE TEST CONTINUES IN THE KNOWLEDGE THAT THE MULTIPLIER IS NEGATIVE.
: THE Q REG IS LOADED WITH A SIGN-BIT-ONLY MASK.
00E6 / 5 MAC ...CONT
/ 5 ALU ...DBUS,QREG,OR,OR,DZ
/ 5 DBUS SDATA 5 DATINSRT D32768
: (5) THE MULTIPLIER IN ER5 IS NEGATED AND LEFT IN ER5.
00E7 / 5 MAC ...CONT
/ 5 ALU ...ANDUS,NOP,SUBR,SUBR,DZ
/ 5 REG ...R5 5 CREG 5 DBUS SALU
/ 5 STATSHFT B#010000000000
: (6) THE MASK IN QREG IS NOW AND'ED WITH THE NOT OF THE MULTIPLICAND IN GRI.
: THIS HAS THE EFFECT OF PRODUCING AN ALU OUTPUT WITH ALL BUT THE MS BIT CLEAR
: AND THE MS BIT BEING THE SIGN BIT FOR THE PRODUCT.
00E8 / 5 MAC ...CONT
/ 5 ALU ...NOP,NOTES,NOTES,AQ
/ 5 DBUS SALU 5 CREG
/ 5 REG ...GRI,GRI,R6
: (7) A RETURN OCCURS IF THE OUTPUT OF THE ALU HAD MS BYTE NOT ZERO. IE THE
: SIGN BIT WAS SET INDICATING THAT THE MULTIPLICAND WAS POSITIVE.
00E9 / 5 MAC ...FALSE,ZMSB,CRTN
/ 5 ALU
: (8) SINCE IT HAS NOW BEEN SHOWN THAT THE MULTIPLICAND IS NEGATIVE, IT IS
: NEGATED. MAC ...FALSE,TZERO,CRTN
00EA / 5 ALU ...RAMF,SUBS,SUBS,ZA
/ 5 REG ...GRI,GRI
/ 5 CKALU
/ 5 STATSHFT B#010000000000
: (9) THIS IS THE CONTINUATION WHEN IT HAS BEEN SHOWN THAT THE MULTIPLIER
: IS POSITIVE. MAC ...CONT
00EB ASSIGN1: MAC ...ANDUS,NOP,AND,AND,DA
/ 5 ALU ...ZRI,GRI,R6
/ 5 REG ...ZRI,GRI,R6
/ 5 DBUS SALU 5 CREG

```

1 MAY 1981

PAGE 31

PAGE 32

AMDOS/29 AMDASH MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EMSL)

```

: (9) RETURN OCCURS IF THE SIGN BIT WAS ZERO.
00EC / 5 MAC ...TRUE,ZMSB,CRTN
/ 5 ALU
: (10) THE MULTIPLICAND IN GRI IS NEGATED AND RETURN TAKES PLACE.
00ED / 5 MAC ...FALSE,TZERO,CRTN
/ 5 ALU ...RAMF,SUBS,SUBS,ZA
/ 5 REG ...GRI,GRI
/ 5 CKALU
/ 5 STATSHFT B#010000000000
: THIS SUBROUTINE MULTIPLIES A SINGLE PRECISION POSITIVE INTEGER IN GRI BY
: ANOTHER SINGLE PRECISION POSITIVE INTEGER IN ER5. THE RESULT, A DOUBLE
: PRECISION POSITIVE INTEGER IS LEFT IN GRI (MS WORD) AND QREG (LS WORD).
: THE MULTIPLIER IN ER5 IS PLACED IN QREG.
00EE ASMULT: MAC ...CONT
/ 5 ALU ...ANDUS,QREG,OR,OR,DZ
/ 5 REG ...R5
/ 5 CKALU
: THE MULTIPLICAND IS PLACED IN ER9
00EF / 5 MAC ...CONT
/ 5 ALU ...NOP,OR,OR,ZA
/ 5 REG ...GRI,GRI,R6
/ 5 DBUS SALU 5 CREG
: (20) THE MULTIPLICATION PROCESS BEGINS.
: GRI IS CLEARED, GRI AND QREG ARE DOWNSHIFTED. ZERO IS PUT IN AT THE TOP
: OF GRI. THE MICRO-PROGRAM COUNTER VALUE PLUS ONE IS PUSHED INTO THE
: MICRO-PROGRAM STACK (ZRI 5 2009).
: THE NUMBER D#15 IS LOADED INTO THE COUNTER IN THE CCU.
: THE INTERNAL CARRY LATCH IS LOADED WITH THE LS BIT FROM THE QREG.
00F0 SPIM2: MAC ...FALSE,TZERO,PUSH
/ 5 ALU ...RAMQD,AND,AND,ZA
/ 5 REG ...GRI,GRI
/ 5 CKALU
/ 5 CARRYSEL CQLSR
/ 5 ADDRESS B#000000001110

```

1 MAY 1981

PAGE 33

PAGE 34

AMDOS/29 AMDASH MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EML)

00P1 / 5 JUMP TO SP1M3 TAKES PLACE IF THE CARRY BIT WAS SET MEANING THAT THE
/ 5 LS BIT OF THE Q REGISTER WAS ONE.

00P2 / 5 MAC TRUE CARRY, CJP
/ 5 ADDRESS SP1M3
/ 5 ALU

00P3 / 5 (22) GRI AND QREG ARE DOWNSHIFTED. THE LS BIT OF GRI GOES INTO THE MS
/ 5 BIT OF QREG. CARRY IS SET ACCORDING TO THE LS BIT OF THE QREG.
/ 5 IF THE COUNTER IS ZERO CONTINUE OCCURS.
/ 5 IF THE COUNTER IS NOT ZERO, IT IS DECREMENTED AND JUMP TAKES PLACE TO
/ 5 THE ADDRESS HELD IN THE STACK WHICH IS (21) ABOVE.

00P4 / 5 MAC FALSE COUNT, RPTC
/ 5 ALU
/ 5 REG .GRI, GRI, RQ
/ 5 CARRYSEL COLSB
/ 5 STATSHFT B0000110000000
/ 5 CICALU

00P5 / 5 (23) THE MULTIPLICATION IS NOW COMPLETE AND AN UNCONDITIONAL RETURN TAKES PLACE
/ 5 MAC FALSE, ZERO, CPTN
/ 5 ALU

00P6 / 5 (24) THE MULTIPLICAND IN R19 IS ADDED TO THE PARTIAL PRODUCT IN 2P1.
/ 5 THE CONTENTS OF GRI AND QREG ARE DOWNSHIFTED.
/ 5 THE LS BIT OF GRI GOES INTO THE MS BIT OF THE QREG.
/ 5 LS BIT OF QREG GOES INTO THE R1 CARRY LATCH.
/ 5 IF THE COUNTER IS ZERO CONTINUE TAKES PLACE.
/ 5 IF THE COUNTER IS NOT ZERO, THE COUNTER IS DECREMENTED AND JUMP
/ 5 TO THE ADDRESS IN THE STACK TAKES PLACE WHICH IS TO
/ 5 (21) ABOVE.

00P7 / 5 MAC FALSE COUNT, RPTC
/ 5 ALU
/ 5 REG .R19S, R19Q, ADD, ADD, DA
/ 5 CARRYSEL COLSB
/ 5 STATSHFT B0000110000000
/ 5 CICALU
/ 5 DBUS QREG

00P8 / 5 (25) THE MULTIPLICATION IS NOW COMPLETE AND UNCONDITIONAL RETURN TAKES PLACE
/ 5 MAC FALSE, ZERO, CPTN
/ 5 ALU

1 MAY 1981

PAGE 34

AMDOS/29 AMDASH MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EML)

ASDLSTG2 --- THIS SUBROUTINE SETS THE MACHINE STATUS REGISTER CORRECTLY
FOR A DOUBLE LENGTH 2'S COMPLEMENT WORD IN GR2 AND GR2 + 1 WITH
THE MOST SIGNIFICANT HALF IN GR2.

((1)) THE MACHINE STATUS REGISTER IS SET ACCORDING TO THE CONTENTS
OF GR2 (THE MOST SIGNIFICANT WORD). CARRY IS SET TO ZERO.

00P9 / 5 MAC .COMT
/ 5 ALU
/ 5 CARRYSEL CZERO
/ 5 STATSHFT B000000010000

((2)) THE CONTENTS OF THE Q REG ARE LOADED INTO GR2+1
THE MICRO STATUS REGISTER IS LOADED ACCORDINGLY.

00P7 / 5 MAC .COMT
/ 5 ALU
/ 5 CICALU
/ 5 STATSHFT B000000010000

((3)) RETURN TAKES PLACE IF THE PREVIOUS LOADING OF THE MICRO STATUS REGISTER
SET MICRO 2 IMPLYING THAT THE LS WORD OF THE PAIR WAS ZERO.

00P8 / 5 MAC .TRUE, MUL, CRTN
/ 5 ALU
/ 5 REG .GR2, STATSHFT B000000000100

((4)) RETURN TAKES PLACE IF THE ZERO BIT OF THE MACHINE STATUS REGISTER IS
NOT SET.

00P9 / 5 MAC .FALSE, MUL, CRTN
/ 5 ALU
/ 5 STATSHFT B000000000100

((5)) WE NOW KNOW THAT THE MACHINE STATUS REGISTER HAS ITS ZERO BIT SET. THIS
IS INCORRECT FOR THE DOUBLE LENGTH RESULT SINCE THE LEAST SIGNIFICANT
WORD IS NOT ZERO. THE CORRECT SETTING OF THE MACHINE STATUS
WILL BE WITH THE POSITIVE BIT SET. THIS CAN BE ACHIEVED IN ONE OPERATION
BY INVERTING THE MACHINE STATUS REGISTER WITH ONLY M2 AND M4 ENABLED.

00P4 / 5 MAC .COMT
/ 5 ALU
/ 5 STATSHFT B000000000100

00P5 / 5 MAC .FALSE, ZERO, CPTN
/ 5 ALU

AMDOS/29 AMDASH MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EMSL

OVERFLOW ROUTINE

```

00FC OVERFLOW: MAC ,COUNT $ ALU & LPUIS SDATA 5 CYREG
               ;
               ;(1) THE PATTERN 0000100000000000 IS LOADED INTO THE REG

```

DATINSRT 002049 & REL. 0000

2) THE CONTENTS OF THE INTERRUPT REGISTER.

```
MAC, FALSE, ZERO, CRN & ALU & DEUS
MAC, FALSE, ZERO, CRN & ALU & DEUS
```

ENTRY POINTS TO PITTMED OPCODE HANDLERS

[illegible][illegible]

... SUPERBASS ALL S. BCS + CPE

VERBILADD -- THIS ROUTINE RECEIVES THE CONTENTS OF THE MEMORY LOCATION FOLLOWING THAT OF THE CURRENT LOCATION. IF THE SECOND WORD OF A TWO WORD INSTRUCTION, THE WORD IS PLACED IN R2, AND THE CONTENTS OF THE SPECIFIED INDEX REGISTER IS ADDED TO IT (IF ANY). THE CONTENTS OF R2 THEN REPRESENTS THE CORRECT ADDRESS FOR THE FIRST WORD (OR ONLY WORD) OF A DIACLO, OR DIRECT-INDIRECT DELIVER OPERAND. THE PROGRAM COUNTER IS INCREMENTED TO THE NEXT INSTRUCTION. THE NEXT INSTRUCTION WILL BE FOUND NEXT BUT NOT NEARBY. HAVING CARRIED OUT THIS PROCESS, THE NEXT INSTRUCTION WILL BE FOUND NEXT BUT NOT NEARBY.

MAY 1981

pack 75

AMCOS/29 AMDASM MICRO ASSEMBLY, V.1.0
EMULATION SEQUENCES FOR SINGLE INCH SUBSIT FOR 1750. (MSL
... OUT TO THE PRESENT INSTRUCTION.

ONE TO THE PRESENT INSTRUCTION.

```

.....
(1) MEMORY ACCESS FOR READ OF TABS PLACE, THE PROGRAM COUNTER VALUE
    IS INCREMENTED BY 1.
.....

```

IS INCREMENTED BY ONE AND ALL APES, NO. 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356, 357, 358, 359, 360, 361, 362, 363, 364, 365, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375, 376, 377, 378, 379, 380, 381, 382, 383, 384, 385, 386, 387, 388, 389, 390, 391, 392, 393, 394, 395, 396, 397, 398, 399, 400, 401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 418, 419, 420, 421, 422, 423, 424, 425, 426, 427, 428, 429, 430, 431, 432, 433, 434, 435, 436, 437, 438, 439, 440, 441, 442, 443, 444, 445, 446, 447, 448, 449, 450, 451, 452, 453, 454, 455, 456, 457, 458, 459, 460, 461, 462, 463, 464, 465, 466, 467, 468, 469, 470, 471, 472, 473, 474, 475, 476, 477, 478, 479, 480, 481, 482, 483, 484, 485, 486, 487, 488, 489, 490, 491, 492, 493, 494, 495, 496, 497, 498, 499, 500, 501, 502, 503, 504, 505, 506, 507, 508, 509, 510, 511, 512, 513, 514, 515, 516, 517, 518, 519, 520, 521, 522, 523, 524, 525, 526, 527, 528, 529, 530, 531, 532, 533, 534, 535, 536, 537, 538, 539, 540, 541, 542, 543, 544, 545, 546, 547, 548, 549, 550, 551, 552, 553, 554, 555, 556, 557, 558, 559, 560, 561, 562, 563, 564, 565, 566, 567, 568, 569, 570, 571, 572, 573, 574, 575, 576, 577, 578, 579, 580, 581, 582, 583, 584, 585, 586, 587, 588, 589, 590, 591, 592, 593, 594, 595, 596, 597, 598, 599, 600, 601, 602, 603, 604, 605, 606, 607, 608, 609, 610, 611, 612, 613, 614, 615, 616, 617, 618, 619, 620, 621, 622, 623, 624, 625, 626, 627, 628, 629, 630, 631, 632, 633, 634, 635, 636, 637, 638, 639, 640, 641, 642, 643, 644, 645, 646, 647, 648, 649, 650, 651, 652, 653, 654, 655, 656, 657, 658, 659, 660, 661, 662, 663, 664, 665, 666, 667, 668, 669, 670, 671, 672, 673, 674, 675, 676, 677, 678, 679, 680, 681, 682, 683, 684, 685, 686, 687, 688, 689, 690, 691, 692, 693, 694, 695, 696, 697, 698, 699, 700, 701, 702, 703, 704, 705, 706, 707, 708, 709, 710, 711, 712, 713, 714, 715, 716, 717, 718, 719, 720, 721, 722, 723, 724, 725, 726, 727, 728, 729, 730, 731, 732, 733, 734, 735, 736, 737, 738, 739, 740, 741, 742, 743, 744, 745, 746, 747, 748, 749, 750, 751, 752, 753, 754, 755, 756, 757, 758, 759, 760, 761, 762, 763, 764, 765, 766, 767, 768, 769, 770, 771, 772, 773, 774, 775, 776, 777, 778, 779, 780, 781, 782, 783, 784, 785, 786, 787, 788, 789, 790, 791, 792, 793, 794, 795, 796, 797, 798, 799, 800, 801, 802, 803, 804, 805, 806, 807, 808, 809, 810, 811, 812, 813, 814, 815, 816, 817, 818, 819, 820, 821, 822, 823, 824, 825, 826, 827, 828, 829, 830, 831, 832, 833, 834, 835, 836, 837, 838

78
CUBA
IS PLACED IN THE

(12) THE CONTENTS OF THIS DOCUMENT ARE UNCLASSIFIED
DATE 08-09-2001 BY SP-6 PJA/MAG

5 / 1018

(7) THE CONTENTS OF 182 ARE ADDED TO THE CONTENTS OF THE MEMORY DATA REGISTER AND THE RESULT PLACED IN 180. 181 IS ZERO AND 182 REMAINS 6 CCALU

2102
 2103
 2104
 2105
 2106
 2107
 2108
 2109
 2110
 2111
 2112
 2113
 2114
 2115
 2116
 2117
 2118
 2119
 2120
 2121
 2122
 2123
 2124
 2125
 2126
 2127
 2128
 2129
 2130
 2131
 2132
 2133
 2134
 2135
 2136
 2137
 2138
 2139
 2140
 2141
 2142
 2143
 2144
 2145
 2146
 2147
 2148
 2149
 2150
 2151
 2152
 2153
 2154
 2155
 2156
 2157
 2158
 2159
 2160
 2161
 2162
 2163
 2164
 2165
 2166
 2167
 2168
 2169
 2170
 2171
 2172
 2173
 2174
 2175
 2176
 2177
 2178
 2179
 2180
 2181
 2182
 2183
 2184
 2185
 2186
 2187
 2188
 2189
 2190
 2191
 2192
 2193
 2194
 2195
 2196
 2197
 2198
 2199
 2200
 2201
 2202
 2203
 2204
 2205
 2206
 2207
 2208
 2209
 2210
 2211
 2212
 2213
 2214
 2215
 2216
 2217
 2218
 2219
 2220
 2221
 2222
 2223
 2224
 2225
 2226
 2227
 2228
 2229
 2230
 2231
 2232
 2233
 2234
 2235
 2236
 2237
 2238
 2239
 2240
 2241
 2242
 2243
 2244
 2245
 2246
 2247
 2248
 2249
 2250
 2251
 2252
 2253
 2254
 2255
 2256
 2257
 2258
 2259
 2260
 2261
 2262
 2263
 2264
 2265
 2266
 2267
 2268
 2269
 2270
 2271
 2272
 2273
 2274
 2275
 2276
 2277
 2278
 2279
 2280
 2281
 2282
 2283
 2284
 2285
 2286
 2287
 2288
 2289
 2290
 2291
 2292
 2293
 2294
 2295
 2296
 2297
 2298
 2299
 2300
 2301
 2302
 2303
 2304
 2305
 2306
 2307
 2308
 2309
 2310
 2311
 2312
 2313
 2314
 2315
 2316
 2317
 2318
 2319
 2320
 2321
 2322
 2323
 2324
 2325
 2326
 2327
 2328
 2329
 2330
 2331
 2332
 2333
 2334
 2335
 2336
 2337
 2338
 2339
 2340
 2341
 2342
 2343
 2344
 2345
 2346
 2347
 2348
 2349
 2350
 2351
 2352
 2353
 2354
 2355
 2356
 2357
 2358
 2359
 2360
 2361
 2362
 2363
 2364
 2365
 2366
 2367
 2368
 2369
 2370
 2371
 2372
 2373
 2374
 2375
 2376
 2377
 2378
 2379
 2380
 2381
 2382
 2383
 2384
 2385
 2386
 2387
 2388
 2389
 2390
 2391
 2392
 2393
 2394
 2395
 2396
 2397
 2398
 2399
 2400
 2401
 2402
 2403
 2404
 2405
 2406
 2407
 2408
 2409
 2410
 2411
 2412
 2413
 2414
 2415
 2416
 2417
 2418
 2419
 2420
 2421
 2422
 2423
 2424
 2425
 2426
 2427
 2428
 2429
 2430
 2431
 2432
 2433
 2434
 2435
 2436
 2437
 2438
 2439
 2440
 2441
 2442
 2443
 2444
 2445
 2446
 2447
 2448
 2449
 2450
 2451
 2452
 2453
 2454
 2455
 2456
 2457
 2458
 2459
 2460
 2461
 2462
 2463
 2464
 2465
 2466
 2467
 2468
 2469
 2470
 2471
 2472
 2473
 2474
 2475
 2476
 2477
 2478
 2479
 2480
 2481
 2482
 2483
 2484
 2485
 2486
 2487
 2488
 2489
 2490
 2491
 2492
 2493
 2494
 2495
 2496
 2497
 2498
 2499
 2500
 2501
 2502
 2503
 2504
 2505
 2506
 2507
 2508
 2509
 2510
 2511
 2512
 2513
 2514
 2515
 2516
 2517
 2518
 2519
 2520
 2521
 2522
 2523
 2524
 2525
 2526
 2527
 2528
 2529
 2530
 2531
 2532
 2533
 2534
 2535
 2536
 2537
 2538
 2539
 2540
 2541
 2542
 2543
 2544
 2545
 2546
 2547
 2548
 2549
 2550
 2551
 2552
 2553
 2554
 2555
 2556

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
84

(4) THE CONTENTS OF THE ORIGINAL CONTENTS OF THE ART ARE LEFT IN THE ORIGINAL CONTENTS OF THE ART.

2103
HISTORICAL
NAC, VALS, TYPE, CITH S KU, ARUS, PAMA, OF, OR, BY S CANUS B CANES

21 JUL 1954

EXTENDED OPCODE HANDLER:

ALIGN 16

LINE	SWITCH#0:	MAC	..IP	..AU	..ADDRESS	..CNAME
0100	0100	MAC ..IP	..AU	..ADDRESS	..CNAME	..CNAME
0101	0101	MAC ..IP	..AU	..ADDRESS	..CNAME	..CNAME
0102	0102	MAC ..IP	..AU	..ADDRESS	..CNAME	..CNAME
0103	0103	MAC ..IP	..AU	..ADDRESS	..CNAME	..CNAME
0104	0104	MAC ..IP	..AU	..ADDRESS	..CNAME	..CNAME
0105	0105	MAC ..IP	..AU	..ADDRESS	..CNAME	..CNAME
0106	0106	MAC ..IP	..AU	..ADDRESS	..CNAME	..CNAME
0107	0107	MAC ..IP	..AU	..ADDRESS	..CNAME	..CNAME
0108	0108	MAC ..IP	..AU	..ADDRESS	..CNAME	..CNAME
0109	0109	MAC ..IP	..AU	..ADDRESS	..CNAME	..CNAME
0110	0110	MAC ..IP	..AU	..ADDRESS	..CNAME	..CNAME
0111	0111	MAC ..IP	..AU	..ADDRESS	..CNAME	..CNAME
0112	0112	MAC ..IP	..AU	..ADDRESS	..CNAME	..CNAME
0113	0113	MAC ..IP	..AU	..ADDRESS	..CNAME	..CNAME
0114	0114	MAC ..IP	..AU	..ADDRESS	..CNAME	..CNAME
0115	0115	MAC ..IP	..AU	..ADDRESS	..CNAME	..CNAME
0116	0116	MAC ..IP	..AU	..ADDRESS	..CNAME	..CNAME
0117	0117	MAC ..IP	..AU	..ADDRESS	..CNAME	..CNAME
0118	0118	MAC ..IP	..AU	..ADDRESS	..CNAME	..CNAME
0119	0119	MAC ..IP	..AU	..ADDRESS	..CNAME	..CNAME
0120	0120	MAC ..IP	..AU	..ADDRESS	..CNAME	..CNAME

1 MAY 1961

PAGE 37

AMDAS/20 AMDASH MICRO ASSEMBLER, V1.0

EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (ENSL)

0110 MAC ...JP & ALU & ADDRESS IOPCD40C
 0111 MAC ...JP & ALU & ADDRESS ILEGOP1
 0112 MAC ...JP & ALU & ADDRESS IOPCD40C
 0113 MAC ...JP & ALU & ADDRESS IOPCD40C
 0114 MAC ...JP & ALU & ADDRESS IOPCD40C

0120 SWITCH64:
 0121 MAC ...JP & ALU & ADDRESS ILEGOP
 0122 MAC ...JP & ALU & ADDRESS IOPCD41
 0123 MAC ...JP & ALU & ADDRESS IOPCD42
 0124 MAC ...JP & ALU & ADDRESS IOPCD43
 0125 MAC ...JP & ALU & ADDRESS IOPCD44
 0126 MAC ...JP & ALU & ADDRESS IOPCD45
 0127 MAC ...JP & ALU & ADDRESS IOPCD46
 0128 MAC ...JP & ALU & ADDRESS IOPCD47
 0129 MAC ...JP & ALU & ADDRESS IOPCD48
 0130 MAC ...JP & ALU & ADDRESS IOPCD49
 0131 MAC ...JP & ALU & ADDRESS IOPCD4A
 0132 MAC ...JP & ALU & ADDRESS IOPCD4B
 0133 MAC ...JP & ALU & ADDRESS IOPCD4C
 0134 MAC ...JP & ALU & ADDRESS ILEGOP1
 0135 MAC ...JP & ALU & ADDRESS ILEGOP1
 0136 MAC ...JP & ALU & ADDRESS ILEGOP1
 0137 MAC ...JP & ALU & ADDRESS ILEGOP1
 0138 MAC ...JP & ALU & ADDRESS ILEGOP1
 0139 MAC ...JP & ALU & ADDRESS ILEGOP1

INSTR/OPCODE INSTRUCTION

ALIGN 16

(1) JUMP SUB TO PTDIADD TO FETCH THE DIRECT/DIRECT-INDEXED ADDRESS WHICH WILL
 BE USED AS THE I/O COMMAND.

0136 OPCODE48: MAC .FALSE, ZERO, CJS & ALU & ADDRESS PTDIADD

(2) THE CONTENTS OF REGISTER GR1 ARE TRANSFERRED TO ZRS.

0131 / & MAC ...CONT & ALU ...MOP, OR, OR, ZA & REG ...JRI, R5 & DBUS SALU
 CEREK

(3) GR1 IS LOADED WITH THE I/O COMMAND FROM THE.

0132 / & MAC .FALSE, ZERO, PUSH & ALU ...ABUS, RAMP, OR, OR, DZ & REG ...CRI, R2
 CREAL & DATIMST D02 & DBUS SDATA

(4) THE CONTENTS OF GR1 ARE UPSHIFTED 4 TIMES.

1 MAY 1961

PAGE 38

AMDAS/20 AMDASH MICRO ASSEMBLER, V1.0

EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (ENSL)

0133 / & MAC .FALSE, CONT, RECT & ALU ...RAMP, OR, OR, ZA & REG ...CRI, R1 & CREAL
 STATSETT D0001000000000000

(5) THE INSTRUCTION REGISTER IS LOADED WITH THE CONTENTS OF GR1. GR1 IS
 LOADED WITH ITS ORIGINAL CONTENTS FROM ZRS.

0134 / & MAC ...CONT & ALU ...ABUS, RAMP, OR, OR, DZ & REG ...CRI, R1, R5 & CREAL
 DBUS SALU

(6) IN ORDER TO BRANCH TO AN ADDRESS DEPENDENT ON THE MOST SIGNIFICANT FOUR
 BITS OF THE I/O COMMAND, A JUMP TAKES PLACE MODIFIED BY THE GR2 FIELD OF THE
 INSTRUCTION REGISTER CONTENTS (IE. IN 1750 NUMBERING, BITS 0 THROUGH 3 OF
 THE I/O COMMAND).

0135 MAC B01, ...JP & ALU & REG ...GR2 & BADDRESS IORENCH1

ILLEGAL OPCODE ROUTINE

(1) THE REGISTER IS LOADED WITH A WORD WHICH HAS A ONE IN THE ILLEGAL OPCODE
 BIT POSITION FOR THE FAULT REGISTER AND ZERO EVERYWHERE ELSE.

0136 ILLEGOP1: MAC ...CONT & ALU ...DBUS, OREG, OR, OR, DZ & DBUS SDATA
 DATIMST D01024 & CREAL

(2) THE CONTENTS OF THE FAULT REGISTER (R03) IS ORED WITH THE WORD IN THE QREG
 SO THAT THE ILLEGAL OPCODE BIT IS SET IN THE FAULT REG.

0137 ILLEGOP2: MAC ...CONT & ALU ...ABUS, MOP, OR, OR, DQ & DBUS SALU & REG ...R3
 CEREK

(3) JUMP TAKES PLACE TO FAULTING. THE MICROSTACK IS POPPED.

0138 MAC .FALSE, ZERO, CJP & ALU & BADDRESS FAULTINT

THE NEXT 4 INSTRUCTIONS ELIMINATE I/O COMMANDS OTHER THAN IOPCD.

BRANCH TO ILLEGOP TAKES PLACE FOR OTHER COMMANDS.
 THE NORMAL RETURN FROM THIS ROUTINE IS MODIFIED BY THE CONTENTS OF
 THE LEAST SIGNIFICANT FOUR BITS OF THE I/O COMMAND. THE RETURN ADDRESS
 MUST THEREFORE BE A MULTIPLE OF 16 FOR THE ROUTINE TO WORK CORRECTLY.

THE QREG IS ALSO LOADED.

0139 DECH1001: MAC ...CONT & ALU ...ABUS, OREG, OR, OR, DZ & CREAL & DBUS SREG

TM PS 403

PAGE 40

```
0150 RINAMEI: MACP ...CONT & ALU & DRUS SREG & REG ....RI  

(2) THE MACHINE INTERRUPT MASK IN RI15 IS LOADED INTO OPS.  

MAC .....CONT & ALU ...DRUS ,ORIG, OR, DZ & CVALU  

REG .....RIS / S  

(3) THE CONTENTS OF THE REG ARE INTERRED AND PLACED IN CR1.  

MAC PULSE, ZIRZOCAN & ALU ...BLAP, FINOR, ZINOR ZO & CVALU & REG ..GRI  

-----  

READ STATUS WORD  

*****  

(1) THE INSTRUCTION WORD IN IPI IS PLACED BACK IN THE INSTRUCION REGISTER.  

ASTARD: MACP ...CONT & ALU & DRUS SREG & REG ....RI  

(2) THE STATUS WORT IS READ AND PLACED IN CR1. RETURN TAKES PLACE  

MAC PULSE, ZIRZOCAN & ALU ...DRUS RAMI, OR, DZ & REG ..GRI  

CVALU & REG, SREG, STT-SRT, #00000100000  

/
```

READ AND CLEAR FAULT REGISTER

```

; (1) THE INSTRUCTION WORD IN R1 IS PLACED IN THE INSTRUCTION RT?
; RESULT:  NAC ...CONT & ALU & DRUS SREG & REG ...R1
Q0155 />

; (2) THE CONTENTS OF THE FAULT REGISTER ARE LOADED INTO CR1, THE FAULT RTC (FRV)
; IS CLEARED. RETURN TAKES PLACE.
; NAC ...FALSE ...ZERO ...CEN & ALU ...BUSY ...PAMP ...OR ...DZ & DRUS SDATA
Q0156 /> PC ...CR1 ...RT & CALL & CHRG & DATINGP ...DEB

```

READ CPU STATE WORD (NOT STATUS!!)

0157 RSTATE: NACP ...CONT & ACU & PRUS SREG & REG ...R1

(2) THE MACHINE STATE WORD IN EP41 IS PLACED IN GR1 - RETURN TAKES PLACE.

0158 MAC PAID TO ENTER CPU'S ACU .ARUS RAM FOR OR,02 6 REG ,GR1,R4

1 MAY 1981

PAGE 41

AMDS/29 AMDASH MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EMSL)

```

/ 6      CALLU
-----
READ ENGINEER'S SWITCHFIELD
*****
(1) THE INSTRUCTION IN ER1 IS LOADED INTO THE INSTRUCTION REGISTER.
  RMANDT:  MACF ...CONT & ALU & DBUS SREG & REG ...R1
(2) THE MANUAL DATA SWITCHFIELD ON THE ENGINEER'S CONTROL BOX IS LOADED
  INTO ER1.
  RETURN TAKES PLACE.
  MACF, FALSE, ZERO, CRTN & ALU ...DBUS, RAMP, OR, OR, DZ & DBUS SHANDT
  CALLU & REG ...R1
/ 6
-----
SET PENDING INTERRUPT REGISTER
*****
(1) THE CONTENTS OF THE PENDING INTERRUPT REGISTER IN BOTH MACHINE INTERRUPT
  UNITS ARE CLEARED.
  0150 STPDINT:  MACF ...CONT & ALU & DBUS SREG & REG ...R1 & MACINT CLRIN
  / 5      ENAUIK & STATSHFT 000000011000
(2) THE CONTENTS OF CR1 ARE LOADED INTO THE PENDING INTERRUPT REGISTER.
  015C      RETURN TAKES PLACE.
  / 5      MACF, FALSE, ZERO, CRTN & ALU ...NOP, OR, OR, DZ & REG ...CR1 & DBUS SALU
  / 5      ENAUIK & STATSHFT 000100000000
  ORG RINTMSK-10-1
  PROCESSOR AND AUXILIARY REGISTER CONTROL ( COMMAND GROUP 0001 )
  THIS SECTION CAUSES BRANCHING TO THE INDIVIDUAL INPUT ROUTINES DEPENDENT
  ON THE LAST FOUR BITS OF THE I/O COMMAND.
(1) A JUMP IS MADE TO BRCH00K IN ORDER TO ELIMINATE THE CASE WHERE THE
  015F CENTRE 8 BITS OF THE COMMAND WORD IS NOW ZERO.
  015F INPUTR1:  MACF, FALSE, ZERO, CRTN & ALU & ADDRESS BRCH00K
  0160
  / 5      MACF ...JP & ALU & ADDRESS RINTMSK ; CMD 0000

```

1 MAY 1981

PAGE 42

AMDS/29 AMDASH MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EMSL)

```

0161      MACF ...JP & ALU & ADDRESS RSTATE ; CMD 0001
0162      MACF ...JP & ALU & ADDRESS RMANDT ; CMD 0002
0163      MACF ...JP & ALU & ADDRESS ILLR00P ; CMD 0003
0164      MACF ...JP & ALU & ADDRESS ILLR00P ; CMD 0004
0165      MACF ...JP & ALU & ADDRESS ILLR00P ; CMD 0005
0166      MACF ...JP & ALU & ADDRESS ILLR00P ; CMD 0006
0167      MACF ...JP & ALU & ADDRESS ILLR00P ; CMD 0007
0168      MACF ...JP & ALU & ADDRESS ILLR00P ; CMD 0008
0169      MACF ...JP & ALU & ADDRESS ILLR00P ; CMD 0009
016A      MACF ...JP & ALU & ADDRESS ILLR00P ; CMD 000A
016B      MACF ...JP & ALU & ADDRESS ILLR00P ; CMD 000B
016C      MACF ...JP & ALU & ADDRESS ILLR00P ; CMD 000C
016D      MACF ...JP & ALU & ADDRESS RSTATWRD ; CMD 000D
016E      MACF ...JP & ALU & ADDRESS RSTATWRD ; CMD 000E
016F      MACF ...JP & ALU & ADDRESS RSTATWRD ; CMD 000F
-----
PIO OUTPUT
*****
(1) THE INSTRUCTION IS RELOADED INTO THE INSTRUCTION REGISTER.
  0170 OUTPUT0:  MACF ...CONT & ALU & DBUS SREG & REG ...R1
  (1A) A MASK WITH ALL BUT THE TOP FOUR BITS SET IS LOADED INTO THE Q REG.
  / 5      MACF ...CONT & ALU ...DBUS, OR, OR, DZ & DBUS SDATA & CALLU
  / 5      DATINSET D#0005
  (1B) THE CONTENTS OF ER2 ARE AND'ED WITH THE CONTENTS OF THE Q REG AND THE
  / 5      RESULT PLACED BACK IN ER2.
  / 5      MACF ...CONT & ALU ...ABUS, NOP, AND, AND, DQ & CEREG & DBUS SALU
  / 5      REG ...R2
  (2) THE CONTENTS OF CR1 ARE OUTPUT ON THE MEMORY BUS USING THE CONTENTS
  / 5      OF ER2 AS THE ADDRESS. BIT 0 OF THE AUIK 1 FIELD IS HELD HIGH TO SHOW
  / 5      THAT THE OPERATION IS AN I/O OPERATION AND NOT A MEMORY OPERATION.
  / 5      MACF, FALSE, ZERO, CRTN & ALU ...NOP, OR, OR, DZ & REG ...CR1, R2 & DBUS SALU
  / 5      REPORT & WRITE & ENAUIK & STATSHFT 000010000000
  (1) THE INSTRUCTION IS RELOADED INTO THE INSTRUCTION REGISTER.
  PIO INPUT
  *****

```

TM FS 403

PAGE 44

AMDOS/29 AMDASH MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EWSL

PROCESSOR AND AUXILIARY REGISTER CONTROL (COMMAND GROUP C00X)

1
 2
 3
 4
 5
 6
 7
 8
 9
 10
 11
 12
 13
 14
 15
 16
 17
 18
 19
 20
 21
 22
 23
 24
 25
 26
 27
 28
 29
 30
 31
 32
 33
 34
 35
 36
 37
 38
 39
 40
 41
 42
 43
 44
 45
 46
 47
 48
 49
 50
 51
 52
 53
 54
 55
 56
 57
 58
 59
 60
 61
 62
 63
 64
 65
 66
 67
 68
 69
 70
 71
 72
 73
 74
 75
 76
 77
 78
 79
 80
 81
 82
 83
 84
 85
 86
 87
 88
 89
 90
 91
 92
 93
 94
 95
 96
 97
 98
 99
 100
 101
 102
 103
 104
 105
 106
 107
 108
 109
 110
 111
 112
 113
 114
 115
 116
 117
 118
 119
 120
 121
 122
 123
 124
 125
 126
 127
 128
 129
 130
 131
 132
 133
 134
 135
 136
 137
 138
 139
 140
 141
 142
 143
 144
 145
 146
 147
 148
 149
 150
 151
 152
 153
 154
 155
 156
 157
 158
 159
 160
 161
 162
 163
 164
 165
 166
 167
 168
 169
 170
 171
 172
 173
 174
 175
 176
 177
 178
 179
 180
 181
 182
 183
 184
 185
 186
 187
 188
 189
 190
 191
 192
 193
 194
 195
 196
 197
 198
 199
 200
 201
 202
 203
 204
 205
 206
 207
 208
 209
 210
 211
 212
 213
 214
 215
 216
 217
 218
 219
 220
 221
 222
 223
 224
 225
 226
 227
 228
 229
 230
 231
 232
 233
 234
 235
 236
 237
 238
 239
 240
 241
 242
 243
 244
 245
 246
 247
 248
 249
 250
 251
 252
 253
 254
 255
 256
 257
 258
 259
 260
 261
 262
 263
 264
 265
 266
 267
 268
 269
 270
 271
 272
 273
 274
 275
 276
 277
 278
 279
 280
 281
 282
 283
 284
 285
 286
 287
 288
 289
 290
 291
 292
 293
 294
 295
 296
 297
 298
 299
 300
 301
 302
 303
 304
 305
 306
 307
 308
 309
 310
 311
 312
 313
 314
 315
 316
 317
 318
 319
 320
 321
 322
 323
 324
 325
 326
 327
 328
 329
 330
 331
 332
 333
 334
 335
 336
 337
 338
 339
 340
 341
 342
 343
 344
 345
 346
 347
 348
 349
 350
 351
 352
 353
 354
 355
 356
 357
 358
 359
 360
 361
 362
 363
 364
 365
 366
 367
 368
 369
 370
 371
 372
 373
 374
 375
 376
 377
 378
 379
 380
 381
 382
 383
 384
 385
 386
 387
 388
 389
 390
 391
 392
 393
 394
 395
 396
 397
 398
 399
 400
 401
 402
 403
 404
 405
 406
 407
 408
 409
 410
 411
 412
 413
 414
 415
 416
 417
 418
 419
 420
 421
 422
 423
 424
 425
 426
 427
 428
 429
 430
 431
 432
 433
 434
 435
 436
 437
 438
 439
 440
 441
 442
 443
 444
 445
 446
 447
 448
 449
 450
 451
 452
 453
 454
 455
 456
 457
 458
 459
 460
 461
 462
 463
 464
 465
 466
 467
 468
 469
 470
 471
 472
 473
 474
 475
 476
 477
 478
 479
 480
 481
 482
 483
 484
 485
 486
 487
 488
 489
 490
 491
 492
 493
 494
 495
 496
 497
 498
 499
 500
 501
 502
 503
 504
 505
 506
 507
 508
 509
 510
 511
 512
 513
 514
 515
 516
 517
 518
 519
 520
 521
 522
 523
 524
 525

ORG OPUTEIO-15
THIS SECTION DISTINGUISHES BETWEEN CASES WITH DIFFERENT VALUES OF THE LPAST
SIGNIFICANT FOUR BITS OF THE I/O COMMAND.

[illegible]

REGISTER NOTIFICATION: THE CLERK HAS BEEN ADVISED BY THE CLERK OF THE DISTRICT COURT OF THE DISTRICT OF COLUMBIA THAT THE FOLLOWING PERSONS HAVE BEEN ARRESTED FOR VIOLATION OF THE PROHIBITION AGAINST THE SALE OF FIREARMS TO MINORS:

[illegible]

WAC ...JP & ALU & ADDRESS ILLUCOP : CMD C207

... (1) THE INSTRUCTION IS RESTORED TO THE INSTRUCTION REGISTER.

• **Chlorophyll** is the green pigment in plants that captures light energy for photosynthesis.

(1) THE CONTENTS OF FPI (INSTRUCTION) ARE LOADED INTO THE INSTRUCTION REGISTER.
 ...0100 SETIMMR: MACF ...CONT & ALU & DRUS SREG & REG ...R1
 ;
 (4) THE CONTENTS OF GPI ARE INVERTED AND LOADED INTO ERIS
 MAC ...CONT & ALU ...R0E, ERIMOR, ERIMOR, 2A & FLS ...GRI, R15 & DRUS SALU
 ;

;(4) THE CONTENTS OF CRI ARE INVERTED AND LOADED INTO ER15

1 MAY 1981

AMDOS/29 ANDASH MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (ENSL)

PAGE 45

/ 6 CREG

MIMICINT

THIS SUBROUTINE MIMICS THE CONTENTS OF THE REGISTER ER15 IN THE 2914
MACHINE. THE MIMICING TAKES ACCOUNT OF WHETHER INTERRUPTS ARE
DISABLED OR NOT.

(2) A MASK 000000000000 (BIT 9 SET) IS LOADED INTO QREG.

0192 MIMICINT: MAC ...,CONT & ALU ..,DBUS.OREG.OR.OR.DZ & DBUS SDATA & DATINSRT D#64
/ 6 CKALU

(3) THE MACHINE STATE WORD IN ER4 IS AND'ED WITH THE MASK IN QREG
MAC ...,CONT & ALU ..,ABUS.MOP.AND.AND.DQ & REG R4

(4) JUMP TO ENABLINT TAKES PLACE IF ZLSB WAS NOT SET, INDICATING THAT INTERRUPTS
ARE ENABLED. ENABLINT PLACES THE MASK IN ER15 INTO THE MASK REGISTERS IN
THE 2914 CRIPS BUT ALWAYS SETS BIT 0 IN 2914 TO ZERO TO TAKE ACCOUNT OF THE FACT THAT
POWER DOWN CANNOT BE MASKED.

0194 MAC ,FALSE,ZLSP,CJP & ALU & BADDRESS ENABLINT

(5) SINCE INTERRUPTS ARE DISABLED, JUMP TAKES PLACE TO DISABINT WHICH LOADS
THE SPECIAL DISABLE MASK INTO THE 2914 CRIPS.

0195 MAC ...,JP & ALU & BADDRESS DISABINT

CLEAR INTERRUPT REQUESTS

(1) ALL MACHINE INTERRUPTS ARE CLEARED.

0196 CLRINTREQ: MAC ,FALSE,TZFO,CRIM & ALU & ENAUXC & MACINT CLRIM
/ 6 STATSHPT D#00000011000

ENABLE INTERRUPTS

ER15 CONTAINS THE INTERRUPT MASK. TO ENABLE INTERRUPTS, THE INTERRUPT MASK
FROM THIS REGISTER IS PLACED IN THE MASK REGISTERS OF THE 2914 CRIPS.
SINCE THE POWER DOWN INTERRUPT CANNOT BE MASKED, THIS BIT IS ALWAYS
CLEARED IN THE 2914 MASK DESPITE ITS VALUE IN ER15.

(1) A MASK 011111111111 IS PLACED IN QREG.

0197 ENABLINT: MAC ...,CONT & ALU ..,DBUS.OREG.OR.OR.DZ & DBUS SDATA & CKALU

1 MAY 1981

AMDOS/29 ANDASH MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (ENSL)

PAGE 46

/ 6 DATINSRT D#32767

(2) THE CONTENTS OF ER15 ARE AND'ED WITH THE QREG AND THE RESULT PLACED IN
THE MASK REGISTERS OF THE 2914 CRIPS.

0198 MAC ,FALSE,TZFO,CRIM & ALU ..,ABUS.MOP.AND.AND.DQ & REG R15
/ 6 DBUS SALU & ENAUXC & STATSHPT D#00000011000 & MACINT LDM

DISABLE INTERRUPTS

THE 2914 HAS THE FACILITY TO DISABLE ALL INTERRUPTS BUT THIS CANNOT BE USED
BECAUSE THE TWO HIGHEST PRIORITY ARE NOT TO BE DISABLED. THE TECHNIQUE
USED HERE IS THEREFORE TO LOAD A FALSE MASK INTO THE 2914, SO THAT ALL BUT
THE TWO HIGHEST PRIORITY ARE MASKED. CLEARED THE PROPER MASK IN ER15 SO THAT
IT CAN BE RESTORED WHEN IT IS REQUIRED TO ENABLE INTERRUPTS AGAIN.

(1) A MASK IS LOADED INTO THE QREG WITH ALL BITS SET TO ONE EXCEPT THE TWO
HIGHEST PRIORITY.

0199 DISABINT: MAC ...,CONT & ALU ..,DBUS.OREG.OR.OR.DZ & DBUS SDATA & DATINSRT D#16383
/ 6 CKALU

(3) THE DATA IN THE Q REG IS ORED WITH THE INTERRUPT MASK IN ER15 AND LOADED
INTO THE 2914 MASK REGISTERS.

019A MAC ,FALSE,TZFO,CRIM & ALU ..,ABUS.MOP.OR.OR.DQ & DBUS SALU & ENAUXC
/ 6 MACINT LDM & STATSHPT D#00000011000 & REG R15

RESET NORMAL POWER UP

THIS HAS NOT YET BEEN IMPLEMENTED

019B RESHMPU: MAC ...,JP & ALU & BADDRESS ILLEGOP

WRITE STATUS WORD

(1) THE INSTRUCTION IN ER1 IS LOADED INTO THE INSTRUCTION REGISTER.

[illegible]

DICG WTSUSD: NACF ...CONT & ALU & DUS SAEZ & RES ...AL
 ; (2) THE WORD IN CR1 IS WRITTEN INTO THE LEAST SIGNIFICANT 16 BITS OF THE
 ; MACHINE STATUS WORD.
 NACF ...CONT & ALU ...NUP,ORLOZA & REG ...DRI & ENACE & DUS SALU
 ...SUCCE ...MOLLA

[illegible]

1. The first part of the document is a title page. It contains the title "THE HISTORY OF THE UNITED STATES OF AMERICA" and the author "BY JAMES MADISON".

THIS STATION DISTINGUISHES BETWEEN TAIDS WITH DIFFERENT TONES ON THE LEAST

51-85461-58-15

[illegible]

0197 : TH I/O COMMAND.

[illegible]

AMPOS/20 AMDASH MICRO ASSEMBLY, V. 1.0
SUGGEST FOR 1750. 1MSL

START TIME: 000000

 IT IS REQUESTED THAT YOU ADVISE THE BUREAU OF THE RESULTS OF YOUR
 INVESTIGATION. THE RESULTS OF YOUR INVESTIGATION SHOULD BE
 FURNISHED TO THE BUREAU OF INVESTIGATION, WASHINGTON, D. C.
 20535, AND TO THE ATTORNEY GENERAL, WASHINGTON, D. C. 20540.
 YOUR COOPERATION IN THIS MATTER IS APPRECIATED.
 Sincerely,
 Special Agent in Charge

(1) A PASS DETECTOR WITH A 100% DETECTION RATE AND A 100% FALSE ALARM RATE.

1
 2
 3
 4
 5
 6
 7
 8
 9
 10
 11
 12
 13
 14
 15
 16
 17
 18
 19
 20
 21
 22
 23
 24
 25
 26
 27
 28
 29
 30
 31
 32
 33
 34
 35
 36
 37
 38
 39
 40
 41
 42
 43
 44
 45
 46
 47
 48
 49
 50
 51
 52
 53
 54
 55
 56
 57
 58
 59
 60
 61
 62
 63
 64
 65
 66
 67
 68
 69
 70
 71
 72
 73
 74
 75
 76
 77
 78
 79
 80
 81
 82
 83
 84
 85
 86
 87
 88
 89
 90
 91
 92
 93
 94
 95
 96
 97
 98
 99
 100
 101
 102
 103
 104
 105
 106
 107
 108
 109
 110
 111
 112
 113
 114
 115
 116
 117
 118
 119
 120
 121
 122
 123
 124
 125
 126
 127
 128
 129
 130
 131
 132
 133
 134
 135
 136
 137
 138
 139
 140
 141
 142
 143
 144
 145
 146
 147
 148
 149
 150
 151
 152
 153
 154
 155
 156
 157
 158
 159
 160
 161
 162
 163
 164
 165
 166
 167
 168
 169
 170
 171
 172
 173
 174
 175
 176
 177
 178
 179
 180
 181
 182
 183
 184
 185
 186
 187
 188
 189
 190
 191
 192
 193
 194
 195
 196
 197
 198
 199
 200
 201
 202
 203
 204
 205
 206
 207
 208
 209
 210
 211
 212
 213
 214
 215
 216
 217
 218
 219
 220
 221
 222
 223
 224
 225
 226
 227
 228
 229
 230
 231
 232
 233
 234
 235
 236
 237
 238
 239
 240
 241
 242
 243
 244
 245
 246
 247
 248
 249
 250
 251
 252
 253
 254
 255
 256
 257
 258
 259
 260
 261
 262
 263
 264
 265
 266
 267
 268
 269
 270
 271
 272
 273
 274
 275
 276
 277
 278
 279
 280
 281
 282
 283
 284
 285
 286
 287
 288
 289
 290
 291
 292
 293
 294
 295
 296
 297
 298
 299
 300
 301
 302
 303
 304
 305
 306
 307
 308
 309
 310
 311
 312
 313
 314
 315
 316
 317
 318
 319
 320
 321
 322
 323
 324
 325
 326
 327
 328
 329
 330
 331
 332
 333
 334
 335
 336
 337
 338
 339
 340
 341
 342
 343
 344
 345
 346
 347
 348
 349
 350
 351
 352
 353
 354
 355
 356
 357
 358
 359
 360
 361
 362
 363
 364
 365
 366
 367
 368
 369
 370
 371
 372
 373
 374
 375
 376
 377
 378
 379
 380
 381
 382
 383
 384
 385
 386
 387
 388
 389
 390
 391
 392
 393
 394
 395
 396
 397
 398
 399
 400
 401
 402
 403
 404
 405
 406
 407
 408
 409
 410
 411
 412
 413
 414
 415
 416
 417
 418
 419
 420
 421
 422
 423
 424
 425
 426
 427
 428
 429
 430
 431
 432
 433
 434
 435
 436
 437
 438
 439
 440
 441
 442
 443
 444
 445
 446
 447
 448
 449
 450
 451
 452
 453
 454
 455
 456
 457
 458
 459
 460
 461
 462
 463
 464
 465
 466
 467
 468
 469
 470
 471
 472
 473
 474
 475
 476
 477
 478
 479
 480
 481
 482
 483
 484
 485
 486
 487
 488
 489
 490
 491
 492
 493
 494
 495
 496
 497
 498
 499
 500
 501
 502
 503
 504
 505
 506
 507
 508
 509
 510
 511
 512
 513
 514
 515
 516
 517
 518
 519
 520
 521
 522
 523
 524
 525

[illegible]

DATE

TO THE ONE IS A LITER A WALK WITH PIT & SET AND ALL ELSE CLEAR.
TIME: NAC 1000.00Z
FROM: NAC 1000.00Z

THE CONTENTS OF THE SAFE WERE INDEXED WITH THE CONTENTS OF THE QREG
AND THE CONTENTS OF THE SAFE WERE INDEXED WITH THE CONTENTS OF THE QREG
AND THE CONTENTS OF THE SAFE WERE INDEXED WITH THE CONTENTS OF THE QREG

21P10
21P11
21P12
21P13
21P14
21P15
21P16
21P17
21P18
21P19
21P20
21P21
21P22
21P23
21P24
21P25
21P26
21P27
21P28
21P29
21P30
21P31
21P32
21P33
21P34
21P35
21P36
21P37
21P38
21P39
21P40
21P41
21P42
21P43
21P44
21P45
21P46
21P47
21P48
21P49
21P50
21P51
21P52
21P53
21P54
21P55
21P56
21P57
21P58
21P59
21P60
21P61
21P62
21P63
21P64
21P65
21P66
21P67
21P68
21P69
21P70
21P71
21P72
21P73
21P74
21P75
21P76
21P77
21P78
21P79
21P80
21P81
21P82
21P83
21P84
21P85
21P86
21P87
21P88
21P89
21P90
21P91
21P92
21P93
21P94
21P95
21P96
21P97
21P98
21P99
21P100
21P101
21P102
21P103
21P104
21P105
21P106
21P107
21P108
21P109
21P110
21P111
21P112
21P113
21P114
21P115
21P116
21P117
21P118
21P119
21P120
21P121
21P122
21P123
21P124
21P125
21P126
21P127
21P128
21P129
21P130
21P131
21P132
21P133
21P134
21P135
21P136
21P137
21P138
21P139
21P140
21P141
21P142
21P143
21P144
21P145
21P146
21P147
21P148
21P149
21P150
21P151
21P152
21P153
21P154
21P155
21P156
21P157
21P158
21P159
21P160
21P161
21P162
21P163
21P164
21P165
21P166
21P167
21P168
21P169
21P170
21P171
21P172
21P173
21P174
21P175
21P176
21P177
21P178
21P179
21P180
21P181
21P182
21P183
21P184
21P185
21P186
21P187
21P188
21P189
21P190
21P191
21P192
21P193
21P194
21P195
21P196
21P197
21P198
21P199
21P200
21P201
21P202
21P203
21P204
21P205
21P206
21P207
21P208
21P209
21P210
21P211
21P212
21P213
21P214
21P215
21P216
21P217
21P218
21P219
21P220
21P221
21P222
21P223
21P224
21P225
21P226
21P227
21P228
21P229
21P230
21P231
21P232
21P233
21P234
21P235
21P236
21P237
21P238
21P239
21P240
21P241
21P242
21P243
21P244
21P245
21P246
21P247
21P248
21P249
21P250
21P251
21P252
21P253
21P254
21P255
21P256
21P257
21P258
21P259
21P260
21P261
21P262
21P263
21P264
21P265
21P266
21P267
21P268
21P269
21P270
21P271
21P272
21P273
21P274
21P275
21P276
21P277
21P278
21P279
21P280
21P281
21P282
21P283
21P284
21P285
21P286
21P287
21P288
21P289
21P290
21P291
21P292
21P293
21P294
21P295
21P296
21P297
21P298
21P299
21P300
21P301
21P302
21P303
21P304
21P305
21P306
21P307
21P308
21P309
21P310
21P311
21P312
21P313
21P314
21P315
21P316
21P317
21P318
21P319
21P320
21P321
21P322
21P323
21P324
21P325
21P326
21P327
21P328
21P329
21P330
21P331
21P332
21P333
21P334
21P335
21P336
21P337
21P338
21P339
21P340
21P341
21P342
21P343
21P344
21P345
21P346
21P347
21P348
21P349
21P350
21P351
21P352
21P353
21P354
21P355
21P356
21P357
21P358
21P359
21P360
21P361
21P362
21P363
21P364
21P365
21P366
21P367
21P368
21P369
21P370
21P371
21P372
21P373
21P374
21P375
21P376
21P377
21P378
21P379
21P380
21P381
21P382
21P383
21P384
21P385
21P386
21P387
21P388
21P389
21P390
21P391
21P392
21P393
21P394
21P395
21P396
21P397
21P398
21P399
21P400
21P401
21P402
21P403
21P404
21P405
21P406
21P407
21P408
21P409
21P410
21P411
21P412
21P413
21P414
21P415
21P416
21P417
21P418
21P419
21P420
21P421
21P422
21P423
21P424
21P425
21P426
21P427
21P428
21P429
21P430
21P431
21P432
21P433
21P434
21P435
21P436
21P437
21P438
21P439
21P440
21P441
21P442
21P443
21P444
21P445
21P446
21P447
21P448
21P449
21P450
21P451
21P452
21P453
21P454
21P455
21P456
21P457
21P458
21P459
21P460
21P461
21P462
21P463
21P464
21P465
21P466
21P467
21P468
21P469
21P470
21P471
21P472
21P473
21P474
21P475
21P476
21P477
21P478
21P479
21P480
21P481
21P482
21P483
21P484
21P485
21P486
21P487
21P488
21P489
21P490
21P491
21P492
21P493
21P494
21P495
21P496
21P497
21P498
21P499
21P500
21P501
21P502
21P503
21P504
21P505
21P506
21P507
21P508
21P509
21P510
21P511
21P512
21P513
21P514
21P515
21P516
21P517
21P518
21P519
21P520
21P521
21P522
21P523
21P524
21P525
21P526
21P527
21P528
21P529
21P530
21P531
21P532

UTYUT 71-19 A

11) THE INSTRUCTION REGISTER ARE LOADED WITH THE INSTRUCTION IN ERL.
SINCE SUB TAPES PLACE TO SETTING.

```

JUMP SUB TAKES PLACE TO SERVICE.
MACF, PULSE, ZERO, CTS & ACD & CRUS SREG & REG .....RI
ADDRESS S-RTIME
DIR4 OPTIMA:

```

...CONTENTS OF CASE ARE LOADED INTO TIMER A

1. RETURN PACIFIC PLACE
 2. RETURN PACIFIC PLACE
 3. RETURN PACIFIC PLACE
 4. RETURN PACIFIC PLACE
 5. RETURN PACIFIC PLACE
 6. RETURN PACIFIC PLACE
 7. RETURN PACIFIC PLACE
 8. RETURN PACIFIC PLACE
 9. RETURN PACIFIC PLACE
 10. RETURN PACIFIC PLACE
 11. RETURN PACIFIC PLACE
 12. RETURN PACIFIC PLACE
 13. RETURN PACIFIC PLACE
 14. RETURN PACIFIC PLACE
 15. RETURN PACIFIC PLACE
 16. RETURN PACIFIC PLACE
 17. RETURN PACIFIC PLACE
 18. RETURN PACIFIC PLACE
 19. RETURN PACIFIC PLACE
 20. RETURN PACIFIC PLACE
 21. RETURN PACIFIC PLACE
 22. RETURN PACIFIC PLACE
 23. RETURN PACIFIC PLACE
 24. RETURN PACIFIC PLACE
 25. RETURN PACIFIC PLACE
 26. RETURN PACIFIC PLACE
 27. RETURN PACIFIC PLACE
 28. RETURN PACIFIC PLACE
 29. RETURN PACIFIC PLACE
 30. RETURN PACIFIC PLACE
 31. RETURN PACIFIC PLACE
 32. RETURN PACIFIC PLACE
 33. RETURN PACIFIC PLACE
 34. RETURN PACIFIC PLACE
 35. RETURN PACIFIC PLACE
 36. RETURN PACIFIC PLACE
 37. RETURN PACIFIC PLACE
 38. RETURN PACIFIC PLACE
 39. RETURN PACIFIC PLACE
 40. RETURN PACIFIC PLACE
 41. RETURN PACIFIC PLACE
 42. RETURN PACIFIC PLACE
 43. RETURN PACIFIC PLACE
 44. RETURN PACIFIC PLACE
 45. RETURN PACIFIC PLACE
 46. RETURN PACIFIC PLACE
 47. RETURN PACIFIC PLACE
 48. RETURN PACIFIC PLACE
 49. RETURN PACIFIC PLACE
 50. RETURN PACIFIC PLACE
 51. RETURN PACIFIC PLACE
 52. RETURN PACIFIC PLACE
 53. RETURN PACIFIC PLACE
 54. RETURN PACIFIC PLACE
 55. RETURN PACIFIC PLACE
 56. RETURN PACIFIC PLACE
 57. RETURN PACIFIC PLACE
 58. RETURN PACIFIC PLACE
 59. RETURN PACIFIC PLACE
 60. RETURN PACIFIC PLACE
 61. RETURN PACIFIC PLACE
 62. RETURN PACIFIC PLACE
 63. RETURN PACIFIC PLACE
 64. RETURN PACIFIC PLACE
 65. RETURN PACIFIC PLACE
 66. RETURN PACIFIC PLACE
 67. RETURN PACIFIC PLACE
 68. RETURN PACIFIC PLACE
 69. RETURN PACIFIC PLACE
 70. RETURN PACIFIC PLACE
 71. RETURN PACIFIC PLACE
 72. RETURN PACIFIC PLACE
 73. RETURN PACIFIC PLACE
 74. RETURN PACIFIC PLACE
 75. RETURN PACIFIC PLACE
 76. RETURN PACIFIC PLACE
 77. RETURN PACIFIC PLACE
 78. RETURN PACIFIC PLACE
 79. RETURN PACIFIC PLACE
 80. RETURN PACIFIC PLACE
 81. RETURN PACIFIC PLACE
 82. RETURN PACIFIC PLACE
 83. RETURN PACIFIC PLACE
 84. RETURN PACIFIC PLACE
 85. RETURN PACIFIC PLACE
 86. RETURN PACIFIC PLACE
 87. RETURN PACIFIC PLACE
 88. RETURN PACIFIC PLACE
 89. RETURN PACIFIC PLACE
 90. RETURN PACIFIC PLACE
 91. RETURN PACIFIC PLACE
 92. RETURN PACIFIC PLACE
 93. RETURN PACIFIC PLACE
 94. RETURN PACIFIC PLACE
 95. RETURN PACIFIC PLACE
 96. RETURN PACIFIC PLACE
 97. RETURN PACIFIC PLACE
 98. RETURN PACIFIC PLACE
 99. RETURN PACIFIC PLACE
 100. RETURN PACIFIC PLACE

1 MAY 1981

PAGE 40

AMDS/20 AMDASH MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EMSL)

START TIMER B

BIT 5 OF THE STATE WORD (COUNTING FROM THE MS END) IS ALLOCATED TO
INDICATING WHETHER TIMER B IS RUNNING OR NOT.

01B6 SETTIME: MAC ,FALSE,TZERO,CRTN & ALU ...AND5,NOP,OR,OR,DQ & CIREG & DBUS SALU
/ 6 DAYINSTR D1024

01B7 / 5 (2) THE STATE WORD IS ORED WITH THE CONTENTS OF THE REGISTER AND LOADED
INTO ERA AND THE STATE LATCHES.

01B8 / 5 RETURN TAKES PLACE
MAC ,FALSE,TZERO,CRTN & ALU ...AND5,NOP,OR,OR,DQ & CIREG & DBUS SALU
/ 6 ENAUIC & REC PA & AVICE B00010

BALT TIMER B

01B9 / 6 (1) THE REGISTER IS LOADED WITH A ZERO IN BIT 5 AND A ONE ELSEWHERE.
JUMP SUB TAKES PLACE TO STRTIME.

01BA / 6 (2) THE STATE WORD FROM ERA IS AND'ED WITH THE CONTENTS OF THE REGISTER
AND LOADED INTO ERA AND THE STATE LATCHES.
MAC ,FALSE,TZERO,CRTN & ALU ...AND5,NOP,AND,AND,DQ & CIREG & DBUS SALU
/ 6 REC RA & ENAUIC & AVICE B00010

OUTPUT TIMER B

01BB / 6 (1) THE INSTRUCTION REGISTER IS LOADED WITH THE INSTRUCTION IN ERA.
JUMP SUB TAKES PLACE TO STRTIME.

01BC / 6 (2) THE CONTENTS OF CRI ARE LOADED INTO THE TIMER B.
RETURN TAKES PLACE.

01BD / 6 MAC ,FALSE,TZERO,CRTN & ALU ...NOP,OR,OR,2A & REC ...CRI & DBUS SALU

1 MAY 1981

PAGE 50

AMDS/20 AMDASH MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EMSL)

/ 5 ENAUIC & STATSEPT B000010000000

CLEAR FAULT REGISTER

01BE / 5 (1) THE FAULT REGISTER IS CLEARED AND RETURN TAKES PLACE
CIRCTNG: MAC ,FALSE,TZERO,CRTN & ALU & DBUS SDATA & DAYINSTR D00 & REC RS
/ 6 CRRFC

OUTPUT TO DATA DISPLAY FIELD

01BF / 6 (1) THE INSTRUCTION IS RELOADED TO THE INSTRUCTION REGISTER
OPDIDSP: MAC ,FALSE,TZERO,CRTN & ALU & DBUS SREG & REC ...RI

01BG / 6 (2) THE DATA IN CRI IS LOADED INTO THE DATA DISPLAY LATCHES.
RETURN TAKES PLACE

01BH / 6 MAC ,FALSE,TZERO,CRTN & ALU ...NOP,OR,OR,2A & REC ...CRI & DBUS SALU
/ 6 ENAUIC & AVICE B00001

PROCESSOR AND AUXILIARY REGISTER CONTROL (COMMAND GROUP XIII)

OPC STRTIME+15

01BI / 6 (1) JUMP TAKES PLACE TO BRCB00P1.

01BJ / 6 OPUTE14: MAC ,FALSE,TZERO,CJS & ALU & DBUS SMOKE & ADDRESS BRCB00P1

01BK / 6 MAC ...JP & ALU & ADDRESS ILLEGOP ; CMD 4000

01CL / 6 MAC ...JP & ALU & ADDRESS ILLEGOP ; CMD 4001

01CM / 6 MAC ...JP & ALU & ADDRESS ILLEGOP ; CMD 4002

01CN / 6 MAC ...JP & ALU & ADDRESS ILLEGOP ; CMD 4003

01CO / 6 MAC ...JP & ALU & ADDRESS ILLEGOP ; CMD 4004

01CP / 6 MAC ...JP & ALU & ADDRESS ILLEGOP ; CMD 4005

01CQ / 6 MAC ...JP & ALU & ADDRESS ILLEGOP ; CMD 4006

01CR / 6 MAC ...JP & ALU & ADDRESS ILLEGOP ; CMD 4007

01CS / 6 MAC ...JP & ALU & ADDRESS ILLEGOP ; CMD 4008

01CT / 6 MAC ...JP & ALU & ADDRESS ILLEGOP ; CMD 4009

01CU / 6 MAC ...JP & ALU & ADDRESS ILLEGOP ; CMD 400A

01CV / 6 MAC ...JP & ALU & ADDRESS ILLEGOP ; CMD 400B

2401 52

```

; NO - OPERATION AND BREAKPOINT
; *****
;
; @CODEFF:: MAC ...CONT & ALU ...ABDS.NOP.OR.OR.DZ & REP ...X.
;
; (2) RETURN THIS PLACE IF LSP WAS SET BY THE LAST INSTRUCTION, THE CONTENTS
; OF R1 ARE INSTANTLY CACHED THROUGH THE ALU AGAIN.
; NO TRUE ALU CATCH & ALU ...ABDS.NOP.INSTR.DZ & REP ...R1
; @D6

```


Page 5

AMLOS/20 ANDAM MICRO ASSISTANT, V.1.0
 EVALUATION SEQUENCES FOR SINGLE LENGTH SUBJECT FOR 1250 (EMSL)

THE FOLLOWING INFORMATION IS FOR THE ROUTINE OCCURS.

5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000 1001 1002 1003 1004 1005 1006 1007 1008 1009 1010 1011 1012 1013 1014 1015 1016 1017 1018 1019 1020 1021 1022 1023 1024 1025 1026 1027 1028 1029 1030 1031 1032 1033 1034 1035 1036 1037 1038 1039 1040 1041 1042

14. THIS INFORMATION MAY BE USED FOR DIRECT-MAILED ADDRESSING.
 15. ADDRESS: 10000 13TH AVE. N.E. BOX 10000
 EDWARDS, WY 83503

A JUMP IS MADE TO THE
PAGE OF THE INDEX

104 THIS IS THE ADDRESS FOR THE MAIL ADDRESS.
A JUNE 1964 COPY OF THE JUNE 1964 PAPER.
DPC0411: MAY 1964, P. 104
S BUREAU OF THE
S ALL INFORMATION

2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841

SINGLE PRECISION INTER-SUBJECT SURTHACT

•••••

1) THIS IS THE ENTRY INFORMATION TO REGISTER ADDRESS MODE.
THE COMPANIES OF THE SIGHTED SUBJECTS ARE SUBTRACTED.
THE MACHINE AND THE DATA ARE LOADED DIRECTLY FROM
THE INDEXING.

THE CONTENTS OF THE SPECIALTY REGISTERS ARE SUBTRACTED.
THE MACHINE AND MICROSTATUS REGISTERS ARE LOADED DIRECTLY FROM
IZIC, IN IP
CARRY IN TO THE ALU IS SET TO ONE
CARRY TO THE ADDRESS REGISTER IS CARRY OUT OF THE ALU

CARRY TO THE APPROPRIATE SPACES BUT IS CARRY OUT OF THE ALU
RECEIVED: MAR 1968

23

[illegible]

4390

[illegible]

100

LEADERSHIP IN THE 21ST CENTURY

* * * * *

A FIRST PRINT FOR REGISTER TO RECEIVE ADDRESSING MODEL
FOR DATA IN 1980 IS AVAILABLE TO \$300

8422	OPCDEIGI:	NAC	...B
/	/	ALG	...P.P.B.24
/	/	RFG	...R.B.
/	/	CX9E	
/	/	DBU	SAB
/	/	PAC	SAB

```

;C2 ENTRY POINT FOR ISP ADDRESSING
;*****
; JUMP SUB TO ISP OPNARD PATCH ROUTINE
;*****
%023 SPECIES2:: MAC %A2,ITER,%C23
      BARRERES CATSP
      AINCE CLOS SMOKE

```

4242
:(3) JUMP TO START OF MULTIPLE SEQUENCE
MAC 00000000
BAC 00000000
A00 00000000

062B
; NAC, YAL:

1 MAY 1981

PAGE 10

AMDOS/20 LMDASH MICRO ASSEMBLER, V1.0
 EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (BMSL)

8420 / 5 ADDRESS ASSEMBLY
 / 5 ALU & DRUS SMOKE
 8420 / 5 MAC ...COUNT
 / 5 ALU ...OP, OR, OR, ZA
 / 5 REG ...DR1, DR1
 / 5 DRUS SMOKE

8420 / 5 JUMP TO OVERFLOW TAKES PLACE IF THE MS BIT OF DR1 IS NOT ZERO.
 / 5 SPINMS: MAC ...FALSE, TRUE
 / 5 ADDRESS OVERFLOW
 / 5 ALU ...OP, OR, OR, ZA
 / 5 REG ...DR1, DR1
 / 5 DRUS SMOKE

8420 / 5 JUMP TO OVERFLOW TAKES PLACE IF THE MS BIT OF DR1 IS NOT ZERO.
 / 5 SPINMS: MAC ...FALSE, TRUE
 / 5 ADDRESS OVERFLOW
 / 5 ALU ...OP, OR, OR, ZA
 / 5 REG ...DR1, DR1
 / 5 DRUS SMOKE

8420 / 5 THE MS WORD OF THE PRODUCT IN THE REG IS MAILED TO LEAVE ONLY THE MS
 BIT.

8427 / 5 MAC ...COUNT
 / 5 ALU ...DRUS, DRMP, AND, AND, DR
 / 5 DRUS SMOKE
 / 5 CARINPT DR5576H

8430 / 5 JUMP TO OVERFLOW TAKES PLACE IF THE MS BIT IS NOT ZERO. ERS IS PASSED
 THROUGH THE ALU SO THAT THE TEST CONDITION MSB-B WILL BE LOADED ACCORDING
 TO THE VALUE OF THE MS BIT OF DRH. (THE REST OF ERS IS ZERO).

8430 / 5 MAC ...FALSE, TRUE, B, CJP
 / 5 ADDRESS OVERFLOW
 / 5 ALU ...DRUS, DRMP, OR, OR, D1
 / 5 REG ...DR8
 / 5 DRUS SMOKE

8431 / 5 JUMP TAKES PLACE TO SPINMS IF THE MS BIT OF DRH WAS ZERO.
 / 5 MAC ...TRUE, TRUE, B, CJP
 / 5 ALU & DRUS SMOKE
 / 5 ADDRESS SPINMS

1 MAY 1981

PAGE 11

AMDOS/20 LMDASH MICRO ASSEMBLER, V1.0
 EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (BMSL)

8430 / 5 THE MS WORD OF THE PRODUCT IN THE REG IS MAILED TO LEAVE ONLY THE MS
 ACTION WHEN IT HAS BEEN DETERMINED THAT THE PRODUCT MUST BE NEGATIVE.

8430 / 5 MAC ...COUNT
 / 5 ALU ...DRUS, DRMP, AND, AND, DR
 / 5 REG ...DR1, DR1
 / 5 DRUS SMOKE
 8430 / 5 MAC ...TRUE, TRUE, B, CJP
 / 5 ADDRESS OVERFLOW
 / 5 ALU ...DRUS, DRMP, OR, OR, D1
 / 5 REG ...DR8
 / 5 DRUS SMOKE

8430 / 5 RETURN TAKES PLACE
 / 5 MAC ...TRUE, TRUE, B, CJP
 / 5 ALU & DRUS SMOKE

8430 / 5 IN THE CASE WHEN THE COUNT OF THE PRODUCT HAS BEEN DETERMINED TO
 BE POSITIVE, THE MS BIT OF DR1 AND DRH IS LOADED INTO DR1. THE STATUS
 REGISTER LOADED AND A CARRY AND MAY BE OCCURS.

8430 / 5 SPINMS: MAC ...FALSE, TRUE
 / 5 ADDRESS OVERFLOW
 / 5 ALU ...DRUS, DRMP, OR, OR, D1
 / 5 REG ...DR8
 / 5 DRUS SMOKE
 8430 / 5 MAC ...TRUE, TRUE, B, CJP
 / 5 ADDRESS OVERFLOW
 / 5 ALU ...DRUS, DRMP, OR, OR, D1
 / 5 REG ...DR8
 / 5 DRUS SMOKE

SINGLE PRECISION INTER DIVIDE

ENTRY POINT FOR REGISTER TO REGISTER ADDRESSING MODE

8435 / 5 THE DATA IN DR2 IS TRANSFERRED TO ERS
 / 5 MAC ...TRUE, TRUE, B, CJP
 / 5 ALU ...DRUS, DRMP, OR, OR, ZA
 / 5 REG ...DR8
 / 5 DRUS SMOKE

8435 / 5 ENTRY POINT FOR ISP ADDRESSING MODE

10

10

CO. WIDISH MICRO A
ON SEQUENCES R R

(10) A TEST IS
 P/DG
 NAC
 ALU
 DEL
 CRASH

(1) JUMP TO SPIN
NAC 1400

3333

(12) JUMP TO CTR. = 110
MAC TRAP = 110
RACDR = 110
ALU 5 110

13) A JUMP IS MADE TO THE ADDRESS OF THE FIRST QUADRA T DIV

14) THE DIVISION
5 MAC
5 ALU
5 RES
5 CALCU & CTR
5 DATING

14A) THE DIVISION OF
NAC "COAST"
ALU "ABC"
REC "GREG"
CALU "DESS"

15) THE CONSTANT α OF THE MICROPROGRAM IS NOT LOADED INTO THE COUNTER IS

44-16

[illegible][illegible][illegible][illegible][illegible][illegible][illegible]

100

1 MAY 1981

PAGE 14

1 MAY 1981

PAGE 15

AMDOS/20 ANDASH MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EMSL)

```

/ 6 DATMSRT D#1
/ 6
;(15) QREG AND GR1 ARE UNSHIFTED, & JOES INTO THE MS BIT OF DOTS.
/ 6
0444 / 6 MAC ...CONT
/ 6 ALU B#0,B#11,RAMQU,OR,OR,ZA
/ 6 REC B#01,GR1,GR1
/ 6 CIALU & D#US SNOKE
/ 6 STATSHFT B#000100010000
/ 6
;(16) JUMP BACK TO (15) TAKES PLACE IF N IN THE MICROSTATUS REGISTER IS NOT
/ 6 SET.
/ 6
0445 / 6 MAC TRUE,MUL LOOP
/ 6 ALU & D#US SNOKE
/ 6 STATSHFT B#000000011110
/ 6
;(16A) THE DIVISOR AND THE QUOTIENT MASK ARE DOWNSHIFTED TO COMPENSATE
/ 6 FOR THE FACT THAT ONE EXTRA SHIFT HAS BEEN DONE.
/ 6
0446 / 6 MAC ...CONT
/ 6 ALU ...RAMQU,OR,OR,ZA
/ 6 REC ...GR1,GR1
/ 6 CIALU & D#US SNOKE
/ 6 STATSHFT B#000100000000
/ 6
;(16B) THE DIVISOR AND QUOTIENT MASK ARE DOWNSHIFTED AGAIN SO THAT
/ 6 THE FIRST NON ZERO BIT IS ALIGNED TO BIT POS. 14.
/ 6
0447 / 6 MAC ...CONT
/ 6 ALU ...RAMQU,OR,OR,ZA
/ 6 REC ...GR1,GR1
/ 6 CIALU & D#US SNOKE
/ 6 STATSHFT B#001100000000
/ 6
;(17) THE DIVISOR IS SUBTRACTED FROM THE REMAINDER.
/ 6
0448 / 6 MAC ...CONT
/ 6 ALU ...RAMP,SUBR,SUBR,AB
/ 6 REC B#01,GR1,GR1
/ 6 CIALU & D#US SNOKE
/ 6 STATSHFT B#010000000000
/ 6
;(18) DRS SIGN BIT OF THE REMAINDER IS MASKED OFF.
/ 6
0449 / 6 MAC ...CONT
/ 6 ALU ...D#US,NOP,AND,AND,DA
/ 6 REC B#01,B#01,GR1,GR1
/ 6 D#US SDA7 & DATMSRT D#32768
/ 6
;(19) JUP TAKES PLACE IF REMAINDER IS POSITIVE
/ 6
044A / 6 MAC TRUE,ZMSB,CJP

```

AMDOS/20 ANDASH MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EMSL)

```

/ 6 ADDRESS,SPID4
/ 6 ALU & D#US SNOKE
/ 6
;(20) DIVISOR IS ADDED TO REMAINDER
/ 6
044B / 6 MAC ...CONT
/ 6 ALU ...RAMP,ADD,ADD,AB
/ 6 REC B#01,GR1,GR1
/ 6 CIALU & D#US SNOKE
/ 6 STATSHFT B#000000000000
/ 6
;(21) THE DIVISOR AND THE QUOTIENT MASK ARE SHIFTED DOWN.
/ 6
044C / 6 MAC ...CONT
/ 6 ALU ...RAMQU,OR,OR,ZA
/ 6 REC ...GR1,GR1
/ 6 CIALU & D#US SNOKE
/ 6 CARRYSEL COLSB
/ 6 STATSHFT B#000000000000
/ 6
;(21) JUMP TO SPID2 TAKES PLACE IF CARRY WAS NOT SET. IE BIT IN QUOTIENT MASK
/ 6 HAS NOT BEEN SHIFTED OUT OF GR1.
/ 6
044D / 6 MAC FALSE,CARR,CJP
/ 6 ADDRESS SPID2
/ 6 ALU ...D#US,NOP,OR,OR,DZ
/ 6 REC ...R#
/ 6 D#US SNOKE
/ 6
;(22) JUMP TO SPID5 TAKES PLACE IF THE MS BIT WAS SET INDICATING A NEGATIVE
/ 6 RESULT. MAC ,FALSE,ZMSB,CJP & ALU & D#US SNOKE & ADDRESS SPID5
/ 6
044E / 6 MAC ,FALSE,ZMSB,CJP
/ 6
;(22) THE QUOTIENT IN ERG IS LOADED INTO GR1. RETURN TAKES PLACE IF THE
/ 6 QUOTIENT IS TO BE POSITIVE.
/ 6
044F / 6 MAC FALSE,TZPO,CRTN
/ 6 ALU B#1,B#0,ABUS,RAMP,OR,OR,DZ
/ 6 REC B#01,GR1,GR1,R#
/ 6 CIALU & D#US SNOKE
/ 6 STATSHFT B#000000010000
/ 6 CARRYSEL CZERO
/ 6
;(23) A ONE IS PLACED IN THE APPROPRIATE POSITION IN THE QUOTIENT USING THE
/ 6 QUOTIENT MASK IN QREG.
/ 6
0450 / 6 MAC ...CONT
/ 6 ALU ...ABUS,NOP,OR,OR,DQ
/ 6 REC ...R#
/ 6 CERE & D#US SALU
/ 6

```

1 MAY 1981

PAGE 16

PAGE 17

ANDOS/29 AMDASH MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EMSL)

```

9451 / 5 (26) THE DIVISOR AND THE QUOTIENT MASE ARE SHIFTED DOWN.
      / 5 MAC ...CONT
      / 5 ALU ...RAMPD,OR,OR,2A
      / 5 REG ...GRI,GRI
      / 5 CIALU & DBUS SMOKE
      / 5 CARRISEL CQLSB
      / 5 STATSHFT B#010000000000

```

```

9452 / 5 (25) JUMP TO SPID2 TAKES PLACE IF THE QUOTIENT
      / 5 MASE HAS NOT BEEN SHIFTED OUT OF THE GRI.
      / 5 MAC ...FALSE,CARR,CJP
      / 5 ADDRESS SPID2
      / 5 ALU ...RAMPD,OR,OR,DZ
      / 5 REG ...R#B
      / 5 DBUS SMOKE

```

```

9453 / 5 (25A) JUMP TO SPID5 TAKES PLACE IF THE MS BIT WAS SET INDICATING A NEGATIVE
      / 5 RESULT.
      / 5 MAC ...FALSE,ZMBR,CJP
      / 5 ALU & DBUS SMOKE & ADDRESS SPID5

```

```

9454 / 5 (26) THE QUOTIENT IS PLACED IN GRI, THE MACHINE STATUS REGISTER IS
      / 5 LOADED AND RETURN TAKES PLACE.
      / 5 MAC ...FALSE,TZERO,CRTN
      / 5 ALU B#1000,ANUS,REMP,OR,OR,DZ
      / 5 REG ...GRI,GRI,R#B
      / 5 CIALU & DBUS SMOKE
      / 5 STATSHFT B#000000010000

```

```

9455 / 5 (26A) THE REMAINDER IN GRI + 1 IS NEGATED.
      / 5 MAC ...CONT
      / 5 ALU ...RAMPD,OR,OR,2A
      / 5 REG ...RAMPD,OR,OR,2A
      / 5 CARRISEL CQLSB
      / 5 CIALU & DBUS SMOKE
      / 5 STATSHFT B#010000000000

```

```

9456 / 5 (26B) THE QUOTIENT IS NEGATED AND RETURN TAKES PLACE.
      / 5 MAC ...FALSE,TZERO,CRTN
      / 5 ALU B#1,ANUS,REMP,OR,OR,DZ
      / 5 CIALU & DBUS SMOKE

```

ANDOS/29 AMDASH MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EMSL)

```

/ 5 REG ...GRI,GRI,R#B
/ 5 STATSHFT B#010000010000
/ 5 CARRISEL CZERO

```

SINGLE PRECISION INTEGER COMPARE

ENTRY POINT FOR REGISTER TO REGISTER ADDRESSING

THE CONTENTS OF REGISTER GR2 ARE SUBTRACTED FROM THE CONTENTS
REGISTER GR1. THE MACHINE STATUS REGISTER IS SET BASED ON THE
RESULT WHICH IS NOT LOADED INTO ANY REGISTER. CARRY IN TO THE MACHINE
STATUS REGISTER IS SET TO ZERO.

```

0457 / 5 OP CODE Y1: MAC ...FALSE,TZERO,CRTN
      / 5 ALU ...R#B,ANUS,SUBS,SUBS,AB
      / 5 REG ...GR2,GRI
      / 5 DBUS SMOKE
      / 5 STATSHFT B#010000010000
      / 5 CARRISEL CZERO

```

ENTRY POINT FOR BASE RELATIVE ADDRESSING

```

0458 / 5 OP CODE Y2: MAC ...FALSE,TZERO,CJS
      / 5 ALU & DBUS SMOKE
      / 5 BADDRESS OFB1

```

```

0459 / 5 MAC ...FALSE,TZERO,CRTN
      / 5 ALU ...R#B,ANUS,ANUS,ANUS,SUBA,SUBA,DA
      / 5 REG ...R#10,RZERO,R5
      / 5 DBUS SMOKE
      / 5 CARRISEL CZERO & STATSHFT B#010000010000

```

ENTRY POINT FOR BASE RELATIVE INDEXED ADDRESSING

```

045A / 5 JUMP SUB TO OPERAND FETCH ROUTINE
      / 5 OP CODE Y0: MAC ...FALSE,TZERO,CJS
      / 5 ALU & DBUS SMOKE & BADDRESS OFB1

```

1 MAY 1981 PAGE 18

AMDOS/29 AMDASM MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EMSL)

0453 : DERIVED OP IN R15 IS SUBTRACTED FROM CONTENTS OF R12
/ 6
MAC ,FALSE,TZERO,CRTN
ALU ,R15,ADUS,MOP,SUBR,SUBR,DA
REG ,R15,TZERO,R15
DBUS SNOKE & CARRISEL CZERO
STATSHT B010000010000
/ 6
ENTRY POINT FOR ISP ADDRESSING

045C : MAC ,FALSE,TZERO,CJS
/ 6
ALU & DBUS SNOKE & ADDRESS OFISP

045D : MAC ,FALSE,TZERO,CRTN
/ 6
ALU ,R15,ADUS,MOP,SUBR,SUBR,DA
REG ,R15,TZERO,R15
DBUS SNOKE & CARRISEL CZERO & STATSHT B010000010000
/ 6
ENTRY POINT FOR ISM ADDRESSING

045E : MAC ,FALSE,TZERO,CJS
/ 6
ALU & DBUS SNOKE & ADDRESS OFISM

045F : MAC ,FALSE,TZERO,CRTN
/ 6
ALU ,R15,ADUS,MOP,SUBR,SUBR,DA
REG ,R15,TZERO,R15
DBUS SNOKE & CARRISEL CZERO
STATSHT B010000010000
/ 6
ENTRY POINT FOR DIRECT OR DIRECT IMPLIED ADDRESSING

0460 : MAC ,FALSE,TZERO,CJS
/ 6
ALU & DBUS SNOKE & ADDRESS OFD11

0461 : MAC ,FALSE,TZERO,CRTN
/ 6
ALU ,R15,ADUS,MOP,SUBR,SUBR,DA
REG ,R15,TZERO,R15
DBUS SNOKE & CARRISEL CZERO
STATSHT B010000010000
/ 6
ENTRY POINT FOR IMMEDIATE ADDRESSING

1 MAY 1981 PAGE 19

AMDOS/29 AMDASM MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EMSL)

0462 : MAC ,FALSE,TZERO,CJS
/ 6
ALU & DBUS SNOKE & ADDRESS OFIM

0463 : MAC ,FALSE,TZERO,CRTN
/ 6
ALU ,R15,ADUS,MOP,SUBR,SUBR,DA
REG ,R15,TZERO,R15
DBUS SNOKE & CARRISEL CZERO
STATSHT B010000010000
/ 6
COMPARE BETWEEN LIMITS

ENTRY POINT FOR DIRECT OR DIRECT IMPLIED ADDRESSING

0464 : MAC ,FALSE,TZERO,CJS
/ 6
ALU & DBUS SNOKE & ADDRESS OFD12

0465 : MAC ,FALSE,TZERO,CRTN
/ 6
ALU ,R15,ADUS,MOP,SUBR,SUBR,DA
REG ,R15,TZERO,R15
DBUS SNOKE & CARRISEL CZERO
STATSHT B010000010000
/ 6
ENTRY POINT FOR DIRECT OR DIRECT IMPLIED ADDRESSING

0466 : MAC ,FALSE,TZERO,CRTN
/ 6
ALU ,R15,ADUS,MOP,SUBR,SUBR,DA
REG ,R15,TZERO,R15
DBUS SNOKE & CARRISEL CZERO
STATSHT B010000010000
/ 6
ENTRY POINT FOR DIRECT OR DIRECT IMPLIED ADDRESSING

0467 : MAC ,FALSE,TZERO,CJS
/ 6
ALU & DBUS SNOKE & ADDRESS OFD11

0468 : MAC ,FALSE,TZERO,CRTN
/ 6
ALU ,R15,ADUS,MOP,SUBR,SUBR,DA
REG ,R15,TZERO,R15
DBUS SNOKE & CARRISEL CZERO
STATSHT B010000010000
/ 6
ENTRY POINT FOR IMMEDIATE ADDRESSING

1 MAY 1981

PAGE 28

AMDOS/28 AMDASH MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EMSL)

```

046A / 6 THE UPPER LIMIT IS SUBTRACTED FROM THE CONTENTS OF GR1. THE MACHINE STATUS
      / 6 IS SET ON THE RESULT OF THE SUBTRACTION. CARRY IS SET EQUAL TO ZERO.
      / 6 MAC ...CONT
      / 6 ALU ...ABUS,NOP,SUBR,SUBR,DA
      / 6 REG ...GR1,RO & DBUS SNONE & CARRYSEL CZERO
      / 6 STATSEPT B#010000010000

046B / 6 RETURN TAKES PLACE IF THE RESULT OF THE SUBTRACTION WAS POSITIVE INDICATING
      / 6 THAT THE CONTENTS OF GR1 WERE GREATER THAN THE UPPER LIMIT.
      / 6 MAC TRUE,MUE,CRYN
      / 6 ALU & DBUS SNONE & STATSEPT B#000000100110

046C / 6 THE CONTENTS OF GR1 ARE BETWEEN THE LIMITS SO THE MACHINE STATUS IS SET
      / 6 FOR A ZERO RESULT.
      / 6 MAC FALSE,TZFR,CRTN
      / 6 ALU ...NOP,AND,AND,DZ
      / 6 DBUS SNONE & CARRYSEL CZERO
      / 6 STATSEPT B#00000010000

```

INCREMENT MEMORY BY POSITIVE INTEGER

ENTRY POINT

```

046D / 6 (1) JUMP SUB TAKES PLACE TO OPD11
      / 6 OPCODES: MAC FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS OPD11
      / 6 THE INSTRUCTION IN IR0 IS SHIPPED WITH THE CONTENTS OF IR0.
      / 6 MAC ...CONT & ALU ...ABUS,RAM,OR,OR,DZ & REG ...RZERO,RZERO,R1
      / 6 CEARL & CEREZ & DBUS SALU

046E / 6 (3) THE INSTRUCTION IN IR0 IS DOWNSHIFTED.
      / 6 MAC ...CONT & ALU ...RAMD,OR,OR,DZ & REG ...RZERO,RZERO & CEARL
      / 6 DBUS SNONE & STATSEPT B#000000000000

046F / 6 (4) THE INSTRUCTION IN IR0 IS DOWNSHIFTED.
      / 6 MAC ...CONT & ALU ...RAMD,OR,OR,DZ & REG ...RZERO,RZERO,R1
      / 6 CEARL & CEREZ & DBUS SALU

```

1 MAY 1981

PAGE 21

AMDOS/28 AMDASH MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EMSL)

```

0470 / 6 MAC ...CONT & ALU ...RAMD,OR,OR,DZ & REG ...RZERO,RZERO & CEARL
      / 6 DBUS SNONE & STATSEPT B#000000000000

0471 / 6 (5) THE INSTRUCTION IN IR0 IS MASKED BY 00000000111111 AND DOWNSHIFTED.
      / 6 MAC ...CONT & ALU ...DBUS,AND,AND,DA & REG ...RZERO,RZERO
      / 6 DBUS SDATA & DATINSRT D#63 & CEARL

0472 / 6 (6) 2 IS ADDED TO THE CONTENTS OF IR0 AND IT IS DOWNSHIFTED.
      / 6 MAC ...CONT & ALU ...DBUS,AND,AND,DA & REG ...RZERO,RZERO & CEARL
      / 6 DBUS SDATA & DATINSRT D#2

0473 / 6 (7) THE POSITIVE INTEGER NOW OBTAINED IS ADDED TO THE CONTENTS OF RRS.
      / 6 MAC ...CONT & ALU ...RRO,ABUS,NOP,ADD,ADD,DA & DBUS SALU & CEREZ
      / 6 REG ...RZERO,R5 & STATSEPT B#00000010000 & CARRYSEL CGAR
      / 6 MACSTEN B#0000

0474 / 6 (3) JUMP TO OSGENPUR TAKES PLACE IF OVERFLOW WAS NOT SET BY THE PREVIOUS
      / 6 INSTRUCTION.
      / 6 MAC FALSE,OVER,CJP & ALU ...ABUS,AND,AND,DA & REG ...RZERO,R1
      / 6 DBUS SNONE & CEARL & BADDRESS OSGENPUR

0475 / 6 (4) JUMP SUB TO OSGENPUR TAKES PLACE.
      / 6 MAC FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS OSGENPUR

0476 / 6 (5) JUMP TO OVERFLOW TAKES PLACE.
      / 6 MAC ...JP & ALU & DBUS SNONE & BADDRESS OVERFLOW

```

DECREMENT MEMORY BY POSITIVE INTEGER

DECREMENT MEMORY BY POSITIVE INTEGER

ENTRY POINT

```

0477 / 6 (1) JUMP SUB TAKES PLACE TO OPD11
      / 6 OPCODES: MAC FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS OPD11
      / 6 THE INSTRUCTION IN IR0 IS SHIPPED WITH THE CONTENTS OF IR0.
      / 6 MAC ...CONT & ALU ...ABUS,RAM,OR,OR,DZ & REG ...RZERO,RZERO,R1
      / 6 CEARL & CEREZ & DBUS SALU

```

1 MAY 1981

PAGE 22

AMDOS/29 AMDASM MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EMSL)

```

: (3) THE INSTRUCTION IN IR0 IS DOWNSHIFTED.
0479 / 5 MAC ...CONT & ALU ...RAMF,OP,CR,TA & REG ...RZERO,RZERO & CICALU
      DBUS SNOKE & STATSHFT #0000000000000000
: (4) THE INSTRUCTION IN IR0 IS DOWNSHIFTED.
047A / 5 MAC ...CONT & ALU ...RAMD,OP,CR,TA & REG ...RZERO,RZERO & CICALU
      DBUS SNOKE & STATSHFT #0000000000000000
: (5) THE INSTRUCTION IN IR0 IS MASKED BY 0000000000000000 AND DOWNSHIFTED.
047B / 5 MAC ...CONT & ALU ...RAMD,AND,AND,DA & REG ...RZERO,RZERO
      DBUS SNOKE & STATSHFT #0000000000000000
: (6) 2 IS ADDED TO THE CONTENTS OF IR0 AND THE RESULT DOWNSHIFTED.
      THE BIT PATTERN FOR THE DATA INSERSION IS OK FOR THE SHIFT CONTROL ALSO.
047C / 5 MAC ...CONT & ALU ...DBUS,RAND,ADD,DA & REG ...RZERO,RZERO
      DBUS SNOKE & STATSHFT #0000000000000000
: (7) THE POSITIVE INTERCEPT NOW OBTAINED IS SUBTRACTED FROM THE CONTENTS OF IR0.
047D / 5 MAC ...CONT & ALU ...RAMF,OP,CR,TA & REG ...RZERO,RZERO
      REG ...RZERO,RZERO & STATSHFT #0000000000000000
      MACSTEN #00000000
: (3) JUMP TO OSCENPUP TAKES PLACE IF OVERFLOW WAS NOT SET BY THE PREVIOUS
      INSTRUCTION.
047E / 5 MAC ...FALSE,OP,CR,TA & ALU ...RAMF,OP,OP,OP,DA & REG ...RZERO,RZERO
      DBUS SNOKE & CICALU & PADDRSS OSCENPUP
: (4) JUMP SUB TO OSCENPUP.
047F / 5 MAC ...FALSE,RZERO,CJS & ALU & DBUS SNOKE & PADDRSS OSCENPUP
: (5) JUMP TO OVERFLOW TAKES PLACE.
0480 / 5 MAC ...JP & ALU & DBUS SNOKE & PADDRSS OVERFLOW

```

SINGLE PRECISION NIZATE REGISTER

```

: (1) THE CONTENTS OF CR2 (RB) ARE PLACED IN CR1 (RA), THE MACHINE STATUS IS
      LOADED.
0481 / 5 MAC ...CONT & ALU ...RAMF,OP,OP,TA & REG ...RZERO,RZERO
      CICALU & DBUS SNOKE & STATSHFT #0000000000000000 & CARNISEL CZERO
      MACSTEN #00000000

```

1 MAY 1981

PAGE 23

AMDOS/29 AMDASM MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EMSL)

```

: (2) RETURN TAKES PLACE IF MACHINE STATUS N IS NOT SET.
0482 / 5 MAC ...FALSE,MUI,CRFN & ALU & DBUS SNOKE & STATSHFT #0000000000000000
: (3) THE CONTENTS OF CR2 (RB) ARE NEGATED AND PLACED IN CR1 (RA).
0483 / 5 MAC ...FALSE,TZERO,CRFN & ALU ...RAMF,SUBS,SUBS,TA & CICALU
      DBUS SNOKE & STATSHFT #0000000000000000 & CARNISEL CZERO
      REG ...RZERO,RZERO

```

SINGLE PRECISION NIZATE REGISTER

ENTRY POINT

```

: (1) THE CONTENTS OF CR2 (RB) ARE NEGATED AND PLACED IN CR1 (RA).
      THE MACHINE STATUS IS SET ACCORDINGLY AND RETURN TAKES PLACE.
0484 / 5 MAC ...CONT & ALU ...RAMF,OP,OP,TA & REG ...RZERO,RZERO
      REG ...RZERO,RZERO & STATSHFT #0000000000000000 & CARNISEL CZERO
      STATSHFT #0000000000000000

```

```

: (2) RETURN OCCURS IF OVERFLOW HAS NOT BEEN SET.
0485 / 5 MAC ...FALSE,OP,CRFN & ALU & DBUS SNOKE

```

```

: (3) JUMP TO OVERFLOW
0486 / 5 MAC ...JP & ALU & DBUS SNOKE & PADDRSS OVERFLOW

```

INCLUSIVE LOGICAL OR

ENTRY POINT FOR REGISTER ADDRESSING

```

: (1) THE CONTENTS OF CR2 ARE ORED WITH THE CONTENTS OF CR1 AND THE RESULT PLACED
      IN CR1. THE MACHINE STATUS REGISTER IS LOADED ACCORDING TO THE RESULT.
0487 / 5 MAC ...FALSE,TZERO,CRFN & ALU ...RAMF,OP,OP,TA & REG ...RZERO,RZERO
      CICALU & DBUS SNOKE & STATSHFT #0000000000000000 & CARNISEL CZERO
      MACSTEN #00000000

```

ENTRY POINT FOR BASE RELATIVE ADDRESSING

1 MAY 1981

PAGE 24

AMDOS/29 AMDASH MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EMSL)

```

0480 OPD036:: MAC ,FALSE,TZERO,CJS & ALU & DBUS SNOKE & BADDRESS OFB1
      (1) JUMP SUB TO OFB1
      (2) THE DERIVED OPERAND IS ORED WITH THE CONTENTS OF IR2 AND THE RESULT
      PLACED IN IR2
0489 / &
      MAC ,FALSE,TZERO,CRTN & ALU ,B#0,ABUS,RAMP,OR,OR,DA
      REG R#10,R#10,RZERO,RZERO,R5 & DBUS SNOKE & STATSHFT #000000010000
      CICALU & CARRISEL CZERO
      ENTRY POINT FOR BASE RELATIVE INDEXED ADDRESSING
      *****
048A OPD040F: MAC ,FALSE,TZERO,CJS & ALU & DBUS SNOKE & BADDRESS OFB1
      (1) JUMP SUB TO OFB1
      (2) THE DERIVED OPERAND IS ORED WITH THE CONTENTS OF IR2 AND THE RESULT
      PLACED IN IR2
048B / &
      MAC ,FALSE,TZERO,CRTN & ALU ,B#0,ABUS,RAMP,OR,OR,DA
      REG R#10,R#10,RZERO,RZERO,R5 & DBUS SNOKE & STATSHFT #000000010000
      CICALU & CARRISEL CZERO
      ENTRY POINT FOR DIRECT OR DIRECT INDEXED ADDRESSING
      *****
048C OPD040E:: MAC ,FALSE,TZERO,CJS & ALU & DBUS SNOKE & BADDRESS OFD11
      (1) JUMP SUB TO OFD11
      (2) THE DERIVED OPERAND IS ORED WITH THE CONTENTS OF GRI AND THE RESULT PLACED
      IN GRI.
048D / &
      MAC ,FALSE,TZERO,CRTN & ALU ,B#0,ABUS,RAMP,OR,OR,DA
      REG ,GRI,GRI,R5 & DBUS SNOKE & STATSHFT #000000010000 & CICALU
      CARRISEL CZERO
      ENTRY POINT FOR IMMEDIATE ADDRESSING
      *****
048E OPD04AB: MAC ,FALSE,TZERO,CJS & ALU & DBUS SNOKE & BADDRESS OFIM
      (1) JUMP SUB TO OFIM
      (2) THE DERIVED OPERAND IS ORED WITH THE CONTENTS OF GRI AND THE RESULT LEFT
      IN GRI.
048F / &
      MAC ,FALSE,TZERO,CRTN & ALU ,B#0,ABUS,RAMP,OR,OR,DA
      REG ,GRI,GRI,R5 & DBUS SNOKE & STATSHFT #000000010000 & CICALU
      CARRISEL CZERO

```

1 MAY 1981

PAGE 25

AMDOS/29 AMDASH MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EMSL)

```

      LOGICAL AND
      *****
      ENTRY POINT FOR REGISTER ADDRESSING
      (1) THE CONTENTS OF GR2 (R2) ARE AND'ED WITH THE CONTENTS OF GRI (R1).
      THE RESULT IS STORED IN GRI (R1).
0490 OPD04E1: MAC ,FALSE,TZERO,CRTN & ALU ,B#0,RAMP,AND,AND,AB & REG ,GRI,GR2
      / &
      CICALU & DBUS SNOKE & STATSHFT #000000010000
      CARRISEL CZERO
      ENTRY POINT FOR BASE RELATIVE ADDRESSING
      *****
0491 OPD04E1: MAC ,FALSE,TZERO,CJS & ALU & DBUS SNOKE & BADDRESS OFB1
      (1) JUMP SUB TO OFB1
      (2) THE DERIVED OPERAND IS AND'ED WITH THE CONTENTS OF IR2 AND THE RESULT
      LEFT IN IR2
0492 / &
      MAC ,FALSE,TZERO,CRTN & ALU ,B#0,ABUS,RAMP,AND,AND,DA
      REG R#10,R#10,RZERO,RZERO,R5 & DBUS SNOKE
      STATSHFT #000000010000
      CARRISEL CZERO
      ENTRY POINT FOR BASE RELATIVE INDEXED ADDRESSING.
      *****
0493 OPD04E1: MAC ,FALSE,TZERO,CJS & ALU & DBUS SNOKE & BADDRESS OFB1
      (1) JUMP SUB TO OFB1
      (2) THE DERIVED OPERAND IS AND'ED WITH THE CONTENTS OF IR2 AND THE RESULT LEFT
      IN IR2.
0494 / &
      MAC ,FALSE,TZERO,CRTN & ALU ,B#0,ABUS,RAMP,AND,AND,DA
      REG R#10,R#10,RZERO,RZERO,R5 & DBUS SNOKE
      STATSHFT #000000010000
      CARRISEL CZERO
      ENTRY POINT FOR DIRECT AND DIRECT INDEXED ADDRESSING
      *****
0495 OPD04E2: MAC ,FALSE,TZERO,CJS & ALU & DBUS SNOKE & BADDRESS OFD11
      (1) JUMP SUB TO OFD11

```

1 MAY 1981

PAGE 26

AMDOS/29 AMDASM MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EMSL)

```

0496      (2) THE DERIVED OPERAND IS AND'ED WITH THE CONTENTS OF GRI AND THE RESULT
          : PLACED IN GRI.
          : MAC ,FALSE,TZERO,CRTN & ALU ,B#0,ABUS,RAMP,AND,AND,DA
          : REG ,GRI,GRI,R5 & CICALU & DBUS SNOKE & STATSBT B#000000010000
          : / 5
          : CARRISEL CZERO
          : ENTRY POINT FOR IMMEDIATE ADDRESSING
          : *****

```

```

0497 TOPCDA7: MAC ,FALSE,TZERO,CJS & ALU & DBUS SNOKE & BADDRESS OPD1

```

```

          : (2) THE DERIVED OPERAND IS AND'ED WITH THE CONTENTS OF GRI (RA) AND THE RESULT
          : LEFT IN GRI (RA).
          : MAC ,FALSE,TZERO,CRTN & ALU ,B#0,ABUS,RAMP,AND,AND,DA
          : REG ,GRI,GRI,R5 & CICALU & DBUS SNOKE & STATSBT B#000000010000
          : / 5
          : CARRISEL CZERO

```

```

          : EXCLUSIVE LOGICAL OR
          : *****

```

```

          : ENTRY POINT FOR REGISTER ADDRESSING
          : *****

```

```

          : (1) THE CONTENTS OF CR2 (RB) IS EXCLUSIVE OR'ED WITH THE CONTENTS OF GRI (RB).
          : THE RESULT IS LEFT IN GRI (RB).

```

```

0499 OPDCE3: MAC ,FALSE,TZERO,CRTN & ALU ,B#0,RAMP,EIOR,EIOR,DA
          : REG ,GRI,GRI,R5 & CICALU & DBUS SNOKE & STATSBT B#000000010000
          : / 5
          : CARRISEL CZERO

```

```

          : ENTRY POINT FOR DIRECT OR DIRECT IMMEDIATE ADDRESSING
          : *****

```

```

049A OPDCE4: MAC ,FALSE,TZERO,CJS & ALU & DBUS SNOKE & BADDRESS OPD1
          : (2) THE DERIVED OPERAND IS EXCLUSIVE OR'ED WITH THE CONTENTS OF GRI (RA) AND
          : THE RESULT LEFT IN GRI (RA).

```

```

049B      : MAC ,FALSE,TZERO,CRTN & ALU ,B#0,ABUS,RAMP,EIOR,EIOR,DA
          : REG ,GRI,GRI,R5 & CICALU & DBUS SNOKE & STATSBT B#000000010000
          : / 5
          : CARRISEL CZERO

```

```

          : ENTRY POINT FOR IMMEDIATE ADDRESSING
          : *****

```

1 MAY 1981

PAGE 27

AMDOS/29 AMDASM MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EMSL)

```

          : (1) JUMP SUB TO OPD1
          : 049C TOPCDA9: MAC ,FALSE,TZERO,CJS & ALU & DBUS SNOKE & BADDRESS OPD1
          : (2) THE DERIVED OPERAND IS EX OR'ED WITH THE CONTENTS OF GRI (RA) AND THE RESULT
          : LEFT IN GRI (RA).
          : MAC ,FALSE,TZERO,CRTN & ALU ,B#0,ABUS,RAMP,EIOR,EIOR,DA
          : REG ,GRI,GRI,R5 & CICALU & DBUS SNOKE & STATSBT B#000000010000
          : / 5
          : CARRISEL CZERO

```

```

          : LOGICAL AND
          : *****

```

```

          : ENTRY POINT FOR REGISTER ADDRESSING
          : *****

```

```

          : (1) THE CONTENTS OF CR2 (RB) ARE AND'ED WITH THE CONTENTS OF GRI (RA) AND
          : THE RESULT LEFT IN GRI (RA).
          : MAC ,FALSE,TZERO,CRTN & ALU ,B#0,RAMP,AND,AND,AND,DA & REG ,GRI,GRI & CICALU
          : / 5
          : DBUS SNOKE

```

```

          : (2) THE CONTENTS OF GRI (RA) ARE INVERTED (BY TAKING THE EXCLUSIVE NOR WITH
          : ZERO) AND PLACED IN GRI (RA).

```

```

049F      : MAC ,FALSE,TZERO,CRTN & ALU ,B#0,RAMP,EIOR,EIOR,DA
          : REG ,GRI,GRI,R5 & CICALU & DBUS SNOKE & STATSBT B#000000010000
          : / 5
          : CARRISEL CZERO

```

```

          : ENTRY POINT FOR DIRECT OR DIRECT IMMEDIATE ADDRESSING
          : *****

```

```

04A0 OPDCE6: MAC ,FALSE,TZERO,CJS & ALU & DBUS SNOKE & BADDRESS OPD1
          : (2) THE DERIVED OPERAND IS AND'ED WITH THE CONTENTS OF GRI (RA) AND THE RESULT
          : LEFT IN GRI (RA).

```

```

04A1      : MAC ,FALSE,TZERO,CRTN & ALU ,B#0,ABUS,RAMP,AND,AND,DA & REG ,GRI,GRI,R5 & CICALU
          : / 5
          : DBUS SNOKE

```

```

          : (3) THE NOT OF THE CONTENTS OF GRI (RA) IS TAKEN AND THE RESULT PLACED IN
          : GRI (RA).

```

```

04A2      : MAC ,FALSE,TZERO,CRTN & ALU ,B#0,RAMP,EIOR,EIOR,DA
          : REG ,GRI,GRI & CICALU & DBUS SNOKE & STATSBT B#000000010000
          : / 5

```

```

          : ENTRY POINT FOR IMMEDIATE ADDRESSING
          : *****

```

EOT SH KL

MAY 1941

AMDOS/PS ACADEMIC (1969), V. 2, WITH SUPPL. P. 1762. (EML)

[illegible]

0440 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000 1001 1002 1003 1004 1005 1006 1007 1008 1009 1010 1011 1012 1013 1014 1015 1016 1017 1018 1019 1020 1021 1022 1023 1024 1025 1026 1027 1028 1029 1030 1031 1032 1033 1034 1035 1036 1037 1038 1039 1

(2) THE INSTRUCTION FROM FBI IS MARKED AND LOADED IN 1991. THE CONTINUED OF 192 ARE TEMPORARILY PLACED IN 1991.

SHIP LEFT LOGICAL
ENTRY POINT

```

04AE OP000000:: MAC -FALSE, T7PRO,CJS
/ & ALU & INUS, SMOFT & ADDRESS SHFTMM
: ; (2) THE CONTENTS OF GR2 ARE UPSHIFTED LOGICAL.
04AP SELL: MAC ...CONT

```

0490
/ 5 STATIST PR#0601000000
CRIO S REG UNLDRZ
/ 5

(S) I A SUBPATED FROM THE CONTENTS OF BRQ
NAC ...CONT
/ 6 ALC PLANNING SUPR. DIBS. DZ

•
Y
•
•
•
•

PAGE 12

AMERICAN ASSOCIATION OF UNIVERSITY AND COLLEGE LIBRARIANS

[illegible]

3

8

CT

7

6

5

4

3

2

1

1. 2. 3. 4. 5. 6. 7. 8. 9. 10. 11. 12. 13. 14. 15. 16. 17. 18. 19. 20. 21. 22. 23. 24. 25. 26. 27. 28. 29. 30. 31. 32. 33. 34. 35. 36. 37. 38. 39. 40. 41. 42. 43. 44. 45. 46. 47. 48. 49. 50. 51. 52. 53. 54. 55. 56. 57. 58. 59. 60. 61. 62. 63. 64. 65. 66. 67. 68. 69. 70. 71. 72. 73. 74. 75. 76. 77. 78. 79. 80. 81. 82. 83. 84. 85. 86. 87. 88. 89. 90. 91. 92. 93. 94. 95. 96. 97. 98. 99. 100. 101. 102. 103. 104. 105. 106. 107. 108. 109. 110. 111. 112. 113. 114. 115. 116. 117. 118. 119. 120. 121. 122. 123. 124. 125. 126. 127. 128. 129. 130. 131. 132. 133. 134. 135. 136. 137. 138. 139. 140. 141. 142. 143. 144. 145. 146. 147. 148. 149. 150. 151. 152. 153. 154. 155. 156. 157. 158. 159. 160. 161. 162. 163. 164. 165. 166. 167. 168. 169. 170. 171. 172. 173. 174. 175. 176. 177. 178. 179. 180. 181. 182. 183. 184. 185. 186. 187. 188. 189. 190. 191. 192. 193. 194. 195. 196. 197. 198. 199. 200. 201. 202. 203. 204. 205. 206. 207. 208. 209. 210. 211. 212. 213. 214. 215. 216. 217. 218. 219. 220. 221. 222. 223. 224. 225. 226. 227. 228. 229. 230. 231. 232. 233. 234. 235. 236. 237. 238. 239. 240. 241. 242. 243. 244. 245. 246. 247. 248. 249. 250. 251. 252. 253. 254. 255. 256. 257. 258. 259. 260. 261. 262. 263. 264. 265. 266. 267. 268. 269. 270. 271. 272. 273. 274. 275. 276. 277. 278. 279. 280. 281. 282. 283. 284. 285. 286. 287. 288. 289. 290. 291. 292. 293. 294. 295. 296. 297. 298. 299. 300. 301. 302. 303. 304. 305. 306. 307. 308. 309. 310. 311. 312. 313. 314. 315. 316. 317. 318. 319. 320. 321. 322. 323. 324. 325. 326. 327. 328. 329. 330. 331. 332. 333. 334. 335. 336. 337. 338. 339. 340. 341. 342. 343. 344. 345. 346. 347. 348. 349. 350. 351. 352. 353. 354. 355. 356. 357. 358. 359. 360. 361. 362. 363. 364. 365. 366. 367. 368. 369. 370. 371. 372. 373. 374. 375. 376. 377. 378. 379. 380. 381. 382. 383. 384. 385. 386. 387. 388. 389. 390. 391. 392. 393. 394. 395. 396. 397. 398. 399. 400. 401. 402. 403. 404. 405. 406. 407. 408. 409. 410. 411. 412. 413. 414. 415. 416. 417. 418. 419. 420. 421. 422. 423. 424. 425. 426. 427. 428. 429. 430. 431. 432. 433. 434. 435. 436. 437. 438. 439. 440. 441. 442. 443. 444. 445. 446. 447. 448. 449. 450. 451. 452. 453. 454. 455. 456. 457. 458. 459. 460. 461. 462. 463. 464. 465. 466. 467. 468. 469. 470. 471. 472. 473. 474. 475. 476. 477. 478. 479. 480. 481. 482. 483. 484. 485. 486. 487. 488. 489. 490. 491. 492. 493. 494. 495. 496. 497. 498. 499. 500. 501. 502. 503. 504. 505. 506. 507. 508. 509. 510. 511. 512. 513. 514. 515. 516. 517. 518. 519. 520. 521. 522. 523. 524. 525. 526. 527. 528. 529. 530. 531. 532. 533. 534. 535. 536. 537. 538. 539. 540. 541. 542. 543. 544. 545. 546. 547. 548. 549. 550. 551. 552. 553. 554. 555. 556. 557. 558. 559. 560. 561. 562. 563. 564. 565. 566. 567. 568. 569. 570. 571. 572. 573. 574. 575. 576. 577. 578. 579. 580. 581. 582. 583. 584. 585. 586. 587. 588. 589. 590. 591. 592. 593. 594. 595. 596. 597. 598. 599. 600. 601. 602. 603. 604. 605. 606. 607. 608. 609. 610. 611. 612. 613. 614. 615. 616. 617. 618. 619. 620. 621. 622. 623. 624. 625. 626. 627. 628. 629. 630. 631. 632. 633. 634. 635. 636. 637. 638. 639. 640. 641. 642. 643. 644. 645. 646. 647. 648. 649. 650. 651. 652. 653. 654. 655. 656. 657. 658. 659. 660. 661. 662. 663. 664. 665. 666. 667. 668. 669. 670. 671. 672. 673. 674. 675. 676. 677. 678. 679. 680. 681. 682. 683. 684. 685. 686. 687. 688. 689. 690. 691. 692. 693. 694. 695. 696. 697. 698. 699. 700. 701. 702. 703. 704. 705. 706. 707. 708. 709. 710. 711. 712. 713. 714. 715. 716. 717. 718. 719. 720. 721. 722. 723. 724. 725. 726. 727. 728. 729. 730. 731. 732. 733. 734. 735. 736. 737. 738. 739. 740. 741. 742. 743. 744. 745. 746. 747. 748. 749. 750. 751. 752. 753. 754. 755. 756. 757. 758. 759. 760. 761. 762. 763. 764. 765. 766. 767. 768. 769. 770. 771. 772. 773. 774. 775. 776. 777. 778. 779. 780. 781. 782. 783. 784. 785. 786. 787. 788. 789. 790. 791. 792. 793. 794. 795. 796. 797. 798. 799. 800. 801. 802. 803. 804. 805. 806. 807. 808. 809. 810. 811. 812. 813. 814. 815. 816. 817. 818. 819. 820. 821. 822. 823. 824. 825. 826. 827. 828. 829. 830. 831. 832. 833. 834. 835. 836. 837. 838. 839. 840. 84

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25

7-10-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25-26-27-28-29-30-31-32-33-34-35-36-37-38-39-40-41-42-43-44-45-46-47-48-49-50-51-52-53-54-55-56-57-58-59-60-61-62-63-64-65-66-67-68-69-70-71-72-73-74-75-76-77-78-79-80-81-82-83-84-85-86-87-88-89-90-91-92-93-94-95-96-97-98-99-100-101-102-103-104-105-106-107-108-109-110-111-112-113-114-115-116-117-118-119-120-121-122-123-124-125-126-127-128-129-130-131-132-133-134-135-136-137-138-139-140-141-142-143-144-145-146-147-148-149-150-151-152-153-154-155-156-157-158-159-160-161-162-163-164-165-166-167-168-169-170-171-172-173-174-175-176-177-178-179-180-181-182-183-184-185-186-187-188-189-190-191-192-193-194-195-196-197-198-199-200-201-202-203-204-205-206-207-208-209-210-211-212-213-214-215-216-217-218-219-220-221-222-223-224-225-226-227-228-229-230-231-232-233-234-235-236-237-238-239-240-241-242-243-244-245-246-247-248-249-250-251-252-253-254-255-256-257-258-259-260-261-262-263-264-265-266-267-268-269-270-271-272-273-274-275-276-277-278-279-280-281-282-283-284-285-286-287-288-289-290-291-292-293-294-295-296-297-298-299-300-301-302-303-304-305-306-307-308-309-310-311-312-313-314-315-316-317-318-319-320-321-322-323-324-325-326-327-328-329-330-331-332-333-334-335-336-337-338-339-340-341-342-343-344-345-346-347-348-349-350-351-352-353-354-355-356-357-358-359-360-361-362-363-364-365-366-367-368-369-370-371-372-373-374-375-376-377-378-379-380-381-382-383-384-385-386-387-388-389-390-391-392-393-394-395-396-397-398-399-400-401-402-403-404-405-406-407-408-409-410-411-412-413-414-415-416-417-418-419-420-421-422-423-424-425-426-427-428-429-430-431-432-433-434-435-436-437-438-439-440-441-442-443-444-445-446-447-448-449-450-451-452-453-454-455-456-457-458-459-460-461-462-463-464-465-466-467-468-469-470-471-472-473-474-475-476-477-478-479-480-481-482-483-484-485-486-487-488-489-490-491-492-493-494-495-496-497-498-499-500-501-502-503-504-505-506-507-508-509-510-511-512-513-514-515-516-517-518-519-520-521-522-523-524-525-526-527-528-529-530-531-532-533-534-535-536-537-538-539-540-541-542-543-544-545-546-547-548-549-550-551-552-553-554-555-556-557-558-559-560-561-562-563-564-565-566-567-568-569-570-571-572-573-574-575-576-577-578-579-580-581-582-583-584-585-586-587-588-589-590-591-592-593-594-595-596-597-598-599-600-601-602-603-604-605-606-607-608-609-610-611-612-613-614-615-616-617-618-619-620-621-622-623-624-625-626-627-628-629-630-631-632-633-634-635-636-637-638-639-640-641-642-643-644-645-646-647-648-649-650-651-652-653-654-655-656-657-658-659-660-661-662-663-664-665-666-667-668-669-670-671-672-673-674-675-676-677-678-679-680-681-682-683-684-685-686-687-688-689-690-691-692-693-694-695-696-697-698-699-700-701-702-703-704-705-706-707-708-709-710-711-712-713-714-715-716-717-718-719-720-721-722-723-724-725-726-727-728-729-730-731-732-733-734-735-736-737-738-739-740-741-742-743-744-745-746-747-748-749-750-751-752-753-754-755-756-757-758-759-760-761-762-763-764-765-766-767-768-769-770-771-772-773-774-775-776-777-778-779-780-781-782-783-784-785-786-787-788-789-790-791-792-793-794-795-796-797-798-799-800-801-802-803-804-805-806-807-808-809-810-811-812-813-814-815-816-817-818-819-820-821-822-823-824-825-826-827-828-829-830-831-832-833-834-835-836-837-838-839-840-841-842-843-844-845-846-847-848-849-850-851-852-853-854-855-856-857-858-859-860-861-862-863-864-865-866-867-868-869-870-871-872-873-874-875-876-877-878-879-880-881-882-883-884-885-886-887-888-889-890-891-892-893-894-895-896-897-898-899-900-901-902-903-904-905-906-907-908-909-910-911-912-913-914-915-916-917-918-919-920-921-922-923-924-925-926-927-928-929-930-931-932-933-934-935-936-937-938-939-940-941-942-943-944-945-946-947-948-949-950-951-952-953-954-955-956-957-958-959-960-961-962-963-964-965-966-967-968-969-970-971-972-973-974-975-976-977-978-979-980-981-982-983-984-985-986-987-988-989-990-991-992-993-994-995-996-997-998-999-1000-1001-1002-1003-1004-1005-1006-1007-1008-1009-1010-1011-1012-1013-1014-1015-1016-1017-1018-1019-1020-1021-1022-1023-1024-1025-1026-1027-1028-1029-1030-1031-1032-1033-1034-1035-1036-1037-1038-1039-1040-1041-1042-1043-10

PAGE 12

[illegible][illegible][illegible]

| | |
|----|----|
| 1 | 1 |
| 2 | 2 |
| 3 | 3 |
| 4 | 4 |
| 5 | 5 |
| 6 | 6 |
| 7 | 7 |
| 8 | 8 |
| 9 | 9 |
| 10 | 10 |

2000

[illegible]

945A 201:
WA: " " "
ALB: " " " P.M.: 76. F. 7 A
/ S

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
84

1. (1) 1 IS SUPPRACTED FROM THE CONTENTS OF YRS

0400
/ 1
AC . C. H.
AC . ARIZ. N. P. SUB. S. P. S. 12

16
TATSHIT INPOPPA, Q10P02 S (K9E)
TALU (M) , , , A. S (P)S SAH

141 JUMP TO 0001 TAKE 4 PLACES IF 2.00 IS NOT SET.

101 STATION ST. N.W. SEAS 3 CITY 5 / 0542

16) RETURN TABLE PLACE, MACHINE STATUS, RETURN TO LOADING.

[illegible]

... ..

AD-A114 029

ROYAL AIRCRAFT ESTABLISHMENT FARNBOROUGH (ENGLAND) F/G 9/2
AN IMPLEMENTATION OF MIL-STD-1750 AIRBORNE COMPUTER INSTRUCTION--ETC(U)
MAY 81 S J SHRIMPON
RAE-F5-403

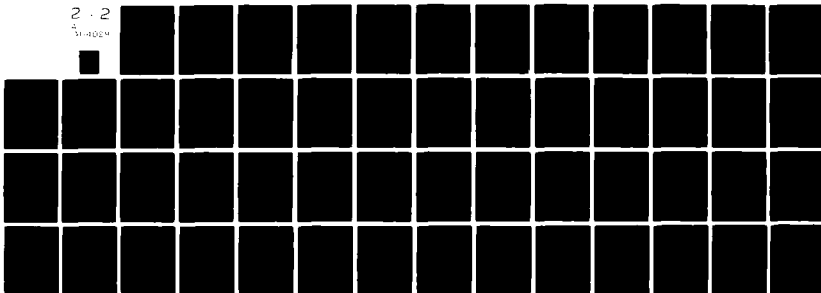
UNCLASSIFIED

DRIC-BR-80502

NL

2 - 2

31-0029



END

DATE

FILMED

5-82

DTIC

1 MAY 1981

PAGE 34

AMDS/29 ANDASH MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EMSL)

DOUBLE SHIFT LEFT LOGICAL

ENTRY POINT

0402 OP000566:: MAC ,FALSE,ZERO,CJS
/ 6 ALU & DBUS SNOKE & ADDRESS SEPTMM

0407 DSRL2: (2) THE CONTENTS OF REGISTER RB+1 (GR2+1) ARE LOADED INTO THE QREG
/ 6 MAC ,CONT
/ 6 ALU ,DBUS,OR,OR,ZA
/ 6 REG ,DBUS,GR2 & DBUS SNOKE & C&ALU

0409 DSRL1: (3) THE CONTENTS OF RB (GR2) AND QREG ARE CONCATENATED AND UPSHIFTED
/ 6 LOGICAL
/ 6 MAC ,CONT
/ 6 ALU ,DBUS,OR,OR,ZA
/ 6 REG ,GR2,GR2
/ 6 DBUS SNOKE & C&ALU & STATSEPT DBUS0110000000

0401 (4) 1 IS SUBTRACTED FROM THE CONTENTS OF RB9 (THE CARRY INPUT IS SET
/ 6 TO 0 BY STATSEPT CAUSING THIS TO HAPPEN)
/ 6 MAC ,CONT
/ 6 ALU ,DBUS,MOP,SUBS,DB
/ 6 REG ,R9 & DBUS SALU
/ 6 CREG & STATSEPT DBUS0000000000

0402 (5) JUMP TO DSRL1 TAKES PLACE IF ZLSB IS NOT SET
/ 6 THE MICRO STATUS REGISTER IS LOADED ACCORDING TO THE CONTENTS OF
/ 6 QREG (THE MOST SIGNIFICANT HALF OF THE DOUBLE LENGTH
/ 6 WORD.
/ 6 MAC ,FALSE,ZLSB,CJP
/ 6 ALU & DBUS SNOKE & ADDRESS DSRL1

0403 (6) JUMP SUB TAKES PLACE TO ASDLSG2, THE ROUTINE THAT DETERMINES THE
/ 6 SETTING OF THE STATUS BITS FOR A DOUBLE LENGTH WORD IN GR2 AND GR2+1.
/ 6 MAC ,FALSE,ZERO,CJS
/ 6 ALU & DBUS SNOKE & ADDRESS ASDLSG2

0404 (9) RETURN TAKES PLACE.
/ 6 MAC ,FALSE,ZERO,CRTN & ALU & DBUS SNOKE

1 MAY 1981

PAGE 35

AMDS/29 ANDASH MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EMSL)

DOUBLE SHIFT RIGHT LOGICAL

ENTRY POINT

0405 OP000566:: MAC ,FALSE,ZERO,CJS
/ 6 ALU & DBUS SNOKE & ADDRESS SEPTMM

0406 DSRL2: (2) THE CONTENTS OF REGISTER RB+1 (GR2+1) ARE LOADED INTO THE QREG
/ 6 MAC ,CONT
/ 6 ALU ,DBUS,OR,OR,ZA
/ 6 REG ,DBUS,GR2 & DBUS SNOKE & C&ALU

0407 DSRL1: (3) THE CONTENTS OF RB (GR2) AND QREG ARE CONCATENATED AND DOWNSHIFTED
/ 6 LOGICAL
/ 6 MAC ,CONT
/ 6 ALU ,DBUS,OR,OR,ZA
/ 6 REG ,GR2,GR2
/ 6 DBUS SNOKE & C&ALU & STATSEPT DBUS0110000000

0408 (4) 1 IS SUBTRACTED FROM THE CONTENTS OF RB9 (THE CARRY INPUT IS SET
/ 6 TO 0 BY STATSEPT CAUSING THIS TO HAPPEN)
/ 6 MAC ,CONT
/ 6 ALU ,DBUS,MOP,SUBS,DB
/ 6 REG ,R9 & DBUS SALU
/ 6 CREG & STATSEPT DBUS0000000000

0409 (5) JUMP TO DSRL1 TAKES PLACE IF ZLSB IS NOT SET
/ 6 MAC ,FALSE,ZLSB,CJP
/ 6 ALU & DBUS SNOKE & ADDRESS DSRL1

040A (6) JUMP SUB TAKES PLACE TO ASDLSG2, THE ROUTINE THAT DETERMINES THE
/ 6 SETTING OF THE STATUS BITS FOR A DOUBLE LENGTH WORD IN GR2 AND GR2+1.
/ 6 MAC ,FALSE,ZERO,CJS
/ 6 ALU & DBUS SNOKE & ADDRESS ASDLSG2

040B (9) RETURN TAKES PLACE.
/ 6 MAC ,FALSE,ZERO,CRTN & ALU & DBUS SNOKE

DOUBLE SHIFT RIGHT ARITHMETIC

TM FS 403

1 MAY 1981

PAGE 36

AMDOS/29 ANDRAM MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EMUL

```

: ENTRY POINT
: *****
043C OFC0B07: MAC ,FALSE,ZERO,CJS
/ C
/ C
: (2) THE CONTENTS OF REGISTER RB+1 (GR2+1) ARE LOADED INTO THE QREG
043D D5B2: MAC ...CONT
/ C
/ C
/ C
: (2A) THE CONTENTS OF GR2 (MS HALF) ARE SHIFTED LEFT SO THAT THE MS BIT GOES
: INTO THE CARRY. THE ALU IS NOT CLOCKED HOWEVER SO THAT THE CONTENTS OF GR2
: ARE UNCHANGED.
043E
/ C
/ C
/ C
: (3) THE CONTENTS OF RB (GR2) AND QREG ARE CONCATENATED AND DOWNSHIFTED
: WITH THE CARRY GOING INTO THE MS BIT..
043F D5B1: MAC ...CONT
/ C
/ C
/ C
: (4) 1 IS SUBTRACTED FROM THE CONTENTS OF ERG (THE CARRY INPUT IS SET
: TO 0 BY STATUS CAUSING THIS TO HAPPEN)
043G
/ C
/ C
/ C
: (5) JUMP TO D5A1 TAKES PLACE IF ZLSB IS NOT SET
: THE MACHINE STATUS REGISTER IS LOADED ACCORDING TO THE CONTENTS OF
: GR2 (THE MOST SIGNIFICANT HALF OF THE DOUBLE LENGTH
: WORD.
043I
/ C
/ C
/ C
: (6) JUMP SUB TAKES PLACE TO ASD1G2, THE ROUTINE THAT DETERMINES THE
: SETTING OF THE STATUS BITS FOR A DOUBLE LENGTH WORD IN GR2 AND GR2+1.
043Z
/ C
/ C
/ C

```

1 MAY 1981

PAGE 37

AMDOS/29 ANDRAM MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EMUL

```

: (9) RETURN TAKES PLACE.
0433
/ C
/ C
/ C
: DOUBLE SHIFT LEFT ARITHMETIC
: *****
0434 D5A1: MAC ...CONT
/ C
/ C
/ C
: (1) THE CONTENTS OF REGISTER RB+1 (GR2+1) ARE LOADED INTO THE QREG
: LOGICAL. THE ORIGINAL CONTENTS ARE PLACED IN ERG.
0435 D5A2: MAC ...CONT
/ C
/ C
/ C
: (2) THE CONTENTS OF RB (GR2) AND QREG ARE CONCATENATED AND UPSHIFTED
: LOGICAL. THE ORIGINAL CONTENTS ARE PLACED IN ERG.
0436
/ C
/ C
/ C
: (3) JUMP SUBROUTINE TO OVERFLOW TAKES PLACE IF THE ZMSD AUXILIARY STATUS
: BIT WAS NOT SET. THE CONTENTS OF IR2 ARE RESTORED.
0437
/ C
/ C
/ C
: (4) 1 IS SUBTRACTED FROM THE CONTENTS OF ERG (THE CARRY INPUT IS SET
: TO 0 BY STATUS CAUSING THIS TO HAPPEN)
0438
/ C
/ C
/ C
: (5) JUMP TO D5A2 TAKES PLACE IF ZLSB IS NOT SET
: THE MACHINE STATUS REGISTER IS LOADED ACCORDING TO THE CONTENTS OF

```

1 MAY 1981

PAGE 38

ANDOS/29 ANDAM MICRO ASSEMBLER, VI.6
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EMSL)

: GR2 (THE MOST SIGNIFICANT HALF OF THE DOUBLE LENGTH
: WORD.

0412A / 6 MAC ,FALSE,ZLSB,CJP
: ALU & DBUS SHOWN & ADDRESS DSI1A2

: (8) JUMP SUB TAKES PLACE TO ASDLSTG2, THE ROUTINE THAT DETERMINES THE
: SETTING OF THE STATUS BITS FOR A DOUBLE LENGTH WORD IN GR2 AND GR2+1.

0413B / 6 MAC ,FALSE,ZZERO,CJS
: ALU & DBUS SHOWN & ADDRESS ASDLSTG2

: (9) RETURN TAKES PLACE.

0413C / 6 MAC ,FALSE,ZZERO,CRTM & ALU & DBUS SHOWN

DOUBLE SHIFT LEFT CYCLIC

ENTRY POINT

0414D OPCODES: MAC ,FALSE,ZZERO,CJS
: ALU & DBUS SHOWN & ADDRESS SEPTM

: (2) THE CONTENTS OF REGISTER RB+1 (GR2+1) ARE LOADED INTO THE QREG

0415E DSI1C2: MAC ,FALSE,ZZERO,CJP
: ALU & DBUS SHOWN & ADDRESS DSI1C2
: REG ,GR2,GR2
: DBUS SHOWN & CIALU & STATUSPT 34001111000000

: (3) THE CONTENTS OF RB (GR2) AND QREG ARE CONCATENATED AND UPSHIFTED

0416F DSI1C1: MAC ,FALSE,ZZERO,CJP
: ALU & DBUS SHOWN & ADDRESS DSI1C1
: REG ,GR2,GR2
: DBUS SHOWN & CIALU & STATUSPT 34001111000000

: (4) 1 IS SUBTRACTED FROM THE CONTENTS OF ER0 (THE CARRY INPUT IS SET
: TO 0 BY STATUSPT CAUSING THIS TO HAPPEN)

0417G / 6 MAC ,FALSE,ZZERO,CJP
: ALU & DBUS SHOWN & ADDRESS DSI1C1
: REG ,GR2,GR2
: DBUS SHOWN & CIALU & STATUSPT 34001111000000

: (5) JUMP TO DSI1C1 TAKES PLACE IF ZLSB IS NOT SET
: THE MACHINE STATUS REGISTER IS LOADED ACCORDING TO THE CONTENTS OF

1 MAY 1981

PAGE 39

ANDOS/29 ANDAM MICRO ASSEMBLER, VI.6
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EMSL)

: GR2 (THE MOST SIGNIFICANT HALF OF THE DOUBLE LENGTH
: WORD.

0417A / 6 MAC ,FALSE,ZLSB,CJP
: ALU & DBUS SHOWN & ADDRESS DSI1C1

: (8) JUMP SUB TAKES PLACE TO ASDLSTG2, THE ROUTINE THAT DETERMINES THE
: SETTING OF THE STATUS BITS FOR A DOUBLE LENGTH WORD IN GR2 AND GR2+1.

0417B / 6 MAC ,FALSE,ZZERO,CJS
: ALU & DBUS SHOWN & ADDRESS ASDLSTG2

: (9) RETURN TAKES PLACE.

0417C / 6 MAC ,FALSE,ZZERO,CRTM & ALU & DBUS SHOWN

DOUBLE SHIFT RIGHT CYCLIC

: (2) THE CONTENTS OF REGISTER RB+1 (GR2+1) ARE LOADED INTO THE QREG

0417D DSI1C2: MAC ,FALSE,ZZERO,CJP
: ALU & DBUS SHOWN & ADDRESS DSI1C2
: REG ,GR2,GR2
: DBUS SHOWN & CIALU & STATUSPT 34001111000000

: (3) THE CONTENTS OF RB (GR2) AND QREG ARE CONCATENATED AND DOWNSHIFTED

0417E DSI1C1: MAC ,FALSE,ZZERO,CJP
: ALU & DBUS SHOWN & ADDRESS DSI1C1
: REG ,GR2,GR2
: DBUS SHOWN & CIALU & STATUSPT 34001111000000

: (4) 1 IS SUBTRACTED FROM THE CONTENTS OF ER0 (THE CARRY INPUT IS SET
: TO 0 BY STATUSPT CAUSING THIS TO HAPPEN)

0417F / 6 MAC ,FALSE,ZZERO,CJP
: ALU & DBUS SHOWN & ADDRESS DSI1C1
: REG ,GR2,GR2
: DBUS SHOWN & CIALU & STATUSPT 34001111000000

: (5) JUMP TO DSI1C1 TAKES PLACE IF ZLSB IS NOT SET
: THE MACHINE STATUS REGISTER IS LOADED ACCORDING TO THE CONTENTS OF

0417G / 6 MAC ,FALSE,ZLSB,CJP
: ALU & DBUS SHOWN & ADDRESS DSI1C1

: (8) JUMP SUB TAKES PLACE TO ASDLSTG2, THE ROUTINE THAT DETERMINES THE
: SETTING OF THE STATUS BITS FOR A DOUBLE LENGTH WORD IN GR2 AND GR2+1.

TN FS 403

1 MAY 1961

PAGE 40

ANDOS/20 ANDASH MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (DMSL)

0476 / 6 MAC .FALSE.,ZERO,CJS
ALU & DBUS SNOKE & ADDRESS ASDLS02

0479 (9) RETURN TAKES PLACE
MAC .FALSE.,ZERO,CRTM & ALU & DBUS SNOKE

SEPTD16 -- SUBROUTINE USED FOR DETERMINING DIRECTION AND NUMBER OF SHIFTS
FOR SINGLE SHIFT COUNT IN REGISTER INSTRUCTIONS.

PLACES MODULUS OF GR2 IN ER9 SETS ER8 AS FOLLOWS:
0 IF ZERO SHIFT
1 IF POSITIVE SHIFT
2 IF NEGATIVE SHIFT
3 IF OVERFLOW.

(1) THE CONTENTS OF GR2 ARE PLACED IN ER9. MICROSTATUS IS LOADED ACCORDING
TO GR2.
SEPTD16: MAC ...CONT & ALU D00...NOP,OR,2A & REG ...GR2,R0 & CKREG
REG ...DBUS & STATSHFT D0000000000000000

(2) RETURN TAKES PLACE IF MICRO Z IS SET INDICATING A ZERO SHIFT. 0 IS PLACED
IN ER8
MAC .TRUE.,MUI,CRTM & ALU ...NOP,AND,AND,DZ & DBUS SALU & REG ...R0
CKREG & STATSHFT D0000000000000000

(2A) 1 IS SUBTRACTED FROM GR2 AND THE RESULT PLACED IN THE QREG.
MAC ...CONT & ALU ...QREG,SUBR,2A & REG ...GR2
STATSHFT D0000000000000000 & CKALU & DBUS SNOKE

(3) QREG CONTENTS ARE MASKED WITH 1111111110000. MICROSTATUS IS LOADED
ACCORDING TO RESULT. NO THE MASK PATTERN NEEDED IS COINCIDENTALLY THE SAME
AS A STATUS AND SHIFT PATTERN THAT WILL LOAD THE MICROSTATUS REGISTER.
MAC ...CONT & ALU D00...DBUS,MOP,AND,AND,DQ & DBUS SDATA
DATINSHT D065520

(4) RETURN TAKES PLACE IF MICROSTATUS N IS SET. THIS IS THE CASE IF THE
CONTENTS OF THE REGISTER INDICATING SHIFT MAGNITUDE AND DIRECTION WAS
POSITIVE AND NOT GREATER THAN 16.
MAC .TRUE.,MUI,CRTM & ALU ...ABUS,MOP,ADD,ADD,DZ & REG ...R0
DBUS SALU & STATSHFT D0010000000000000 & CKREG

(5) GR2 IS NEGATED AND PLACED IN ER9.
MAC ...CONT & ALU ...NOP,SUBS,2A & REG ...GR2,R0 & CKREG

1 MAY 1961

PAGE 41

ANDOS/20 ANDASH MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (DMSL)

/ 6 DBUS SALU & STATSHFT D0010000000000000

(6) 1 IS SUBTRACTED FROM THE CONTENTS OF ER9 AND THE RESULT
PLACED IN QREG.
MAC ...CONT & ALU ...ABUS,QREG,SUBS,2A & REG ...R0 & CKALU
STATSHFT D0000000000000000

(7) CONTENTS OF QREG ARE MASKED WITH 1111111110000 AND THE MICROSTATUS
REGISTER SET ACCORDING TO THE RESULT.
MAC ...CONT & ALU D00...DBUS,MOP,AND,AND,DQ & DBUS SDATA
DATINSHT D065520

(8) RETURN TAKES PLACE IF MICROSTATUS ZERO IS SET. 1 IS ADDED TO THE CONTENTS
OF ER8.
MAC .TRUE.,MUI,CRTM & ALU ...ABUS,MOP,ADD,ADD,DZ & DBUS SALU & CKREG
REG ...R0 & STATSHFT D0010000000000000

(9) RETURN TAKES PLACE. 1 IS ADDED TO ER8.
MAC .FALSE.,ZERO,CRTM & ALU ...ABUS,MOP,ADD,ADD,DZ & DBUS SALU & CKREG
REG ...R0 & STATSHFT D0010000000000000

SEPTD32 -- SUBROUTINE USED FOR DETERMINING DIRECTION AND NUMBER OF SHIFTS
FOR DOUBLE SHIFT COUNT IN REGISTER INSTRUCTIONS.

PLACES MODULUS OF GR2 IN ER9 SETS ER8 AS FOLLOWS:
0 IF ZERO SHIFT
1 IF POSITIVE SHIFT
2 IF NEGATIVE SHIFT
3 IF OVERFLOW.

(1) THE CONTENTS OF GR2 ARE PLACED IN ER9. MICROSTATUS IS LOADED ACCORDING
TO GR2.
SEPTD32: MAC ...CONT & ALU D00...NOP,OR,2A & REG ...GR2,R0 & CKREG
REG ...DBUS & STATSHFT D0000000000000000

(2) RETURN TAKES PLACE IF MICRO Z IS SET INDICATING A ZERO SHIFT. 0 IS PLACED
IN ER8
MAC .TRUE.,MUI,CRTM & ALU ...NOP,AND,AND,DZ & DBUS SALU & REG ...R0
CKREG & STATSHFT D0000000000000000

(2A) 1 IS SUBTRACTED FROM GR2 AND THE RESULT PLACED IN THE QREG.
MAC ...CONT & ALU ...QREG,SUBR,2A & REG ...GR2
STATSHFT D0000000000000000 & CKALU & DBUS SNOKE

(3) QREG CONTENTS ARE MASKED WITH 1111111110000. MICROSTATUS IS LOADED

ANDOS/29 ANDAS MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EASL

85310 / 6
:(3) THE CONTENTS OF INFO ARE DOWNSHIFTED.
MAC ,CONT S ALLU ...RAND.O,OR.ZA & REC ,.2220.2220 & CEALU
DEUS SHONE & STATSWAT #001010000000

0511 / 5
 (4) THE CONTENTS OF INFO ARE DOWNSHIFTED.
 NAC ...CONF & ALU ...RAND,OR,OR,ZA & REC ...ZERO,ZERO & CMLU
 DBUS SNONE & STASHFT #000101000000

: (5) THE CONTENTS OF IIR ARE SHIPPED WITH THE CONTENTS OF ERI. THE INSTRUCTION
: REGISTER IS LOADED.

000000 / 6
;
REC ., PZERO, RZERO, R1 S CKALU S CIREG
HALF .FALSE, IZERO, CRIN S ALD ., ABUS.
NAME, NAME, ON, ON.DZ S DBOUS SALU

```
*****
SHIFT LOGICAL COUNT IN REGISTER
*****

:
:
: ENTRY POINT
:
: ++++++
: ++++++
: ++++++
```

```

00514 OPCODEA:: NAC ,FALSE,TZFRO,CJS 5 ALU 5 DBUS SNOWF & ADDRESS SHPTNDIS

```

```

; (2) JUMP TAKES PLACE TO PLOCGR2 SO THAT THE GR1 FIELD IN THE INSTRUCTION
; REGISTER IS TRANSFERRED TO THE GR2 FIELD SO THAT THE INSTRUCTIONS SLL, SRL, SRA
; SEC.DLL, SRL, SRA, NSLC CAN BE USED. THIS RESULTS IN A LARGE SAVING IN
; CODE ALTHOUGH THERE IS A TIME PENALTY INCURRED BY USING THIS SUBROUTINE.

```

REGISTRATION IS TRANSFERRED TO THE ONE FIELD SO THAT THE INSTRUCTIONS SELECTED,
SEC, DSLL, NSRR, DSRR, NSLC CAN BE USED. THIS RESULTS IN A TARGET SAVING IN
: CODE ALTHOUGH THERE IS A TIME PENALTY INCURRED BY USING THIS SUBROUTINE.
: NAC ,FALSE;77PRO,CJS \$ ALU \$ DUSX SWOFT \$ ADDRESS PEAR10R2
: 0515

MAC 1100001 3 000 00000,000,000,000 3 0000 00000 0 000 00000
 : (4) RETURN TAKES PLACE IF LS BYTE WAS ZERO IMPLYING NO SHIFT.

[illegible]

0518
 (5) INDICATING POSITIVE (I.E. LEFT) SHIFT.
 MAC ,TRUX,ZLSB,CJP & ALU & DBUS SNOWE & ADDRESS 3LL1
 (6) 1 IS SUBTRACTED FROM THE

1 MAY 1991

AM905/29 AM943M MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE INCH SUBSET FOR 1750. (MSL)

[illegible]

```

: SHIFT ARITHMETIC COUNT IN REGISTER
: *****
:
: ENTRY POINT
: *****
:

```

051C OP000803: MAC_PAL5T,2200,CJS & AL0 & D005 SHOME & ADDRESS SEPT01D16

051D

(2) JUMP TAKES PLACE TO PL0R012 SO THAT THE GR1 FIELD IN THE INSTRUCTION REGISTER IS TRANSFERRED TO THE GR2 FIELD SO THAT THE INSTRUCTIONS SLL\$SL,SRA SLL\$M01,M01L,PL0R1 CAN BE USED. THIS RESULTS IN A LARGER SAVING IN CODE ALTHOUGH THERE IS A TIME PENALTY INCURRED BY USING THIS SUBROUTINE.

MAC_PAL5T,2200,CJS & AL0 & D005 SHOME & ADDRESS PL0R1C2

:(3) THE CONTENTS OF REG ARE PASSED THROUGH THE ALU.
MAC ...CONT & ALU ..APUS.MOP.OR.OR.DZ & DBUS SMOKE & REG R8

:(4) RETURN TAKES PLACE IF LS BYTE WAS ZERO INVOLVING NO SHIFT.
MAC TRUE.ZLD.CRM & ALU ..ALUS.MOP.SUPS.SUBS.DZ & REG R8
CERD & STATUS REGISTERED & DBUS LAIN

;(5) JUMP TO SLAI TAKES PLACE IF LS BYTE OF ERG WAS ZERO AFTER SUBTRACTION IN
;(3) INDICATING POSITIVE (IE. LEFT) SHIFT.
MAC TRU, ZLSB, CJP & ALU & DRUG SHOWN & ADDRESS SLAI

0221
;(6) 1 IS SUBTRACTED FROM E88
MAC ...CONT & ALD ...ADUS.MOP,SUP,SUBS,DZ & REC ...RG & DBUS SALT
CENTR & STATIVITY REFERENCE*****
/ 6

1 (7) JUMP TO SBA0 TAKES PLACE IF LS BYTE OF B10 WAS ZERO AFTER THE SUBTRACTION
: MEANING A NEGATIVE SHIFT (IF RIGHT) IS REQUIRED.
MAC TRUS,2,LSB,CJF & ALB & DBO5 SHONE & ADDRESS SBA0
00022

;(6) IN THE EVENT THAT THE IS STILL NONZERO, JUMP TO OVERFLOW TAKES PLACE.

1 MAY 1991

PAGE 44

AMOS/20 ANDAM MICRO ASSEMBLER, V1.0
 EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (ENSL

9523 MAC ...IP & ALU & DEUS SHOWS & ADDRESS OVERFLOW

; SHIFT CYCLIC COUNT IN REGISTER
; *****
; *****

```

: ENTRY POINT
: *****

```

0524 OPCODE6:: MAC .FALSE,TZERO.CJS & ALU & DBUS SNOKE & BADDESS SEPTND16

```

: (2) JUMP TAKES PLACE TO PLGRACR2 SO THAT THE GR1 FIELD IN THE INSTRUCTION
: REGISTER IS TRANSFERRED TO THE GR2 FIELD SO THAT THE INSTRUCTIONS ALL GR2,GR4
: SEC.MILL,GR1,GR2,GR4,BZLC CAN BE USED. THIS RESULTS IN A LARGE SAVING IN
: CODE ALTHOUGH THERE IS A TIME PENALTY INCURRED BY USING THIS SUMMATION.

```

06325 :
MAC , FALSE,TZERO,CJS & ALV & DUS SNOWY & BADDRESS PLAR18RZ
:(S) THE CONTENTS OF ERG ARE PASSED THROUGH THE ALV.
: COME ALTHOUGH THERE IS A TIME PENALTY INCURRED BY USING THIS SUBROUTINE.

0526 :
 (3) THE CONTENTS OF ZED ARE PASSED THROUGH THE ALB.
 WAC .., ABUS, NOT, OR, OR, DZ & BARS SMOKE & RES RE
 ;

0527 :
 ((4) RETURN TACTS PLACE IF LS OPTI WAS ZERO IMPLYING NO SHIPT.
 MAC ,TRUE,ZLS,CYPA & ALG ..ABUS,NOT,S9BS,S9BS,DZ RES RE

1 / 6
 2
 3
 4
 5
 6
 7
 8
 9
 10
 11
 12
 13
 14
 15
 16
 17
 18
 19
 20
 21
 22
 23
 24
 25
 26
 27
 28
 29
 30
 31
 32
 33
 34
 35
 36
 37
 38
 39
 40
 41
 42
 43
 44
 45
 46
 47
 48
 49
 50
 51
 52
 53
 54
 55
 56
 57
 58
 59
 60
 61
 62
 63
 64
 65
 66
 67
 68
 69
 70
 71
 72
 73
 74
 75
 76
 77
 78
 79
 80
 81
 82
 83
 84
 85
 86
 87
 88
 89
 90
 91
 92
 93
 94
 95
 96
 97
 98
 99
 100
 101
 102
 103
 104
 105
 106
 107
 108
 109
 110
 111
 112
 113
 114
 115
 116
 117
 118
 119
 120
 121
 122
 123
 124
 125
 126
 127
 128
 129
 130
 131
 132
 133
 134
 135
 136
 137
 138
 139
 140
 141
 142
 143
 144
 145
 146
 147
 148
 149
 150
 151
 152
 153
 154
 155
 156
 157
 158
 159
 160
 161
 162
 163
 164
 165
 166
 167
 168
 169
 170
 171
 172
 173
 174
 175
 176
 177
 178
 179
 180
 181
 182
 183
 184
 185
 186
 187
 188
 189
 190
 191
 192
 193
 194
 195
 196
 197
 198
 199
 200
 201
 202
 203
 204
 205
 206
 207
 208
 209
 210
 211
 212
 213
 214
 215
 216
 217
 218
 219
 220
 221
 222
 223
 224
 225
 226
 227
 228
 229
 230
 231
 232
 233
 234
 235
 236
 237
 238
 239
 240
 241
 242
 243
 244
 245
 246
 247
 248
 249
 250
 251
 252
 253
 254
 255
 256
 257
 258
 259
 260
 261
 262
 263
 264
 265
 266
 267
 268
 269
 270
 271
 272
 273
 274
 275
 276
 277
 278
 279
 280
 281
 282
 283
 284
 285
 286
 287
 288
 289
 290
 291
 292
 293
 294
 295
 296
 297
 298
 299
 300
 301
 302
 303
 304
 305
 306
 307
 308
 309
 310
 311
 312
 313
 314
 315
 316
 317
 318
 319
 320
 321
 322
 323
 324
 325
 326
 327
 328
 329
 330
 331
 332
 333
 334
 335
 336
 337
 338
 339
 340
 341
 342
 343
 344
 345
 346
 347
 348
 349
 350
 351
 352
 353
 354
 355
 356
 357
 358
 359
 360
 361
 362
 363
 364
 365
 366
 367
 368
 369
 370
 371
 372
 373
 374
 375
 376
 377
 378
 379
 380
 381
 382
 383
 384
 385
 386
 387
 388
 389
 390
 391
 392
 393
 394
 395
 396
 397
 398
 399
 400
 401
 402
 403
 404
 405
 406
 407
 408
 409
 410
 411
 412
 413
 414
 415
 416
 417
 418
 419
 420
 421
 422
 423
 424
 425
 426
 427
 428
 429
 430
 431
 432
 433
 434
 435
 436
 437
 438
 439
 440
 441
 442
 443
 444
 445
 446
 447
 448
 449
 450
 451
 452
 453
 454
 455
 456
 457
 458
 459
 460
 461
 462
 463
 464
 465
 466
 467
 468
 469
 470
 471
 472
 473
 474
 475
 476
 477
 478
 479
 480
 481
 482
 483
 484
 485
 486
 487
 488
 489
 490
 491
 492
 493
 494
 495
 496
 497
 498
 499
 500
 501
 502
 503
 504
 505
 506
 507
 508
 509
 510
 511
 512
 513
 514
 515
 516
 517
 518
 519
 520
 521
 522
 523
 524
 525

MAC .TRUE,ZLSB,CJP & ALV & DBMS SHOWS & ADDRESS 31C1

0629
:(6) 1 IS SUBTRACTED FROM R88
MAC ...CONT & ALU ,ADDS.MOP.SUBS.SUMS.DZ & RDG DO 6 MORE SALS
GENRS & STATYSTT *****
/ &

;(7) JUMP TO SBC1 TAKES PLACE IF LS BYTE OF E28 WAS ZERO AFTER THE SUBTRACTION
; MEANING A NEGATIVE SHIFT (IE RIGHT) IS REQUIRED.
MAC ,TR2E,2LSB,CJP & ALU & D28S SNOOS & ADDRESS SBC1
0024

02329

! (6) IN THE EVENT THAT ERG IS STILL NONZERO, JUMP TO OVERFLOW TAKES PLACE.
MAC ...JP C A2U C D99S SMOOT & BADADDRESS OVERFLOW

DOUBLE SHIFT LOGICAL COUNT IN REGISTER

ENTRY POINT

1 MAY 1961

Page 48

ANDOS/29 ANDASII MICRO ASSEMBLER, V1.0
 TRANSLATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EML)

```

0040 (5) JUMP TO DSIC2 TAKES PLACE IF 15 BYTE OF EBB WAS ZERO AFTER SUBTRACTION IN
      (3) INDICATING POSITIVE (IE. LEFT) SHIFT,
      MAC ,TR05,ZLSB,CJP & ALU & DBUS SMOKE & ADDRESS DSIC2

0041 (6) 1 IS SUBTRACTED FROM EBB
      MAC ,TR05,CONF & ALU
      ZERO & SLTSHFT 0000000000000000

0042 (7) JUMP TO DSIC2 TAKES PLACE IF 15 BYTE OF EBB WAS ZERO AFTER THE SUBTRACTION
      MEANING A NEGATIVE SHIFT (IE RIGHT) IS REQUIRED,
      MAC ,TR05,ZLSB,CJP & ALU & DBUS SMOKE & ADDRESS DSIC2

0043 (8) IN THE EVENT THAT EBB IS STILL NONZERO, JUMP TO OVERTFLOW TAKES PLACE.
      MAC ,CJP & ALU & DBUS SMOKE & ADDRESS OVERTFLOW

```

SINGLE PRECISION LOAD

ENTRY POINT FOR REGISTER ADDRESSING

:(1) THE CONTENTS OF GR2 (RS) ARE PLACED IN GR1 (RA)
0544 OPCODE1: MAC FALSE, ZERO, CPM & ALU, BUS, RAM, OR, OR, ZA
/ C RS GR1, GR2 & CEALU & DRUG SMOKE & CANNISAL CERO
/ 6 STATEP B999999999999999

ENTRY POINT FOR ISP ADDRESSING

;(1) JUMP SUB TAKES PLACE TO OPISB
0545 0PC08021: MAC FALSE 0280-CIS 5 AIN 6 DRSS SNOWP 6 RANDESS OPISB

```

;(2) THE DERIVED OPERAND IS PLACED IN GR1. MACHINE STATUS IS LOADED AND
;RETURN TAKES PLACE.

```

0946 / 6
/ 6
RETURN YAKS PAGE.
MC , FALSF, TZED, CETH & ALD . 380, A BUS, RAMP, OR, DZ & REC ., GR1., R5
CALD & DUS SHONY & STATSHY 380000010000
CARTSEL CZHO

ENTRY POINT FOR ISN ADDRESSING

1 MAY 1981

PAGE 49

ANDOS/29 ANDAS MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (ENSL

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040

```

ENTRY POINT FOR BASE RELATIVE ADDRESSING

;(1) JUMP SUB TO OFB1

0549 0F020001: MAC ,FALS,P200,CJS & ALO & D005 SHONE & MADRESS 0791
 (1) JUMP SUB TO 0791
 (2) THE DERIVED OPRAND IN XPS IS PLACED IN TR2, MACHINE STATUS IS LOADED
 AND RETURN TAPES PLACED.

0506A / 6 MAC FALSE, TZERO, CRYN & ALU , 300, ABUS, RAMP, OR, OR, 02
/ 6 REC 0010, TZERO, .15 & CVALD & DBUS SMOKE & CARYSEL CZERO
/ 6 STATSHFT 3000000000000000

ENTRY POINT FOR BASE RELATIVE INDEXED ADDRESSING

;(1) JUMP SUB TAKES PLACE TO OPEN1
954B XOPCD499: MAC .PAISI,TZER0,CJS & ALB & DRUS SNOWE & ADDRESS OPEN1

```

: (2) THE DERIVED OPERAND IS PLACED IN IN2. THE MACHINE STATUS REGISTER IS
: LOADED ACCORDING TO THE ALG OUTPUT. RETURN TAKES PLACE.

```

STAT / 6 / 6
 REG #010, AZERO, NS & CALU & DROS SMOKE & CARRTEL CZERO
 STATSTP 1000000010000
 MAC #AL31, ZZERO, CHYN & ALU 100, ABUS, HANY, OR, 01, 02
 REG #010, AZERO, NS & CALU & DROS SMOKE & CARRTEL CZERO

ENTRY POINT FOR DIRECT OR DIRECT IMPLIED ADDRESSING

```
;(1) JUMP SUB TO OPD11
```

;(2) THE DERIVED OPERAND IN ERS IS LOADED INTO CR1 (RA), THE MACHINE STATES
; REGISTER IS LOADED ACCORDING TO ITS VALUE. RETURN TAKES PLACE.
MAC .FALSE.TERMO.CRMV 6 ALD .JUG.ABVS.RANT.0R.0R.0R
9542
9543

1 MAY 1981

PAGE 56

IMDS/20 ANDAM MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EMSL)

/ 6 REG ,GRI ,R5 & CEALU & DBUS SNOKE & CARRISSEL CZERO
/ 6 STATSHFT 8400000010000

ENTRY POINT FOR INDIRECT AND INDIRECT INDEED ADDRESSING

(1) JUMP SUB TO OPD11

OPCODER4:: MAC ,FALSE,TZERO,CJS & ALU & DBUS SNOKE & ADDRESS OPD11

(2) THE DERIVED OPERAND IN R5 IS PLACED IN GRI. THE MACHINE STATUS REGISTER IS LOADED AND RETURN TAKES PLACE.

MAC ,FALSE,TZERO,CRTM & ALU ,R50,ANDUS,RAMP,OR,OR,DZ

REG ,GRI ,R5 & CEALU & DBUS SNOKE & CARRISSEL CZERO

STATSHFT 8400000010000

ENTRY POINT FOR IMMEDIATE ADDRESSING

(1) JUMP SUB TO RETD10D

OPCODER5:: MAC ,FALSE,TZERO,CJS & ALU & DBUS SNOKE & ADDRESS RETD10D

(2) THE DERIVED OPERAND IN R5 IS LOADED INTO GRI. THE MACHINE STATUS REGISTER IS LOADED ACCORDING TO ITS VALUE. RETURN TAKES PLACE.

MAC ,FALSE,TZERO,CRTM & ALU ,R50,ANDUS,RAMP,OR,OR,DZ

REG ,GRI ,R5 & CEALU & DBUS SNOKE & CARRISSEL CZERO

STATSHFT 8400000010000

LOAD FROM UPPER BYTE

ENTRY POINT FOR DIRECT OR DIRECT INDEED ADDRESSING

(1) JUMP SUB TO OPD12

OPCODER6:: MAC ,FALSE,TZERO,CJS & ALU & DBUS SNOKE & ADDRESS OPD12

(2) THE DERIVED OPERAND IN R5 IS SHIPPED WITH THE CONTENTS OF GRI.

MAC ,FALSE,TZERO,PUSH & ALU ,ANDUS,RAMA,OR,OR,DZ & REG ,GRI,GRI,R5

DBUS SNOKE & CEALU & CARRISSEL CZERO

STATSHFT 8400000010000

(3) THE CONTENTS OF GRI ARE SHIPPED 8 PLACES DOWN.

1 MAY 1981

PAGE 51

IMDS/20 ANDAM MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EMSL)

0555 / 6 MAC ,FALSE,COUNT,RECT & ALU ,RAMP,OR,OR,DZ
CEALU & DBUS SNOKE & REG ,GRI,GRI & STATSHFT 8400000010000

0556 / 6 (3A) A MASK (1111111100000000) IS PLACED IN QREG

MAC ,FALSE,COUNT,RECT & ALU ,DBUS,CRTM,OR,OR,DZ

DBUS SNOKE & DATINSHFT 8400000010000

(4) THE CONTENTS OF R55 (ORIGINALLY THE CONTENTS OF GRI) ARE MASKED BY

THE CONTENTS OF Q REG (1111111100000000) AND PLACED IN QREG.

MAC ,FALSE,COUNT,RECT & ALU ,ANDUS,QREG,AND,AND,DZ & REG ,R55 & CEALU

DBUS SNOKE

(5) THE CONTENTS OF QREG (MS HALF OF ORIGINAL R5) ARE ORED WITH THE

CONTENTS OF GRI. THE RESULT IS PLACED IN GRI AND THE STATUS REGISTER

LOADED.

MAC ,FALSE,TZERO,CRTM & ALU ,R50,RAMP,OR,OR,DZ

REG ,GRI,GRI & CEALU & DBUS SNOKE & CARRISSEL CZERO

STATSHFT 8400000010000

ENTRY POINT FOR INDIRECT AND INDIRECT INDEED ADDRESSING

(1) JUMP SUB TO OPD11

OPCODER7:: MAC ,FALSE,TZERO,CJS & ALU & DBUS SNOKE & ADDRESS OPD11

(2) JUMP TO START OF SEQUENCE FOR DIRECT ADDRESSING.

MAC ,JP & ALU & DBUS SNOKE & ADDRESS LD01

LOAD FROM LOWER BYTE

ENTRY POINT FOR DIRECT OR DIRECT INDEED ADDRESSING

(1) JUMP SUB TO OPD11

OPCODER8:: MAC ,FALSE,TZERO,CJS & ALU & DBUS SNOKE & ADDRESS OPD11

(2) THE DERIVED OPERAND IS PLACED IN THE QREG

MAC ,FALSE,COUNT,RECT & ALU ,ANDUS,QREG,OR,OR,DZ & REG ,R55 & CEALU

DBUS SNOKE

TM FS 403

TM FS 403

1 MAY 1981

PAGE 52

AMDOS/20 ANDASM MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (ENSL)

```

(3) THE CONTENTS OF THE Q REG ARE MASKED BY (0000000011111111) AND THE
RESULT LEFT IN QREG.
0559 / 6 MAC ...CONT & ALU ...DBUS QREG.AND.AND.DQ & CCLAU
      / 6 DBUS SDATA & DATINSET DQ255
(4) THE CONTENTS OF QRI ARE MASKED BY (1111111000000000) AND THE RESULT
LEFT IN QRI.
055E / 6 MAC ...CONT & ALU ...DBUS RAMP.AND.AND.DA & CCLAU
      / 6 DBUS SDATA & DATINSET DQ5286
(5) THE CONTENTS OF THE QREG ARE ORED WITH THE CONTENTS OF QRI AND THE
RESULT LEFT IN QRI.
055F / 6 MAC ...FALSE.TZERO.CRTM & ALU ...DBUS ...RAMP.OR.OR.AQ
      / 6 REG ...QRI.QRI & CCLAU & STATSHFT DQ00000010000
      / 6 CARNISEL CZERO
ENTRY POINT FOR INDIRECT OR INDIRECT INDEXED ADDRESSING
*****
(1) JUMP SUB TO OPIH
0560 / 6 OPCODE7C1: MAC ...FALSE.TZERO.CJS & ALU & DBUS SNONE & ADDRESS OPIH
(2) JUMP TO CODE FOR DIRECT ADDRESSING
0561 / 6 MAC ...JP & ALU & ADDRESS LL91 & DBUS SNONE

```

LOAD STATUS

ENTRY POINT FOR INDIRECT AND INDIRECT INDEXED ADDRESSING

```

(1) JUMP SUB TO OPIK
0562 / 6 OPCODE7C1: MAC ...FALSE.TZERO.CJS & ALU & DBUS SNONE & ADDRESS OPIK
(2) JUMP TO LOSTAI
0563 / 6 MAC ...JP & ALU & DBUS SNONE & ADDRESS LOSTAI
ENTRY POINT FOR DIRECT AND DIRECT INDEXED ADDRESSING
*****

```

1 MAY 1981

PAGE 53

AMDOS/20 ANDASM MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (ENSL)

```

(1) JUMP SUB TO OPI3
0564 / 6 OPCODE7D1: MAC ...FALSE.TZERO.CJS & ALU & DBUS SNONE & ADDRESS OPI3
(2) THE INTERRUPT MASK (DERIVED OPERAND WORD 1) IS INVERTED AND LOADED INTO QREG.
0565 / 6 LOSTAI: MAC ...CONT & ALU ...ABUS.QREG.KINOR.DZ & REG ....R5 & DBUS SNONE
      / 6 CCLAU
( ) THE CONTENTS OF THE QREG ARE LOADED INTO R15, JUMP SUB TAKES PLACE TO
MIMICINT.
0566 / 6 MAC ...FALSE.TZERO.CJS & ALU ...MOP.OR.OR.ZQ & DBUS SALU & REG ....R15
      / 6 CREG & ADDRESS MIMICINT
(2) THE SECOND WORD OF THE DERIVED OPERAND IS LOADED INTO THE STATUS REGISTER
0567 / 6 MAC ...CONT & ALU & DBUS SREG & REG ....R6 & ENAUGE & AUICE DQ100
(2A) THE LAST SIGNIFICANT FOUR BITS OF THE STATUS REGISTER ARE LOADED.
0568 / 6 MAC ...CONT & ALU ...R6, 5 REG ....R6 & DBUS SREG & STATSHFT DQ000000000000
(3) THE THIRD WORD OF THE DERIVED OPERAND IS LOADED FIRST INTO THE Q REG.
0569 / 6 MAC ...CONT & ALU ...ABUS.QREG.OR.OR.DZ & REG ....R7 & CCLAU
      / 6 DBUS SNONE
(4) THE THIRD WORD IN THE QREG IS NOW LOADED INTO THE PROGRAM COUNTER (PC).
RETURN TAKES PLACE.
056A / 6 MAC ...FALSE.TZERO.CRTM & ALU ...MOP.OR.OR.ZQ & REG ....R8 & CREG
      / 6 DBUS SALU

```

LOAD MULTIPLE REGISTERS

ENTRY POINT FOR DIRECT OR DIRECT INDEXED ADDRESSING

```

(1) JUMP TAKES PLACE TO PTDIAD WHICH IS A SUBROUTINE THAT FETCHES THE
CORRECT ADDRESS OF A DIRECT OR DIRECT INDEXED OPERAND.
THE ADDRESS OF THE OPERAND IS PLACED IN R2.
056B / 6 OPCODE891: MAC ...FALSE.TZERO.CJS & ALU & DBUS SNONE & ADDRESS PTDIAD
(2) A MASK (0000000011100000) IS PLACED IN QREG.
056C / 6 MAC ...CONT & ALU ...DBUS.QREG.OR.OR.DZ & CCLAU
      / 6 DBUS SDATA & DATINSET DQ240
(3) THE INSTRUCTION IN R1 IS MASKED AND LEFT IN THE QREG.

```

1 MAY 1981

PAGE 54

ANDOS/29 ANDASH MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (BMSL)

0560 / 5 MAC ...CONT & ALU ...ABUS.QREG,AND,AND,DQ & REG R1 & CIALU
DBUS SALU

0561 LMI (4) LOADING OF THE INSTRUCTION REGISTER IS CARRIED OUT...

0562 / 5 THE MASKED INSTRUCTION IS ALSO PLACED IN R1
MACP ...CONT & ALU ...NOP,OR,OR,DQ & DBUS SALU
CRAFT & REG R1

0563 / 5 (5) JUMP SUB TAKES PLACE TO OPERAND.

0564 / 5 MAC ...FALSE,TZERO,CJS & ALU & DBUS SNOKE & ADDRESS OPERAND

0565 / 5 (6) THE CONTENTS OF R1 ARE PLACED IN R2.

0566 / 5 MAC ...CONT & ALU ...ABUS.RAMP,OR,OR,D2 & CIALU & REG ...R2,R5
DBUS SNOKE

0567 / 5 (7) THE CONTENTS OF R1 ARE PLACED IN QREG

0568 / 5 MAC ...CONT & ALU ...ABUS.QREG,OR,OR,D2 & CIALU & REG R1
DBUS SNOKE

0569 / 5 (8) THE CONTENTS OF THE Q REGISTER ARE MASKED BY (0000000011110000) BUT NOT
LOADED.

0570 / 5 MAC ...CONT & ALU ...DBUS.NOP,AND,AND,DQ & DBUS SDATA & DATINSRT D6240

0571 / 5 (9) 1 IS ADDED TO THE CONTENTS OF R2. RETURN TAKES PLACE IF ZLSB WAS SET BY
THE PREVIOUS INSTRUCTION.

0572 / 5 MAC ...TRUE,ZLSB,CRTM & ALU ...ABUS.NOP,ADD,ADD,D2 & REG R2 & CREG
DBUS SALU & STATSHFT B001000000000000

0573 / 5 (10) 15 IS SUBTRACTED FROM THE CONTENTS OF QREG.

0574 / 5 THIS MEANS THAT THE M FIELD IS DECREMENTED AND THE R2 FIELD INCREMENTED.

0575 / 5 MAC ...CONT & ALU ...DBUS.QREG,SUB,SUB,DQ & CIALU
DBUS SDATA & DATINSRT D614

0576 / 5 (11) JUMP TAKES PLACE TO LMI.

0577 / 5 MAC ...JP & ALU & DBUS SNOKE & ADDRESS LMI

0578 / 5 EXCHANGE BITES IN REGISTER

0579 / 5

0580 / 5 ENTRY POINT

0581 / 5

1 MAY 1981

PAGE 55

ANDOS/29 ANDASH MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (BMSL)

0576 / 5 OFCODECT: MAC ...FALSE,TZERO,PUSH & ALU & DBUS SNOKE
DATINSRT D66

0577 / 5 (12) THE CONTENTS OF R1 ARE SHIFTED 8 PLACES DOWN CYCLIC

0578 / 5 MAC ...FALSE,COUNT,RECT & ALU ...RAMP,OR,OR,ZA & CIALU & DBUS SNOKE
REG ...R1,R1 & STATSHFT B00101000000

0579 / 5 (13) THE MACHINERY STATUS REGISTER IS LOADED AND RETURN TAKES PLACE

0580 / 5 MAC ...FALSE,TZERO,CRTM & ALU ...RAMP,OR,OR,ZA & REG ...R1,R1
STATSHFT B00000010000 & CARRYSEL CZERO

EXCHANGE WORDS IN REGISTER

0581 / 5

0582 / 5 ENTRY POINT

0583 / 5

0584 / 5 (14) THE CONTENTS OF R1 (R2) ARE PLACED TEMPORARILY IN Q REG

0585 / 5 OFCODECT: MAC ...CONT & ALU ...QREG,OR,OR,ZA & REG ...R2 & CIALU & DBUS SNOKE

0586 / 5 (15) THE CONTENTS OF R1 (R2) ARE PLACED IN R1 (R2).

0587 / 5 (16) THE CONTENTS OF THE QREG ARE PLACED IN R1 (R2) THE STATUS REGISTER

0588 / 5 IS LOADED AND RETURN TAKES PLACE.

0589 / 5 MAC ...FALSE,TZERO,CRTM & ALU ...RAMP,OR,OR,ZA & REG ...R1 & CIALU
DBUS SNOKE & STATSHFT B00000010000 & CARRYSEL CZERO

SINGLE PRECISION STORE

0590 / 5

0591 / 5 ENTRY POINT FOR BASE RELATIVE ADDRESSING

0592 / 5 OFCODECT: MAC ...CONT & ALU ...NOP,OR,OR,ZA & REG ...R1,R5 & CREG
DBUS SALU

0593 / 5 (17) JUMP TO APPROPRIATE OPERAND SEND ROUTINE

0594 / 5 MAC ...JP & ALU & DBUS SNOKE & ADDRESS OSB1

TM FS 403

TNI FS 403

1 MAY 1981

PAGE 56

AMDOS/29 AMBUSH MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EMSL)

ENTRY POINT FOR BASE RELATIVE INDEXED ADDRESSING

#578 OP00402: MAC ...CONT & ALU ...NOP,OR,2A & REG ,B019,,R2ZERO,R5 & CREG
/ & DBUS SALU#577 (2) JUMP TO APPROPRIATE OPERAND SEND ROUTINE
MAC ...JP & ALU & DBUS SNOKE & BADDRESS OSD11

ENTRY POINT FOR DIRECT AND INDIRECT INDEXED ADDRESSING

#580 OP00190: MAC ...CONT & ALU ...NOP,OR,OR,2A & REG ...C011,R5 & CREG
/ & DBUS SALU#581 (2) JUMP TO OPERAND SEND ROUTINE
MAC ...JP & ALU & DBUS SNOKE & BADDRESS OSD11#582 OP00194: MAC ...CONT & ALU ...NOP,OR,OR,2A & REG ...C011,R5 & CREG
/ & DBUS SALU#583 (2) JUMP TO APPROPRIATE OPERAND SEND ROUTINE
MAC ...JP & ALU & DBUS SNOKE & BADDRESS OSD11STORE A NON NEGATIVE CONSTANT

ENTRY POINT FOR DIRECT AND INDIRECT INDEXED ADDRESSING

#584 (1) JUMP SUB TO STC1 -- STC1 TAKES THE CONSTANT FROM THE FIELD WITHIN THE
INSTRUCTION AND PLACES IT IN R5
OP00191: MAC ,FALSE,TZERO,CJS & ALU & DBUS SNOKE & BADDRESS STC1#585 (2) JUMP TO THE APPROPRIATE OPERAND SEND ROUTINE (OSD11)
MAC ...JP & DBUS SNOKE & ALU & BADDRESS OSD11ENTRY POINT FOR INDIRECT AND INDIRECT INDEXED ADDRESSING

1 MAY 1981

PAGE 57

AMDOS/29 AMBUSH MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EMSL)#586 OP00192: MAC ,FALSE,TZERO,CJS & ALU & DBUS SNOKE & BADDRESS STC1
/ & DBUS SALU#587 (2) JUMP TO APPROPRIATE OPERAND SEND ROUTINE. (OSD11)
MAC ...JP & ALU & DBUS SNOKE & BADDRESS OSD11

(1) THE CONTENTS OF IR0 ARE PLACED IN R5

#588 STC1: MAC ...CONT & ALU ...NOP,OR,OR,2A & REG ...R2ZERO,R5 & CREG
/ & DBUS SALU#589 (2) THE INSTRUCTION IN R1 IS SHIFTED DOWN AND PLACED IN IR0
MAC ...CONT & ALU ...AND,RAND,OR,OR,2Z & REG ...R2ZERO,R1 & CVALU
/ & DBUS SNOKE & STATSHFT B0000000000000000#58A (3) THE CONTENTS OF IR0 ARE DOWNSHIFTED
MAC ...CONT & ALU ...RAND,OR,OR,2A & CVALU & REG ...R2ZERO,R2ZERO
/ & DBUS SNOKE & STATSHFT B0200000000000000#58B (4) THE CONTENTS OF IR0 ARE DOWNSHIFTED
MAC ...CONT & ALU ...RAND,OR,OR,2A & CVALU & REG ...R2ZERO,R2ZERO
/ & DBUS SNOKE & STATSHFT B0000000000000000#58C (5) THE CONTENTS OF IR0 ARE MASKED BY 00000000001111 AND DOWNSHIFTED.
MAC ...CONT & ALU ...DBUS,RAND,AND,AND,DA & CVALU & REG ...R2ZERO,R2ZERO
/ & DBUS SNOKE & DATINERT D031#58D (6) THE CONTENTS OF IR0 ARE SWAPPED WITH THE CONTENTS OF R5. RETURN
TAKES PLACE.#58D MAC ,FALSE,TZERO,CRTN & ALU ...AND,RAND,OR,OR,2A & CVALU & CREG
/ & REG ...R2ZERO,R2ZERO,R5 & DBUS SALUSTORE REGISTER THROUGH MASK
*****ENTRY POINT FOR DIRECT AND INDIRECT INDEXED ADDRESSING
*****(1) THE CONTENTS OF REGISTER CR1 (R4) ARE ADDED WITH THE CONTENTS OF
REGISTER CR1+1 AND THE RESULTS PLACED IN R5.

#58E OP00197: MAC ...JP & ALU ...NOP,AND,AND,AB & REG ,B001,CR1,CR1,R5

IN ES 113

1 MAY 1981

PAGE 60

AMDOS/20 AMDASH MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EMSL)

```

(3) THE CONTENTS OF CH1 (RA) ARE MASKED BY (0000000011111111) AND PLACED
IN QREG
MAC ...CONT & ALU ...DBUS.QREG.AND.AND.DA & REG ...CH1 & CVALU
/ & DBUS.SDATA & DATINSTR D0255
(4) THE CONTENTS OF THE QREG ARE OREG WITH THE CONTENTS OF R05 AND THE
RESULT LEFT IN R05. RETURN TAKES PLACE.
MAC ...FALSE.TZERO.CTNM & ALU ...ABUS.QREG.OR.OR.DQ & REG ...R5 & CREG
/ & DBUS.SALU
ENTRY POINT FOR INDIRECT OR INDIRECT INDEXED ADDRESSING
*****
(1) JUMP SUB TO OFI11
OPCODE93:: MAC ...FALSE.TZERO.CJS & ALU & DBUS.SNONE & BADDRESS OFI11
(2) JUMP SUB TO SLB1
MAC ...FALSE.TZERO.CJS & ALU & DBUS.SNONE & BADDRESS SLB1
(3) JUMP TO OSGENPUR
MAC ...JP & ALU & DBUS.SNONE & BADDRESS OSGENPUR

```

STORE MULTIPLE REGISTERS
*****ENTRY POINT FOR DIRECT OR DIRECT INDEXED ADDRESSING

```

(1) JUMP TAKES PLACE TO FETDIADD WHICH IS A SUBROUTINE THAT FETCHES THE
CORRECT ADDRESS AND A DIRECT OR DIRECT INDEXED OPERAND.
THE ADDRESS OF THE OPERAND IS PLACED IN R02.
OPCODE99:: MAC ...FALSE.TZERO.CJS & ALU & DBUS.SNONE & BADDRESS FETDIADD
(2) A MASK (0000000011100000) IS PLACED IN QREG.
MAC ...CONT & ALU ...DBUS.QREG.OR.OR.DQ & CVALU
/ & DBUS.SDATA & DATINSTR D0240
(3) THE INSTRUCTION IN R01 IS MASKED AND LEFT IN THE QREG.
MAC ...CONT & ALU ...ABUS.QREG.AND.AND.DQ & REG ...R1 & CVALU
/ & DBUS.SALU

```

1 MAY 1981

PAGE 61

AMDOS/20 AMDASH MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EMSL)

```

(4) LOADING OF THE INSTRUCTION REGISTER TAKES PLACE.
OPCODE91:: MAC ...CONT & ALU ...NOP.OR.OR.ZA & DBUS.SALU
/ & CREG & REG ...R1
(5) THE CONTENTS OF R02 IS PLACED IN R05. JUMP SUB TAKES PLACE TO
OSGENPUR.
MAC ...FALSE.TZERO.CJS & ALU ...NOP.OR.OR.ZA & REG ...R02.R5 & DBUS.SALU
/ & CREG & BADDRESS OSGENPUR
(5A) THE CONTENTS OF R01 ARE PLACED IN THE QREG.
MAC ...CONT & ALU ...ABUS.QREG.OR.OR.DQ & CVALU & REG ...R1
/ & DBUS.SNONE
(5A) THE CONTENTS OF THE QREG ARE MASKED BY (0000000011100000) BUT NOT LOADED.
MAC ...CONT & ALU ...DBUS.QREG.AND.AND.DQ & DBUS.SDATA & DATINSTR D0240
(6) 1 IS ADDED TO THE CONTENTS OF R02. RETURN TAKES PLACE IF
THE PREVIOUS OPERATION SET Z150.
MAC ...TRUE.Z150.CTNM & ALU ...ABUS.NOP.ADD.ADD.DQ & REG ...R2 & CREG
/ & DBUS.SALU & STATSHFT D0010000000000000
(7) 15 IS SUBTRACTED FROM THE CONTENTS OF QREG.
THIS MEANS THAT THE M FIELD IS DECREMENTED AND THE R02 FIELD INCREMENTED.
MAC ...CONT & ALU ...DBUS.QREG.SUBR.SUBR.DQ & CVALU
/ & DBUS.SDATA & DATINSTR D014
(8) JUMP TAKES PLACE TO ST01
MAC ...JP & ALU & DBUS.SNONE & BADDRESS ST01
MOVE MULTIPLE WORDS. MEMORY TO MEMORY.
*****
ENTRY POINT
*****
(1) THE CONTENTS OF RA+1 ARE PASSED THROUGH THE ALU
OPCODE93:: MAC ...CONT & ALU ...NOP.OR.OR.ZA & DBUS.SNONE
/ & STATSHFT D0000000010000 & REG ...R01...R01
(2) RETURN TAKES PLACE IF RA WAS ZERO IN THE PREVIOUS INSTRUCTION.
MAC ...TRUF.MUI.CTNM & ALU & DBUS.SNONE & STATSHFT D0000000000100

```


TM FS 403

1 MAY 1961

PAGE 64

ANDOS/29 ANDASH MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (DMSL)

MAC ...JP & ALU & DBUS SMOKE & ADDRESS PUSM1

05C5

POP MULTIPLE REGISTERS OFF STACK
.....(1) THE NUMBER 15 IS LOADED INTO THE INSTRUCTION REGISTER
OPCODES: MACP ...CONT & ALU & DBUS SMOKE & DATINSRT D015

05C6

(3) THE CONTENTS OF IR15 (THE STACK POINTER) ARE PLACED
IN IR2.

05C7

MAC ...CONT & ALU ...NOP,OR,OR,2A & REG ...0R2,R2 & CEREQ
/ & DBUS SMOKE

05C8

(4) THE CONTENTS OF IR1 ARE SHIFTED ONE PLACE RIGHT AND PLACED IN IR15.
MAC ...CONT & ALU ...AND,OR,OR,2A & REG ...0R2,R1 & CEALU
/ & DBUS SMOKE & STATSHFT D0000000000000000

05C9

(5) THE CONTENTS OF IR15 ARE SHIFTED ONE PLACE RIGHT.
MAC ...CONT & ALU ...AND,OR,OR,2A & REG ...0R2,R2 & CEALU
/ & DBUS SMOKE & STATSHFT D0000000000000000

05CA

(6) THE CONTENTS OF IR15 ARE SHIFTED ONE PLACE RIGHT.
MAC ...CONT & ALU ...AND,OR,OR,2A & REG ...0R2,R2 & CEALU
/ & DBUS SMOKE & STATSHFT D0000000000000000

05CB

(7) THE CONTENTS OF IR15 ARE MASKED BY 0000000000001111 AND SHIFTED ONE
PLACE RIGHT. THE VALUE OF N NOW OCCUPIES THE LEAST SIG FOUR BITS
OF THE WORD. THE REST BEING ZERO.
MAC ...CONT & ALU ...AND,AND,AND,2A & REG ...0R2,R2 & CEALU
/ & DBUS SMOKE & STATSHFT D031

05CC

(9) THE CONTENTS OF IR15 IS LOADED INTO REG.
MAC ...CONT & ALU ...NOP,OR,OR,2A & REG ...0R2,R2 & CEREQ & DBUS SMOKE

05CD

(10) THE LOADING OF THE INSTRUCTION REGISTER TAKES PLACE. JUMP S05 TO
OPCODES:
/ & DBUS SMOKE

05CE

(12) THE CONTENTS OF REG5 ARE PLACED IN REG2.
MAC ...CONT & ALU ...AND,AND,AND,OR,OR,2A & REG ...0R2,R5 & CEALU
/ & DBUS SMOKE

1 MAY 1961

PAGE 65

ANDOS/29 ANDASH MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (DMSL)(13) THE CONTENTS OF REG6 ARE PASSED THROUGH THE ALU.
MAC ...CONT & ALU ...AND,AND,AND,OR,OR,2A & REG ...0R2,R6 & DBUS SMOKE

05CF

(14) JUMP TO POPM2 TAKES PLACE IF THE PREVIOUS OPERATION GAVE ZLS.
MAC ...TRUE,ZLS,CJP & ALU & DBUS SMOKE & ADDRESS POPM2

05D0

(15) THE CONTENTS OF REG6 ARE DECREMENTED.
MAC ...CONT & ALU ...AND,AND,AND,OR,OR,2A & REG ...0R2,R6 & DBUS SMOKE
/ & CEREQ & STATSHFT D0000000000000000

05D1

(16) THE CONTENTS OF REG1 ARE INCREMENTED.
MAC ...CONT & ALU ...AND,AND,AND,OR,OR,2A & REG ...0R2,R1 & DBUS SMOKE
/ & CEREQ & STATSHFT D0010000000000000

05D2

(17) THE CONTENTS OF REG2 ARE INCREMENTED.
MAC ...CONT & ALU ...AND,AND,AND,OR,OR,2A & REG ...0R2,R2 & DBUS SMOKE
/ & CEREQ & STATSHFT D0010000000000000

05D3

(18) JUMP TAKES PLACE TO POPM1.
MAC ...JP & ALU & DBUS SMOKE & ADDRESS POPM1

05D4

(12) THE INSTRUCTION REGISTER IS LOADED WITH 15.
OPCODES: MACP ...CONT & ALU & DBUS SMOKE & DATINSRT D015

05D5

(14) THE CONTENTS OF REG2 ARE INCREMENTED AND PLACED IN IR15. RETURN TAKES
PLACE.
MAC ...FALSE,ZERO,CATM & ALU ...AND,AND,AND,OR,OR,2A & REG ...0R2,R2
/ & CEALU & DBUS SMOKE & STATSHFT D0010000000000000

05D6

STACK IC AND JUMP TO SUBROUTINE
.....ENTRY POINT
.....(1) JUMP TAKES PLACE TO RETIAD0.
OPCODES: MAC ...FALSE,ZERO,CJS & ALU & DBUS SMOKE & ADDRESS RETIAD0

05D7

(2) THE CONTENTS OF REG1 (RA) ARE DECREMENTED AND LEFT IN REG1.
MAC ...CONT & ALU ...AND,AND,AND,OR,OR,2A & CEALU & REG ...0R2,R1 & DBUS SMOKE & STATSHFT D0000000000000000

05D8

TM FS 40

1 MAY 1981

PAGE 60

ANDOS/20 ANDALM MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (DMSL)

```

: EXECUTION OF STEP (6).
/ 6 MAC TRUE,MI,LOOP
/ 6 ALU B00..ABUS.QREG.SUBR.SUBR.DQ & CVALU & DBUS SMOKE
/ 6 STATSHFT B0010000000100
/ 6 REG ...R10
:
: (7) THE CONTENTS OF IRO AND REG ARE SHIPPED -- RETURN TAKES PLACE.
/ 6 MAC .FALSE,ZZERO,CHTN & ALU ..ABUS.RAMA,OR,OR,DZ
/ 6 REG ..RZERO,RZERO,R0 & CVALU & CREG & DBUS SALU
:
:
: REGSELBIT -- THIS ROUTINE LOADS REG WITH A MASK TO INDICATE THE BIT FOR
: OPERATION. THE BIT POSITION IS DETERMINED FROM THE COUNT IN THE LEAST
: SIGNIFICANT FOUR BIT POSITIONS OF GR2 (R0).
:
: THE CONTENTS OF GR1 ARE MASKED BY 000000000001111 AND PLACED IN THE QREG.
/ 6 MICRO REGSELBIT: MAC ...CONT & ALU ..DBUS.QREG.AND,AND,DA & REG ...GR1 & CVALU
/ 6 DBUS SDATA & DATMSHT DMSL
:
: (2) THE CONTENTS OF THE Q REGISTER ARE PASSED THROUGH THE ALU AND THE
: MICROSTATUS REGISTER LOADED ACCORDINGLY. THE CONTENTS OF IRO ARE PLACED
: IN REG.
/ 6 MAC ...CONT & ALU B00..RAMA,OR,OR,ZQ & REG ..RZERO,RZERO,R0
/ 6 DBUS SALU & CREG & STATSHFT B0000000010000
:
: (3) IRO IS LOADED WITH 1 IN ITS LEAST SIG BIT POSITION AND ZERO EVERYWHERE
: ELSE. THE NEXT ADDRESS IS PUSHED INTO THE MICRO STACK. THE COUNTER IS
: NOT LOADED.
/ 6 MAC TRUE,ZZERO,PUSH & ALU ..DBUS.RAMT,OR,OR,DZ & DBUS SDATA
/ 6 DATMSHT D01 & CVALU & REG ..RZERO,RZERO
:
: (4) THE CONTENTS OF IRO ARE SHIPPED DOWN CYCLIC.
/ 6 MAC ...CONT & ALU ..RAMD,OR,OR,ZA & REG ..RZERO,RZERO
/ 6 CVALU & DBUS SMOKE & STATSHFT B001010000000
:
: (5) THE CONTENTS OF THE QREG ARE DECREMENTED. END OF LOOP TAKES PLACE
: IF MICRO 2 WAS SET EITHER DURING STEP (2) OR DURING THE PREVIOUS EXECUTION
: OF STEP (5).
/ 6 MAC TRUE,MI,LOOP & ALU B00....QREG.SUBR.SUBR,ZQ & CVALU & DBUS SMOKE
/ 6 STATSHFT B0000000000100
:
: (6) THE CONTENTS OF IRO AND REG ARE SHIPPED -- RETURN TAKES PLACE
/ 6 MAC .FALSE,ZZERO,CHTN & ALU ..ABUS.RAMA,OR,OR,DZ & REG ..RZERO,RZERO,R0
/ 6 CVALU & CREG & DBUS SALU

```

1 MAY 1981

PAGE 60

ANDOS/20 ANDALM MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (DMSL)

```

: SET BIT
/ 6
:
: ENTRY POINT FOR REGISTER ADDRESSING
/ 6
:
: OP005F1: MAC .FALSE,ZZERO,CJS & ALU & DBUS SMOKE & ADDRESS SELBIT
/ 6
: (2) THE MASK IN REG IS ORED WITH THE CONTENTS OF GR2. THE RESULT IS LEFT
: IN GR2 AND RETURN TAKES PLACE.
/ 6 MAC .FALSE,ZZERO,CHTN & ALU ..ABUS.RAMP,OR,OR,DA & CVALU
/ 6 REG ..GR2,GR2,R0 & DBUS SMOKE
:
: ENTRY POINT FOR DIRECT OR INDIRECT INDEXED ADDRESSING
/ 6
:
: OP005F1: MAC .FALSE,ZZERO,CJS & ALU & DBUS SMOKE & ADDRESS OFD01
/ 6
: (2) JUMP SUB TO SELBIT
/ 6 MAC .FALSE,ZZERO,CJS & ALU & DBUS SMOKE & ADDRESS SELBIT
:
: (3) THE MASK IN REG IS LOADED INTO THE Q REGISTER.
/ 6 MAC ...CONT & ALU ..ABUS.QREG,OR,OR,DZ & REG ....R0 & DBUS SMOKE
/ 6 CVALU
:
: (4) THE DERIVED OPERAND IN REG IS ORED WITH THE CONTENTS OF THE Q REGISTER
: AND THE RESULT LEFT IN REG.
/ 6 MAC ...CONT & ALU ..ABUS.NOP,OR,OR,DQ & REG ....R0 & DBUS SALU
/ 6 CREG
:
: (5) UNCONDITIONAL JUMP TAKES PLACE TO OSCMPOR THE RETURN WILL MARK THE END
: OF THE EMULATION SEQUENCE WITH THE STACK CORRECTLY PRESERVED.
/ 6 MAC ...JMP & ALU & DBUS SMOKE & ADDRESS OSCMPOR
:
: ENTRY POINT FOR INDIRECT AND INDIRECT INDEXED ADDRESSING
/ 6
:
: OP005F2: MAC .FALSE,ZZERO,CJS & ALU & DBUS SMOKE & ADDRESS OFI01
/ 6
: (2) JUMP SUB TO SELBIT
/ 6 MAC .FALSE,ZZERO,CJS & ALU & DBUS SMOKE & ADDRESS SELBIT
:
: (3) THE CONTENTS OF REG (MASK) IS PLACED IN THE QREG.
/ 6 MAC ...CONT & ALU ..ABUS.QREG,OR,OR,DZ & REG ....R0 & DBUS SMOKE

```


AMPOCS/20 ANALOG MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1950. (EMUL
/ 5 CARRYSEL C280

```

: : ENTRY POINT FOR INDIRECT AND INDIRECT INDEXED ADDRESSING
: : *****
: : OPCODES:== MAC ..PALSZ,TTZERO,CJS & ALU & DBUS SMOKE & BADDRESS OP111
: :
: : (2) JUMP SUB TO SELFIT
: : : MAC ..PALSZ,TTZERO,CJS & ALU & DBUS SMOKE & BADDRESS SELFIT
: :
: : (3) THE DERIVED OPERAND IN REG IS PLACED IN REG
: : : MAC ...CONT & ALU ..ABUS,REG,OR,02 & CCALL & REG ....R5
: : : DBUS SMOKE
: : / 6
: :
: : (4) THE MASK IN REG IS AND'ED WITH THE DERIVED OPERAND IN THE REG.
: : : MACHINE STATUS IS LOADED ACCORDING TO THE RESULT
: : : RETURN TAKES PLACE.
: : : MAC ..PALSZ,TTZERO,CRM & ALU ..R0,ABUS,HOP,AND,00 & DBUS SMOKE
: : : REG ....R8 & STATUSP XXXXXXXXXXXX
: : / 6
: : : CARRYSEL ZERO
: : / 6

```

```

: TEST AND SET BIT
: *****
:
: ENTRY POINT FOR DIRECT AND DIRECT INDEED ADDRESSING
:
: OPCODE50:: MAC ,FALSE,ZERO,CJS & ALU & DBUS SMOKE & BADDRESS OFD01
: *****
:
: (2) JUMP SET TO SELBIT
: MAC ,FALSE,ZERO,CJS & ALU & DBUS SMOKE & BADDRESS SELBIT
:
: (3) THE MASK IN REG IS PLACED IN QREG.
: MAC ,COUNT & ALU ,ABUS,QREG,OR,DI & CEALU & REG ,,,,BO
: DBUS SMOKE
: / &
:
: (4) THE DERIVED OPERAND IN REG IS ANDED WITH THE MASK IN QREG. THE MACHINE
: STATUS IS LOADED ACCORDING TO THE RESULT.
: MAC ,COUNT & ALU ,ABUS,MOP,AND,DI & REG ,,,,M5 & DBUS SMOKE
: / & STATUS,REG,CEALU,DI
: / & CARRYSEL CIFT.
:
: (5) THE DERIVED OPERAND IN REG IS ORED WITH THE MASK IN THE QREG. THE RESULT
: IS PLACED IN REG. JUMP PIPELINE TAKES PLACE TO OSCIPUR
: MAC ,JFF & ALU ,ABUS,MOP,OR,DI & REG ,,,,M5 & CEALU & DBUS SMOKE
: / &
:
:

```

ANDOS/29 ANDASII MICRO ASSEMBLER, V1.0
 EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EHS)

```

/ & ADDRESS 050ENP0R
-----
SET VARIABLE BIT IN REGISTER
*****
(1) JUMP TAKES PLACE TO REGSELBIT WHICH PROVIDES A MASK IN E80 WHICH
SELECTS THE REQUIRED BIT ACCORDING TO THE CONTENTS OF R1.
OPCODESA: MAC ,FALSE,,ZERO,CJS & ALU & DBUS SHONE & ADDRESS REGSELBIT
0613
(2) THE MASK IN E80 IS ORED WITH THE CONTENTS OF REGISTER GR2, RETURN TAKES
PLACE.
MAC ,FALSE,,ZERO,CNTM & ALU ,,ANUS,RAMP,OR,OR,DA & REG ,,GR2,GR2,R0
& C&LU & DBUS SHONE
0614 / &

```

```

0616 TEST VARIABLE BIT IN REGISTER
      .....
ENTRY POINT
      .....
      (1) JUMP SUB TO REGSELBIT (DESCRIBED ABOVE).
0615 OP0CODES: MAC ,FALSE,TZERO,CJS & ALU & DBUS SMOKE & BADDRESS REGSELBIT

      (2) THE NOT OF THE MASK IN REG IS ANDDED WITH THE CONTENTS OF REGISTER GR2
      THE RESULT IS PLACED IN GR2.
      RETURN TAKES PLACE.
      MAC ,FALSE,TZERO,CTRN & ALU ,.ABUS,RAMP,NOTES,NOTES,DA & REG ,.GR2,.GR2,AB
      CEAL0 & DBUS SMOKE
      /
      .....
0616 TEST VARIABLE BIT IN REGISTER
      .....
ENTRY POINT
      .....
      (1) JUMP SUB TO REGSELBIT
0617 OP0CODES: MAC ,FALSE,TZERO,CJS & ALU & DBUS SMOKE & BADDRESS REGSELBIT

```

1 MAY 1961

PAGE 74

ARMOR/20 ANDAM MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE INSTRUCTION SUBSET FOR 1750. (END)

```

(1) THE CONTENTS OF REG ARE ADDED WITH THE MASK IN REG AND MACHINE STATUS
    IS LOADED ACCORDING TO THE RESULT. THE CONTENTS OF REG ARE NOT CHANGED.
    RETURN TAKES PLACE
  
```

```

0010 / 6 MAC .FALSE, COUNT, RPT & ALU .AND, OR, 2A & REG .ZERO, ZERO
      / 6 CALL & STATUS, RPT & ALU .AND, OR, 2A & REG .ZERO, ZERO
      / 6 CALL & STATUS, RPT & ALU .AND, OR, 2A & REG .ZERO, ZERO
  
```

JUMP ON CONDITION

```

(1) THE CONTENTS OF REG ARE PLACED IN REG.
    (THE LABEL) INTO REG AND INDEED WITH THE CONTENTS OF THE INDEED REGISTER
    IF ONE IS SPECIFIED.
  
```

```

0011 / 6 MAC .FALSE, ZERO, CTRN & ALU & DMS, SHONE & ADDRESS, PTHADP
      / 6 CALL & STATUS, RPT & ALU .AND, OR, 2A & REG .ZERO, ZERO
  
```

```

(1) THE CONTENTS OF REG ARE PLACED IN REG.
    (THE LABEL) INTO REG AND INDEED WITH THE CONTENTS OF THE INDEED REGISTER
    IF ONE IS SPECIFIED.
  
```

```

0012 / 6 MAC .FALSE, ZERO, CTRN & ALU & DMS, SHONE & ADDRESS, PTHADP
      / 6 CALL & STATUS, RPT & ALU .AND, OR, 2A & REG .ZERO, ZERO
  
```

```

(1) THE INSTRUCTION IS MASKED BY (00000001110000).
    (THE LABEL) INTO REG AND INDEED WITH THE CONTENTS OF THE INDEED REGISTER
    IF ONE IS SPECIFIED.
  
```

```

0013 / 6 MAC .FALSE, ZERO, CTRN & ALU & DMS, SHONE & ADDRESS, PTHADP
      / 6 CALL & STATUS, RPT & ALU .AND, OR, 2A & REG .ZERO, ZERO
  
```

```

(1) THE CONTENTS OF REG ARE DOWNSHIPPED.
    (THE LABEL) INTO REG AND INDEED WITH THE CONTENTS OF THE INDEED REGISTER
    IF ONE IS SPECIFIED.
  
```

```

0014 / 6 MAC .FALSE, ZERO, CTRN & ALU & DMS, SHONE & ADDRESS, PTHADP
      / 6 CALL & STATUS, RPT & ALU .AND, OR, 2A & REG .ZERO, ZERO
  
```

(1) THE CONTENTS OF REG ARE DOWNSHIPPED 7 TIMES CIRCULARLY.

1 MAY 1961

PAGE 75

ARMOR/20 ANDAM MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE INSTRUCTION SUBSET FOR 1750. (END)

```

0015 / 6 MAC .FALSE, COUNT, RPT & ALU .AND, OR, 2A & REG .ZERO, ZERO
      / 6 CALL & STATUS, RPT & ALU .AND, OR, 2A & REG .ZERO, ZERO
  
```

```

(1) THE CONTENTS OF THE STATUS REGISTER ARE ADDED WITH THE CONTENTS OF
    REG.
  
```

```

0016 / 6 MAC .FALSE, ZERO, CTRN & ALU & DMS, SHONE & ADDRESS, PTHADP
      / 6 CALL & STATUS, RPT & ALU .AND, OR, 2A & REG .ZERO, ZERO
  
```

```

(1) THE CONTENTS OF REG ARE PLACED BACK IN REG. RETURN TAKES PLACE IF
    ZERO WAS SET.
  
```

```

0017 / 6 MAC .FALSE, ZERO, CTRN & ALU & DMS, SHONE & ADDRESS, PTHADP
      / 6 CALL & STATUS, RPT & ALU .AND, OR, 2A & REG .ZERO, ZERO
  
```

```

(1) THE QREG IS LOADED WITH THE CONTENTS OF REG.
    (THE LABEL) INTO REG AND INDEED WITH THE CONTENTS OF THE INDEED REGISTER
    IF ONE IS SPECIFIED.
  
```

```

0018 / 6 MAC .FALSE, ZERO, CTRN & ALU & DMS, SHONE & ADDRESS, PTHADP
      / 6 CALL & STATUS, RPT & ALU .AND, OR, 2A & REG .ZERO, ZERO
  
```

```

(1) THE PC IS LOADED WITH THE CONTENTS OF THE QREG. RETURN TAKES PLACE.
    (THE LABEL) INTO REG AND INDEED WITH THE CONTENTS OF THE INDEED REGISTER
    IF ONE IS SPECIFIED.
  
```

```

0019 / 6 MAC .FALSE, ZERO, CTRN & ALU & DMS, SHONE & ADDRESS, PTHADP
      / 6 CALL & STATUS, RPT & ALU .AND, OR, 2A & REG .ZERO, ZERO
  
```

```

(1) THE DERIVED OPERAND IN REG IS PLACED IN THE QREG.
    (THE LABEL) INTO REG AND INDEED WITH THE CONTENTS OF THE INDEED REGISTER
    IF ONE IS SPECIFIED.
  
```

```

0020 / 6 MAC .FALSE, ZERO, CTRN & ALU & DMS, SHONE & ADDRESS, PTHADP
      / 6 CALL & STATUS, RPT & ALU .AND, OR, 2A & REG .ZERO, ZERO
  
```

```

(1) THE QREG CONTENTS ARE PLACED IN REG. JUMP TO JCI TAKES PLACE.
    (THE LABEL) INTO REG AND INDEED WITH THE CONTENTS OF THE INDEED REGISTER
    IF ONE IS SPECIFIED.
  
```

```

0021 / 6 MAC .FALSE, ZERO, CTRN & ALU & DMS, SHONE & ADDRESS, PTHADP
      / 6 CALL & STATUS, RPT & ALU .AND, OR, 2A & REG .ZERO, ZERO
  
```

```

(1) THE QREG CONTENTS ARE PLACED IN REG. JUMP TO JCI TAKES PLACE.
    (THE LABEL) INTO REG AND INDEED WITH THE CONTENTS OF THE INDEED REGISTER
    IF ONE IS SPECIFIED.
  
```

```

0022 / 6 MAC .FALSE, ZERO, CTRN & ALU & DMS, SHONE & ADDRESS, PTHADP
      / 6 CALL & STATUS, RPT & ALU .AND, OR, 2A & REG .ZERO, ZERO
  
```

```

(1) THE QREG CONTENTS ARE PLACED IN REG. JUMP TO JCI TAKES PLACE.
    (THE LABEL) INTO REG AND INDEED WITH THE CONTENTS OF THE INDEED REGISTER
    IF ONE IS SPECIFIED.
  
```

```

0023 / 6 MAC .FALSE, ZERO, CTRN & ALU & DMS, SHONE & ADDRESS, PTHADP
      / 6 CALL & STATUS, RPT & ALU .AND, OR, 2A & REG .ZERO, ZERO
  
```

TM PS 403

IM FS 403

1 MAY 1961

PAGE 76

AMDOS/29 ANDRAM MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (ENSL)

```

(1) JUMP TO PTDIADD. THIS PITCHES THE SECOND WORD OF THE TWO WORD INSTRUCTION
    INTO ER2. THE RESULT IS THEN INDEXED AT THE CONTENTS OF THE INDEX REGISTER
    IF SPECIFIED. THE PROGRAM COUNTER CONTENTS ARE INCREMENTED TO TAKE
    ACCOUNT OF THE TWO WORD INSTRUCTION. AS THE PROGRAM COUNTER WILL ALREADY
    POINT TO THE NEXT LOCATION AFTER THE CURRENT INSTRUCTION SINCE IT IS
    INCREMENTED ONCE IN THE PREVIOUS SEQUENCE.

```

0020 OPCODE72: MAC ,FALSE,TZERO,CJS & ALU & DBUS SNOKE & ADDRESS PTDIADD

```

0027 / &
      MAC ...CONT & ALU ...ABUS,RAMP,OR,OR,DZ & RFS ,CRI,,P2 & CICALU
      DBUS SNOKE

```

```

(3) THE CONTENTS OF CRI (RA) ARE SHIPPED WITH THE CONTENTS OF ER0, RETURN
    TAKES PLACE.
      MAC ,FALSE,TZERO,CRTM & ALU ...ABUS,RAMP,OR,OR,DZ & REG ,CRI,CRI,NO
      CEREK & CICALU & DBUS SALU

```

SUBTRACT ONE AND JUMP

ENTRY POINT

0029 OPCODE73: MAC ,FALSE,TZERO,CJS & ALU & DBUS SNOKE & ADDRESS PTDIADD

```

(1) THE CONTENTS OF CRI (RA) ARE DECREMENTED, THE MACHINE STATUS REGISTER
    IS LOADED ACCORDINGLY.
      MAC ...CONT & ALU ,R00,RAMP,SUBR,SUBR,24 & REG ,CRI,CRI
      CICALU & DBUS SNOKE & STATSHFT 000000010000 & MACSTEN 000000
      CARRYSEL CZERO

```

```

(2) RETURN TAKES PLACE IF MACHINE STATUS ZERO IS SET.
      MAC ,TRUE,M01,CRTM & ALU & DBUS SNOKE & STATSHFT 0000000100100

```

002C (4) THE QREG IS LOADED WITH THE CONTENTS OF ER2.

```

      MAC ...CONT & ALU ...ABUS,QREG,OR,OR,DZ & REG ....R2 & CICALU
      DBUS SNOKE

```

0029 (5) THE PC (ER0) IS LOADED WITH THE CONTENTS OF QREG.

```

      RETURN TAKES PLACE.
      MAC ,FALSE,TZERO,CRTM & ALU ...MOP,OR,OR,DZ & REG ....R0 & CEREK
      DBUS SALU

```

1 MAY 1961

PAGE 77

AMDOS/29 ANDRAM MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (ENSL)

BRANCH UNCONDITIONALLY

002E OPCODE74: MAC ...CONT & ALU ...ABUS,MOP,SUBS,SUBS,DZ & CEREK & DBUS SALU

```

/ &
      MAC ...R0 & STATSHFT 000000000000

```

002F (1) THE CONTENTS OF ER1 IS PLACED IN QREG.

```

      MAC ...CONT & ALU ...ABUS,QREG,OR,OR,DZ & REG ....R1 & CICALU
      DBUS SNOKE

```

```

(2) THE CONTENTS OF QREG ARE MASKED WITH 0000000010000000 BUT NO REGISTERS
    ARE LOADED.
      MAC ...CONT & ALU ...DBUS,MOP,AND,AND,DQ & DBUS SDATA
      DATINSRT D0120

```

0030 (3) JUMP TO ER1 TAKES PLACE IF THE MASKED BIT WAS SET (IE LSB NOT -6)

```

      MAC ,FALSE,ZLSB,CJP & ALU & DBUS SNOKE & ADDRESS BR1

```

```

(4) CONTINUE KNOWING THAT INSTRUCTION COUNTER RELATIVE OPERAND IS POSITIVE.
    THE CONTENTS OF QREG ARE MASKED BY (00000000111111) AND LEFT IN QREG
      MAC ...CONT & ALU ...DBUS,QREG,AND,AND,DQ & CICALU & DBUS SDATA
      DATINSRT D0127

```

0033 (5) THE CONTENTS OF QREG ARE ADDED TO ER0 (PC). RETURN TAKES PLACE.

```

      MAC ,FALSE,TZERO,CRTM & ALU ...ABUS,MOP,ADD,ADD,DQ & CEREK
      REG ....R0 & DBUS SALU & STATSHFT 000000000000

```

```

(6) CONTINUE KNOWING THAT THE INSTRUCTION COUNTER RELATIVE OPERAND MUST BE
    TREATED AS NEGATIVE.
    THE CONTENTS OF QREG ARE ORED WITH (1111111100000000). THIS CONVERTS THE
    8 BIT 2'S COMPLEMENT NEGATIVE NUMBER TO A 16 BIT 2'S COMPLEMENT NEGATIVE
    NUMBER.
      MAC ...CONT & ALU ...DBUS,QREG,OR,OR,DQ & CICALU & DBUS SDATA
      DATINSRT D05200

```

```

(7) THE CONTENTS OF QREG ARE ADDED TO THE CONTENTS OF ER0 (PC). RETURN
    TAKES PLACE.
      MAC ,FALSE,TZERO,CRTM & ALU ...ABUS,MOP,ADD,ADD,DQ & DBUS SALU
      CEREK & REG ....R0 & STATSHFT 000000000000

```

BRANCH IF EQUAL TO ZERO

1 MAY 1981

PAGE 78

AM05/29 ANDASH MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EWSL)

```

*****
: ENTRY POINT
: *****
:
: (1) THE STATUS REGISTER CONTENTS ARE LOADED INTO THE QREG
0636 OPCODE75:: MAC ...CONT & ALU ...DBUS.QREG.OR.OR.DZ & DBUS SSTATUS & CVALU
: / & STATUSBIT 0000000000000000
:
: (2) THE CONTENTS OF THE Q REGISTER ARE MASKED BY (0010000000000000) SO THAT
: ONLY THE ZERO BIT IS LEFT.
0637 / &
: MAC ...CONT & ALU ...DBUS.NOP.AND.AND.DQ & DBUS SDATA
: DATASRT D4192
:
: (3) RETURN TAKES PLACE IF THE BIT WAS ZERO (IF ZMSB IS SET)
0638 / &
: MAC ...TRUE.ZMSB.CRTM & ALU & DBUS SNONE
:
: (4) JUMP TO OPCODE74
0639 / &
: MAC ...JP & ALU & DBUS SNONE & BADDRESS OPCODE74
:
: BRANCH IF LESS THAN ZERO
: *****
:
: (1) THE STATUS REGISTER CONTENTS ARE LOADED INTO THE QREG.
063A OPCODE76:: MAC ...CONT & ALU ...DBUS.QREG.OR.OR.DZ & DBUS SSTATUS & CVALU
: / & STATUSBIT 0000000000000000
:
: (2) THE CONTENTS OF THE QREG ARE MASKED WITH (0010000000000000) SO THAT
: ONLY THE NEGATIVE BIT IS LEFT.
063B / &
: MAC ...CONT & ALU ...DBUS.NOP.AND.AND.DQ & DBUS SDATA
: DATASRT D4496
:
: (3) RETURN TAKES PLACE IF THE BIT WAS ZERO (IF ZMSB IS SET.)
063C / &
: MAC ...TRUE.ZMSB.CRTM & ALU & DBUS SNONE
:
: (4) JUMP TO OPCODE74
063D / &
: MAC ...JP & ALU & DBUS SNONE & BADDRESS OPCODE74
:
: BRANCH IF LESS THAN OR EQUAL TO ZERO
: *****
:
: (1) THE STATUS REGISTER CONTENTS ARE LOADED INTO THE QREG.

```

1 MAY 1981

PAGE 79

AM05/29 ANDASH MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EWSL)

```

063E OPCODE78:: MAC ...CONT & ALU ...DBUS.QREG.OR.OR.DZ & DBUS SSTATUS & CVALU
: / & STATUSBIT 0000000000000000
:
: (2) THE CONTENTS OF THE QREG ARE MASKED WITH (0010000000000000) SO THAT
: THE NEGATIVE BIT AND ZERO BIT ARE LEFT.
063F / &
: MAC ...CONT & ALU ...DBUS.NOP.AND.AND.DQ & DBUS SDATA
: DATASRT D412248
:
: (3) RETURN TAKES PLACE IF BOTH THE BITS WERE ZERO (IF ZMSB IS SET.)
0640 / &
: MAC ...TRUE.ZMSB.CRTM & ALU & DBUS SNONE
:
: (4) JUMP TO OPCODE74
0641 / &
: MAC ...JP & ALU & DBUS SNONE & BADDRESS OPCODE74
:
: BRANCH IF GREATER THAN ZERO
: *****
:
: (1) THE STATUS REGISTER CONTENTS ARE LOADED INTO THE QREG.
0642 OPCODE79:: MAC ...CONT & ALU ...DBUS.QREG.OR.OR.DZ & DBUS SSTATUS & CVALU
: / & STATUSBIT 0000000000000000
:
: (2) THE CONTENTS OF THE QREG ARE MASKED WITH (0100000000000000) TO LEAVE
: ONLY THE POSITIVE BIT.
0643 / &
: MAC ...CONT & ALU ...DBUS.NOP.AND.AND.DQ & DBUS SDATA
: DATASRT D415284
:
: (3) RETURN TAKES PLACE IF THE POSITIVE BIT WAS NOT SET. (IF ZMSB WAS SET)
0644 / &
: MAC ...TRUE.ZMSB.CRTM & ALU & DBUS SNONE
:
: (4) JUMP TO OPCODE74
0645 / &
: MAC ...JP & ALU & DBUS SNONE & BADDRESS OPCODE74
:
: BRANCH IF NOT EQUAL TO ZERO
: *****
:
: (1) THE STATUS REGISTER CONTENTS ARE LOADED INTO THE QREG.
0646 OPCODE7A:: MAC ...CONT & ALU ...DBUS.QREG.OR.OR.DZ & DBUS SSTATUS & CVALU
: / & STATUSBIT 0000000000000000
:
: (2) THE CONTENTS OF THE QREG ARE MASKED WITH (0101000000000000) SO THAT
: THE NEGATIVE AND POSITIVE BITS ARE LEFT.
0647 / &
: MAC ...CONT & ALU ...DBUS.NOP.AND.AND.DQ & DBUS SDATA

```

TM FS 303

1 MAY 1961

PAGE 88

AMD320 MICRO ASSEMBLER, V1.0
 EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EMSL)

/ & DATINSTR D029400
 0040 (3) RETURN TAKES PLACE IF NEITHER OF THESE TWO BITS WERE SET. IE ZMSB IS SET.
 MAC ,TRUE,ZMSB,CEN & ALU & DBUS SMOKE
 0040 (4) JUMP TO OPCODE74
 MAC ,...JP & ALU & DBUS SMOKE & BADDRESS OPCODE74

BRANCH IF GREATER THAN OR EQUAL TO ZERO

 0040 (1) THE CONTENTS OF THE STATUS REGISTER ARE LOADED INTO THE QREG.
 OPCODE74: MAC ,...CONT & ALU ,DBUS,QREG,OR,OR,DZ & CALU & DBUS STATUS
 / & STATSHRT D0000000000000000

0040 (2) THE CONTENTS OF THE QREG ARE MASKED BY (0110000000000000) SO THAT
 ONLY THE POSITIVE AND ZERO BITS ARE LEFT.
 MAC ,...CONT & ALU ,DBUS,MOP,AND,AND,DQ & DBUS SDATA
 / & DATINSTR D024370

0040 (3) RETURN TAKES PLACE IF BOTH BITS WERE ZERO. IE ZMSB IS SET.
 MAC ,TRUE,ZMSB,CEN & ALU & DBUS SMOKE

0040 (4) JUMP TO OPCODE74
 MAC ,...JP & ALU & DBUS SMOKE & BADDRESS OPCODE74

THIS IS THE END OF THE EMULATION SEQUENCES.

END

REFERENCES

| <u>No.</u> | <u>Author</u> | <u>Title, etc</u> |
|------------|---------------------------------|--|
| 1 | USAF | MIL-STD-1750 Airborne computer instruction set architecture.
Washington DC 20360, 1 March 1980 |
| 2 | A.A. Callaway
S.J. Shrimpton | RAE contract with the USAF MIL-STD-1750 instruction set
architecture standardisation programme.
RAE Technical Memorandum FS 286 (1979) |
| 3 | Advanced Micro
Devices | Micro Channel Architecture |
| 4 | Advanced Micro
Computers | Advanced Micro Computers and Peripherals system.
Specifications and Applications |

REPORTS AND DOCUMENTS AVAILABLE
FROM THE NATIONAL AERONAUTICS AND SPACE
ADMINISTRATION

Fig 1

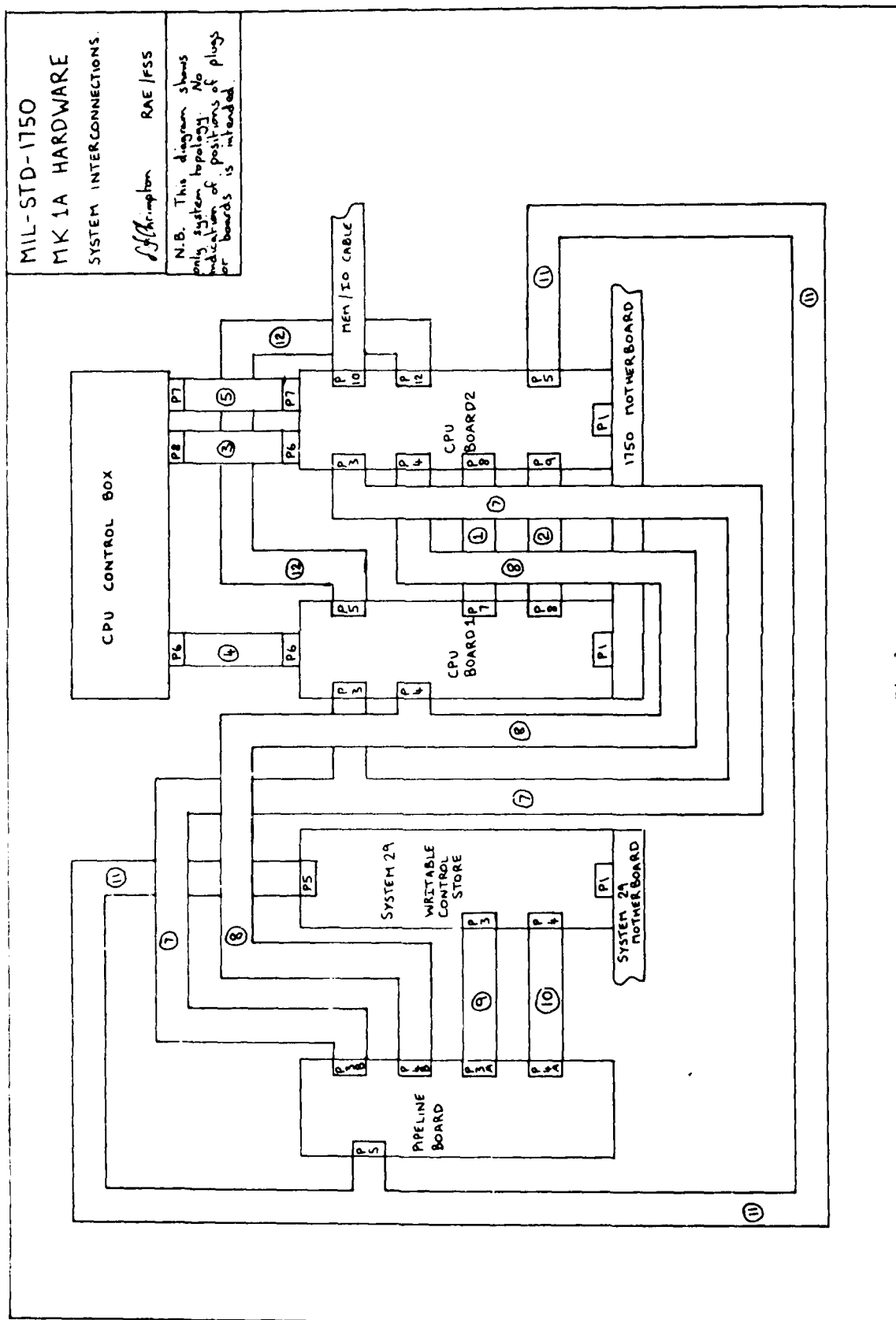
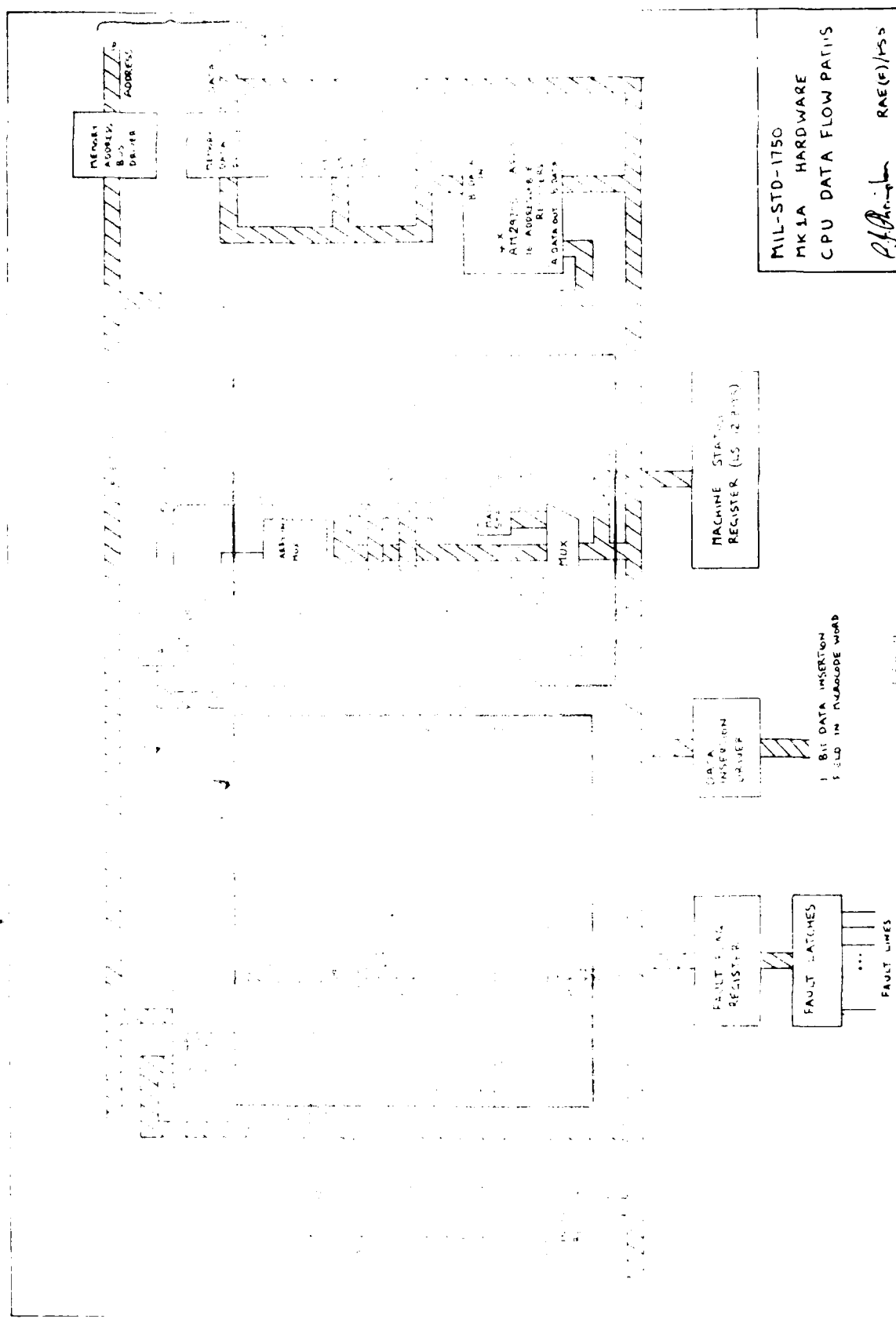


Fig 1

Fig 2



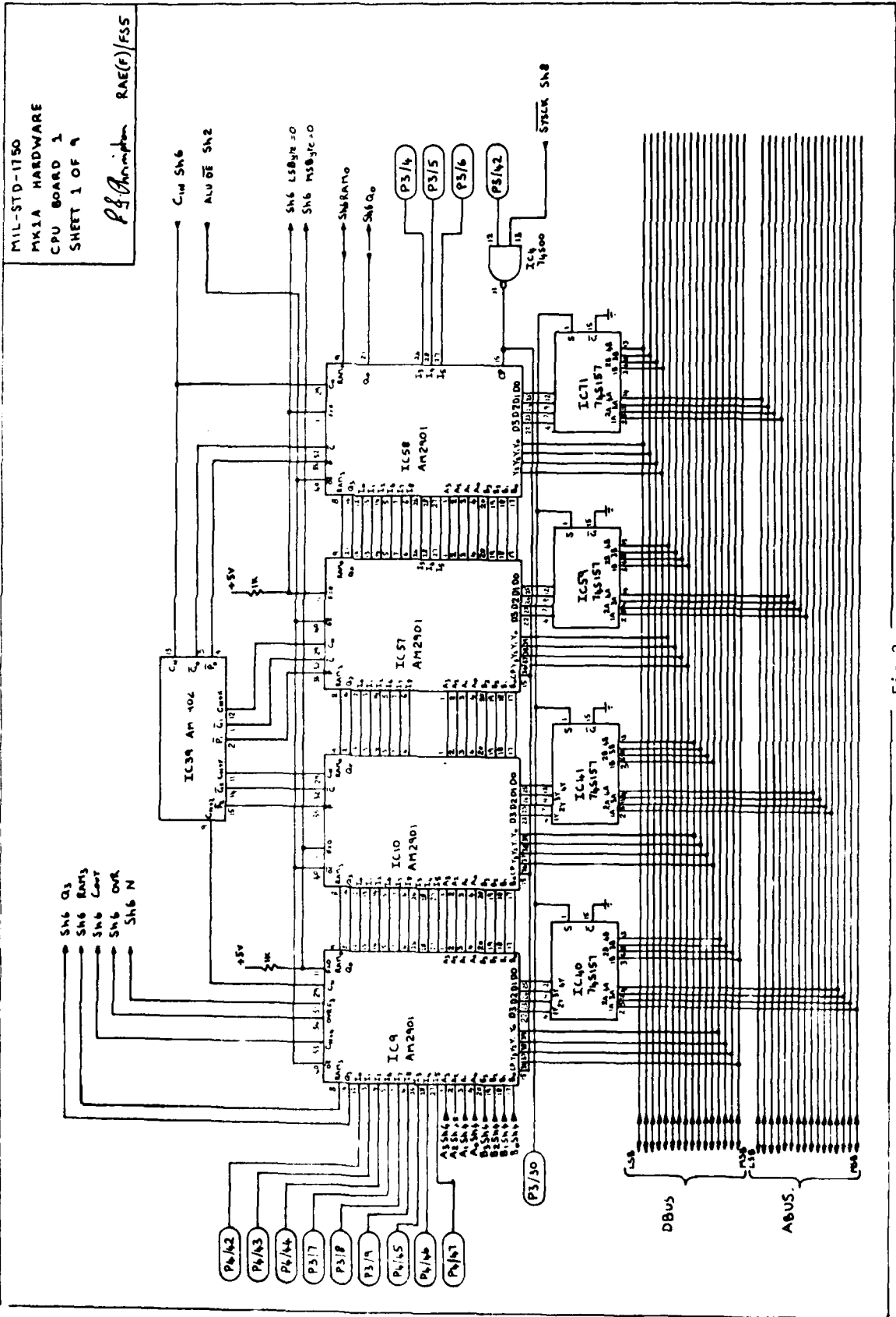


Fig 3

Fig 4

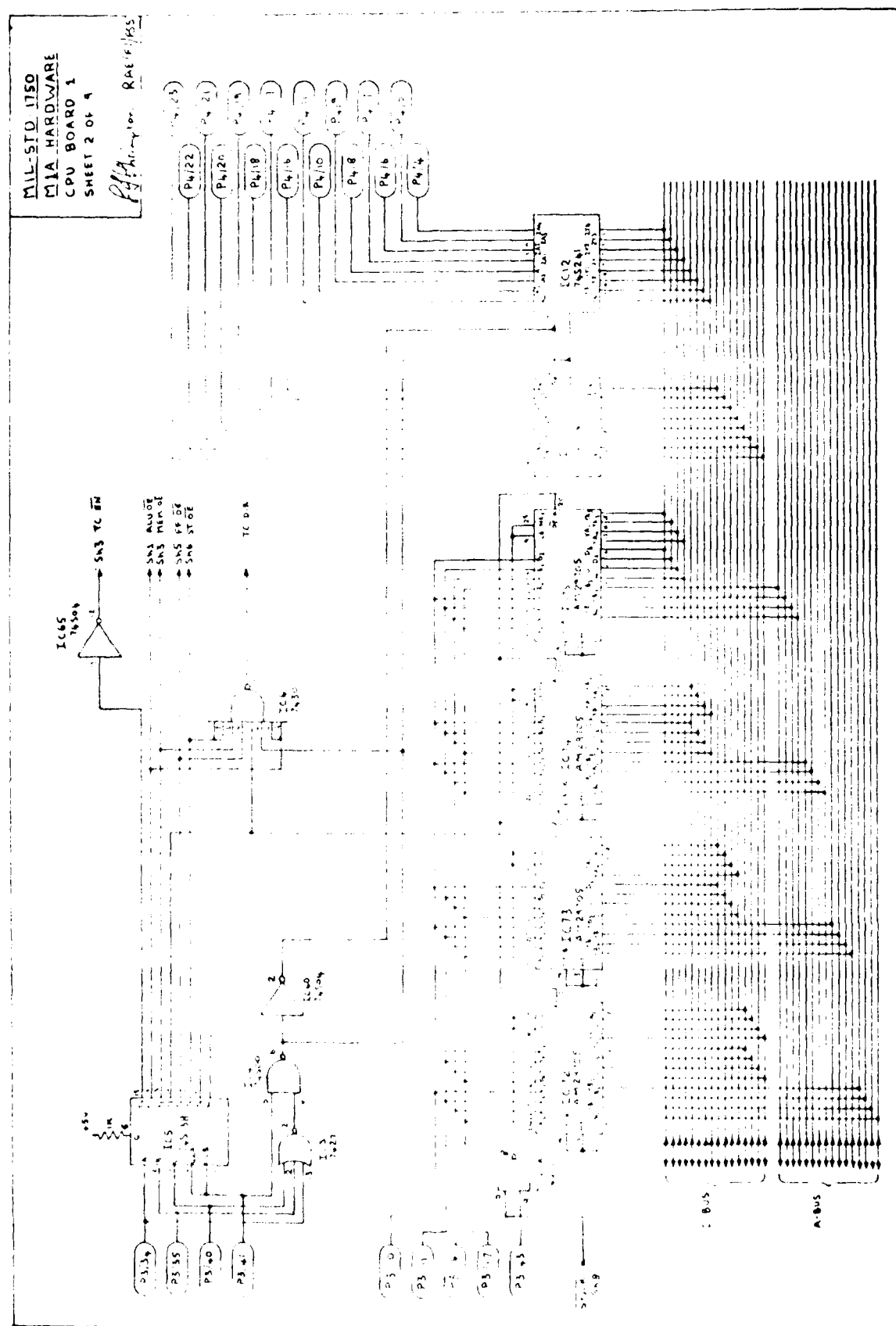


Fig 4

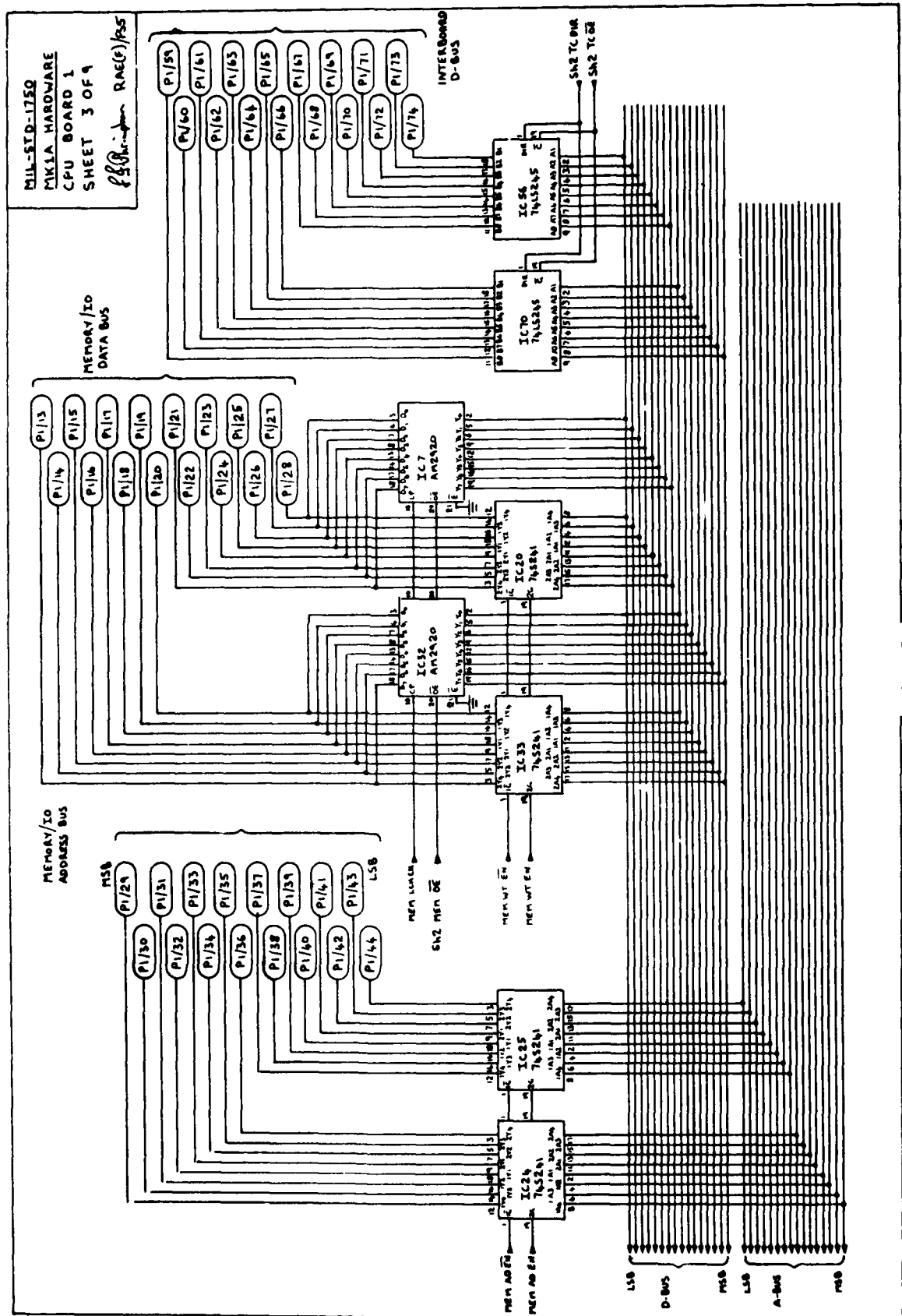


Fig 5

Fig 6

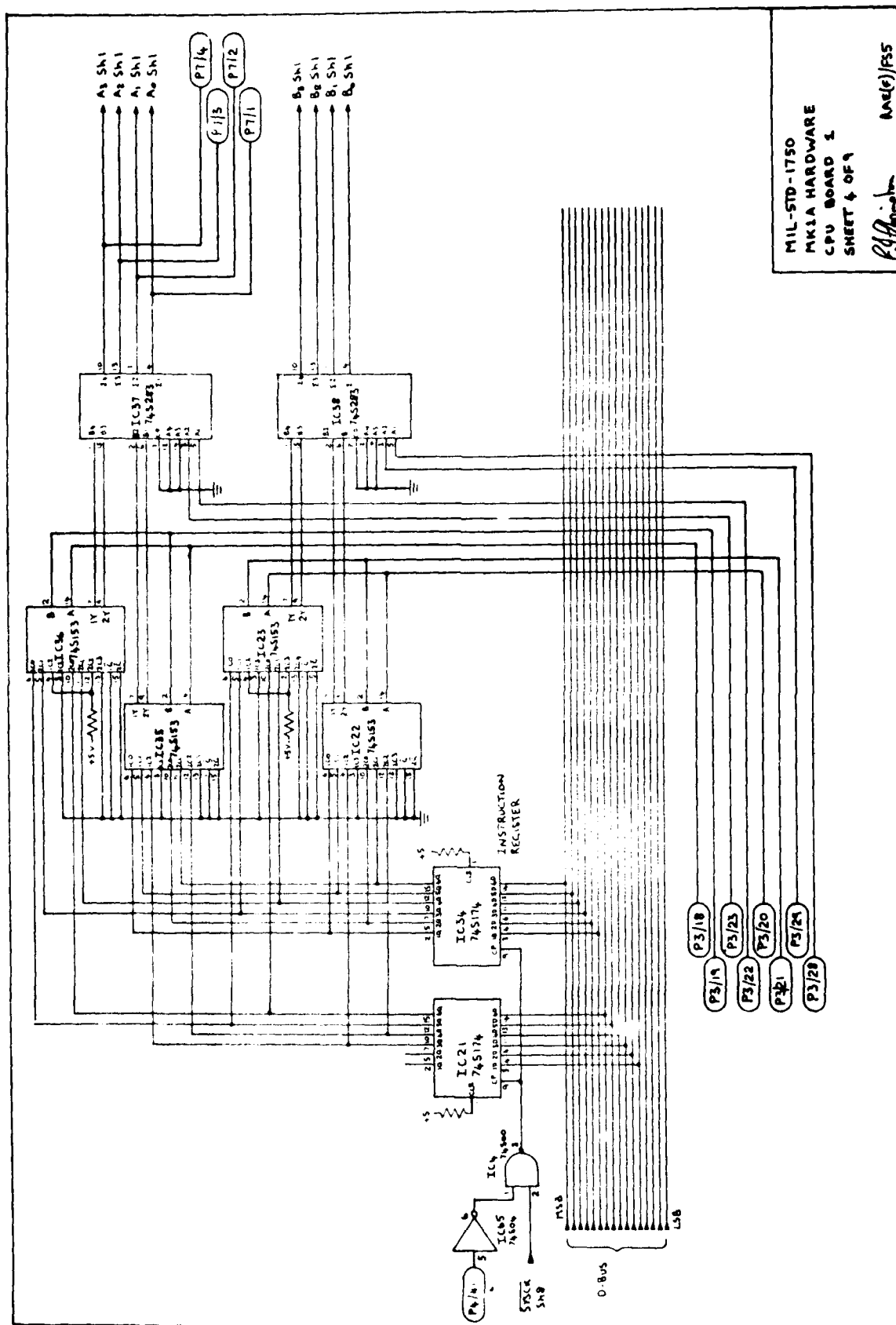


Fig 6

Fig 7

TM 15-403

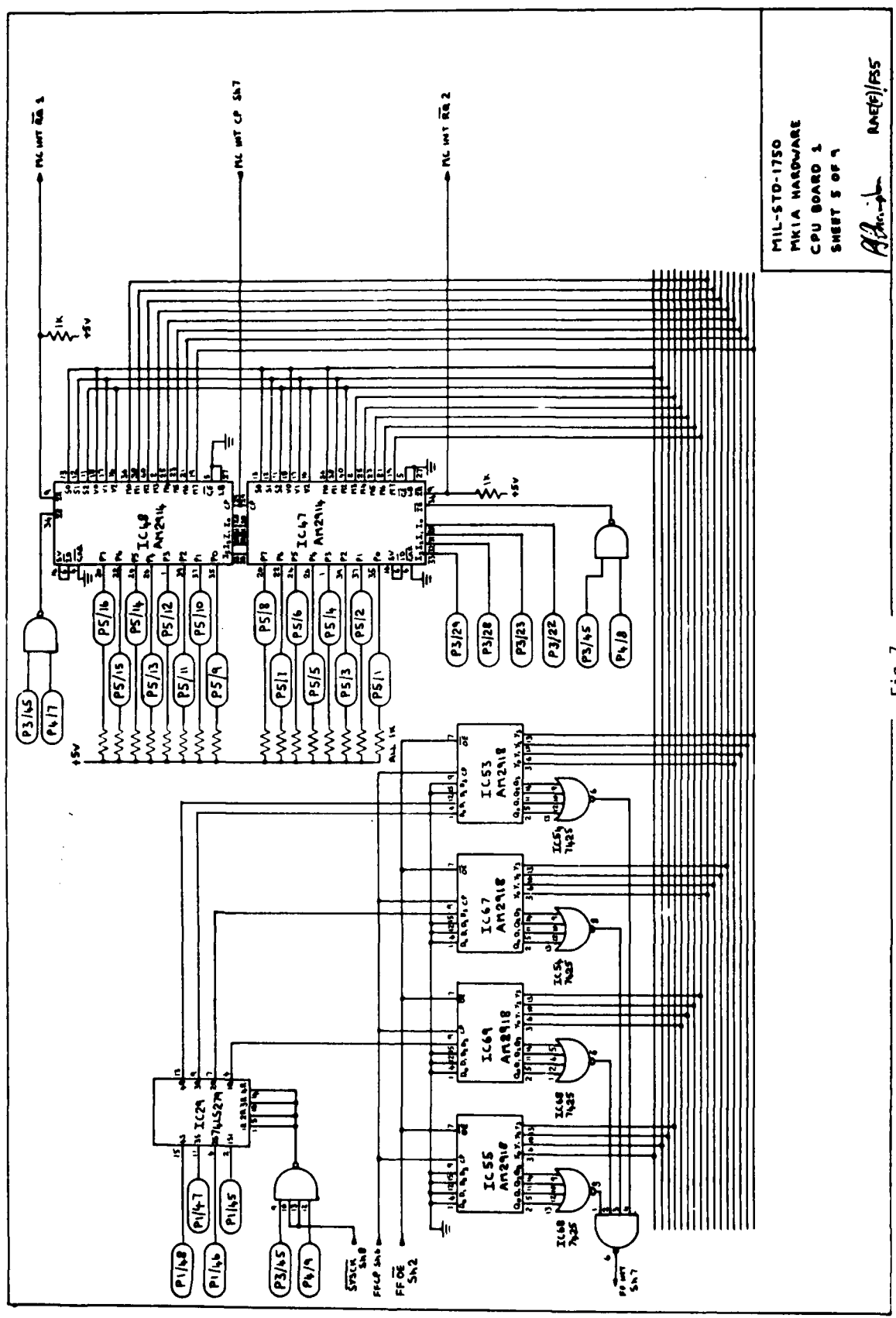


Fig 7

Fig 8

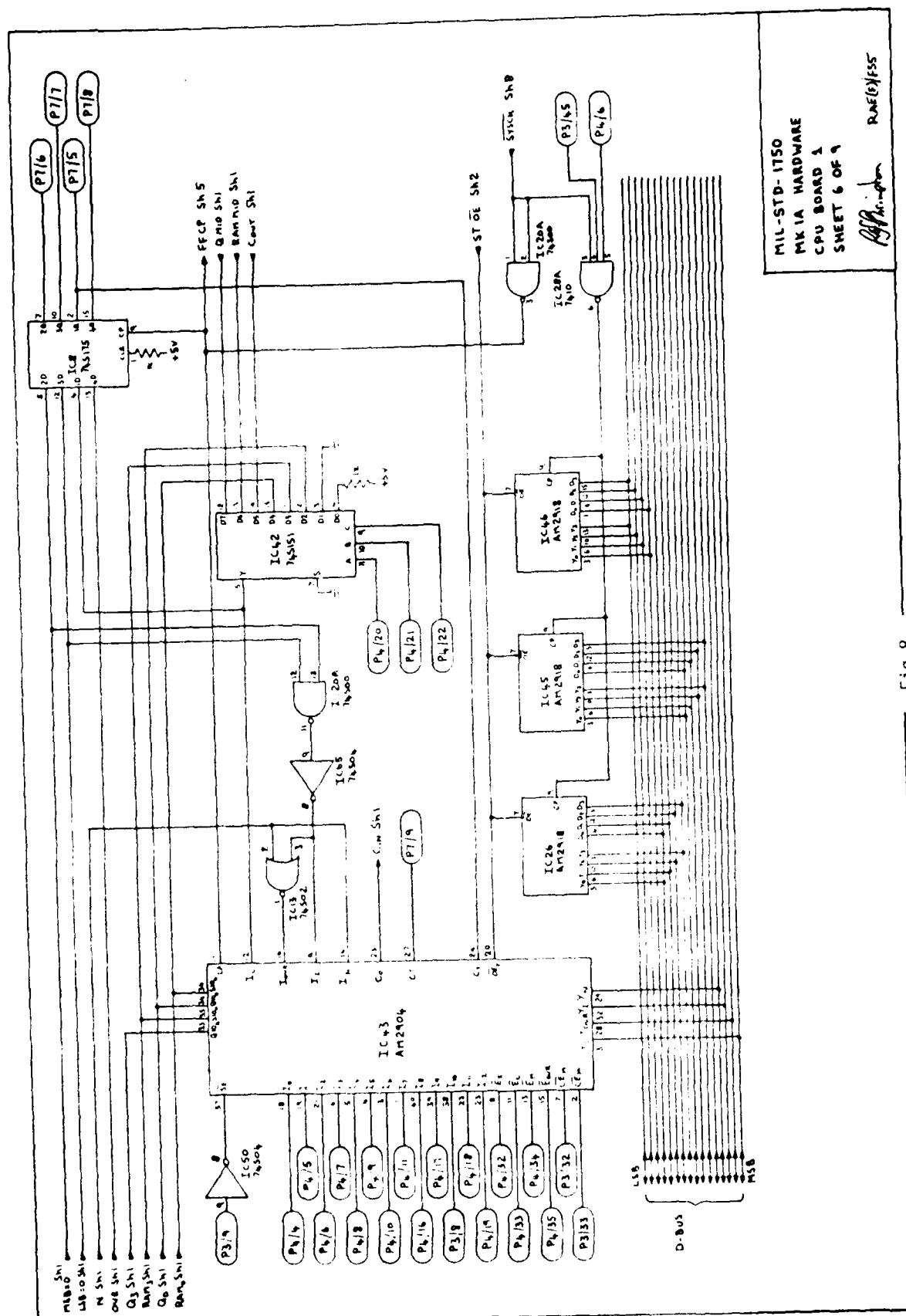


Fig 8

Fig 9

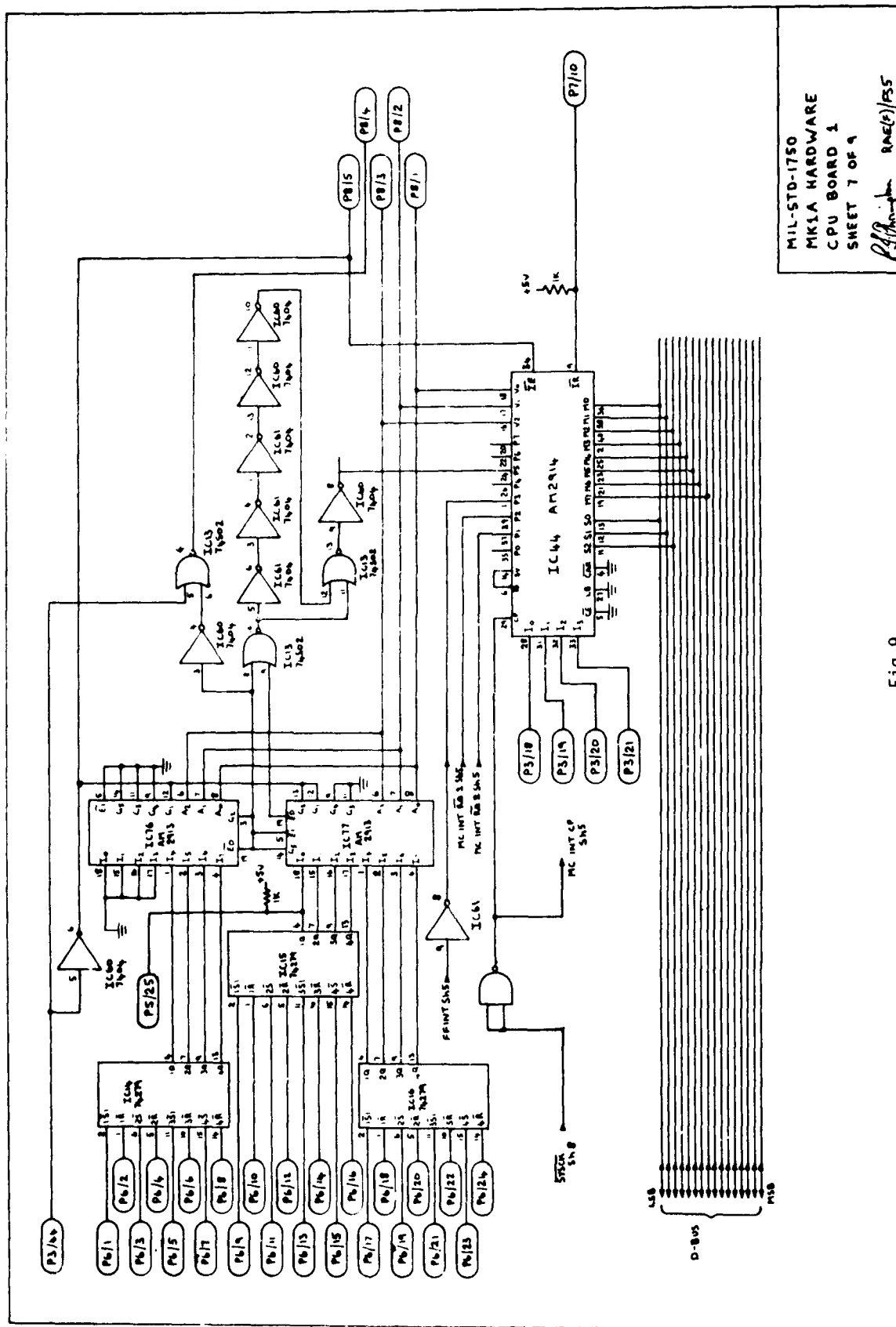
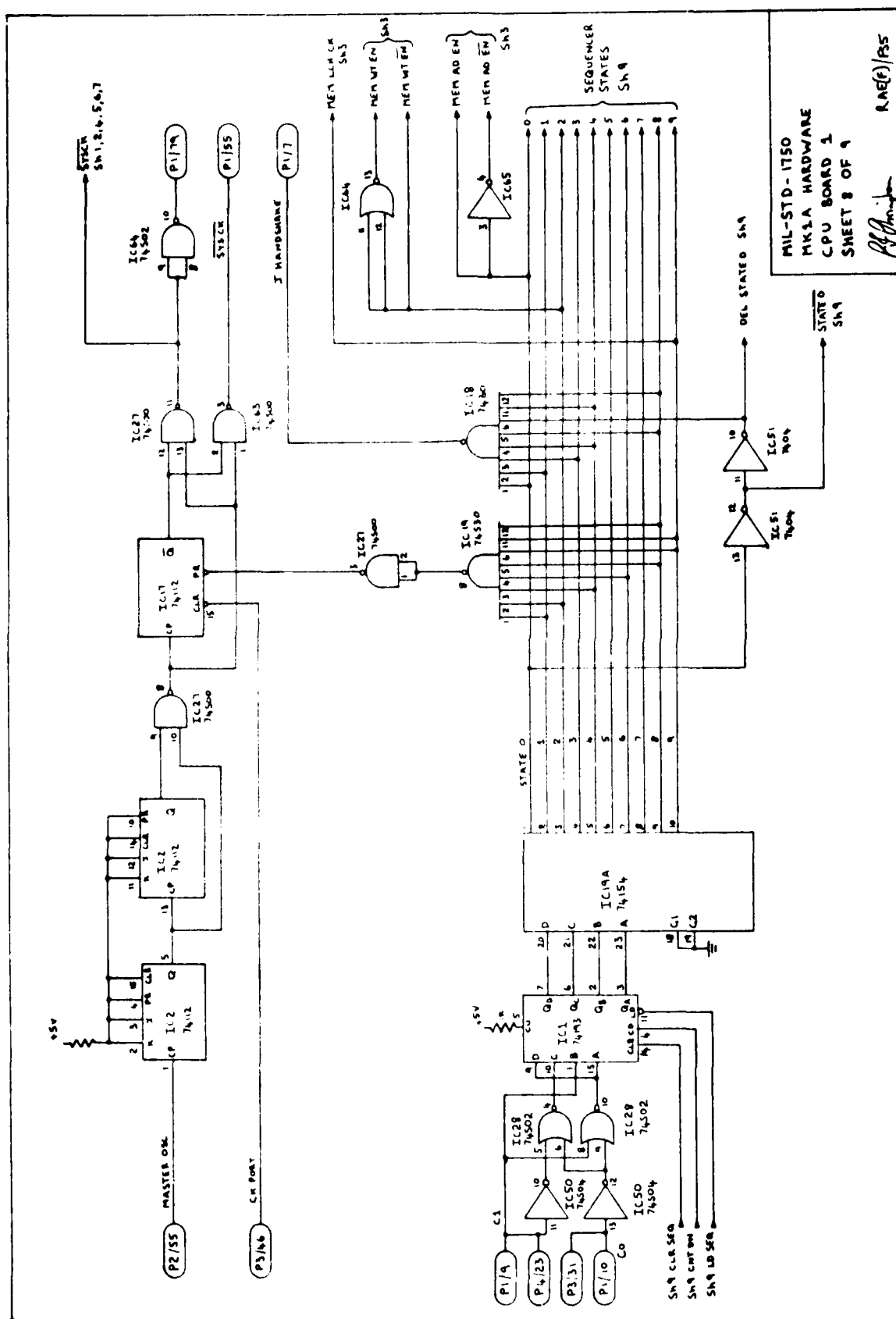


Fig 9

MIL-STD-1750
MK1A HARDWARE
CPU BOARD 1
SHEET 7 OF 9
P. J. Thompson RAE(1)/P55

Fig 10



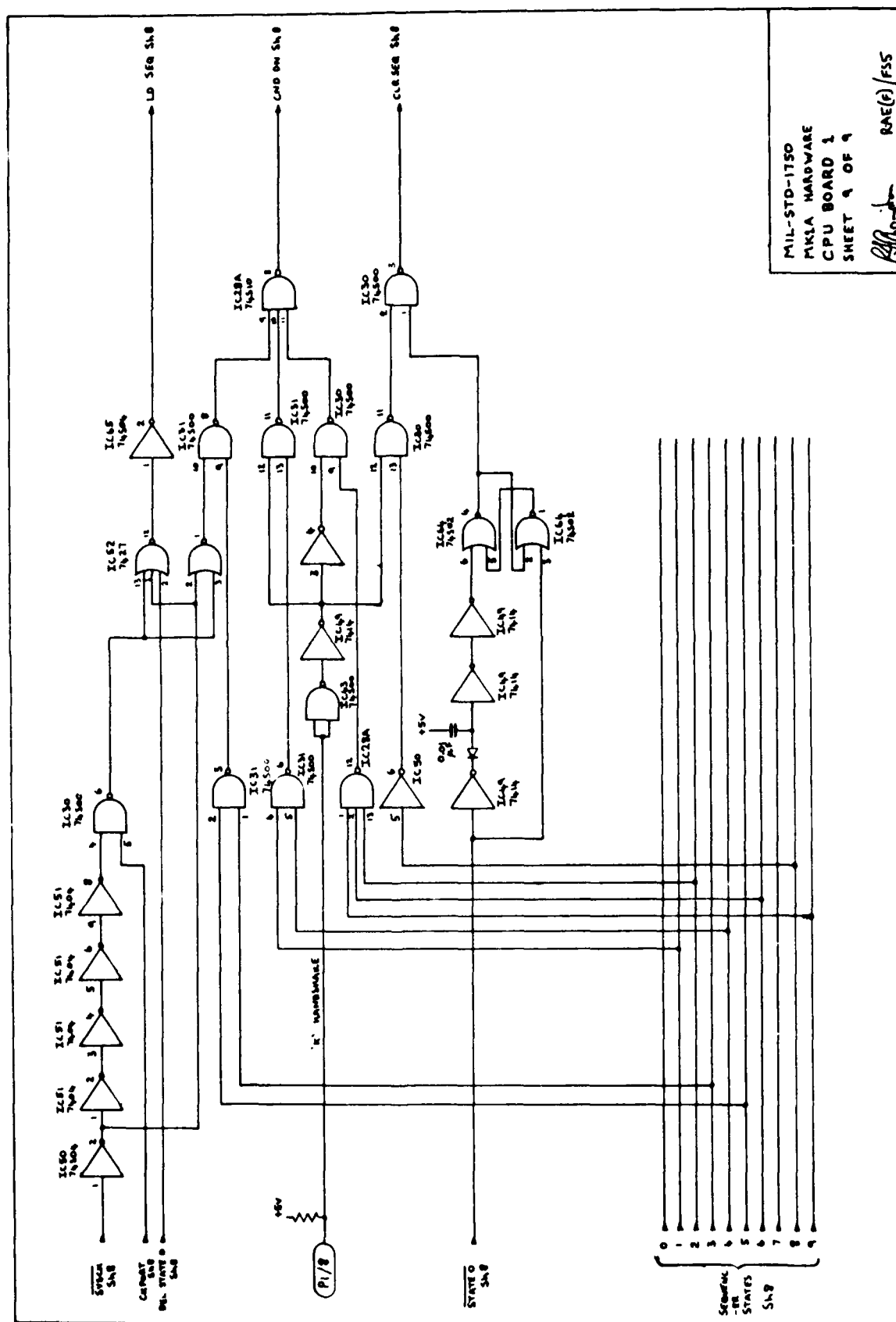


Fig 11

Fig 12

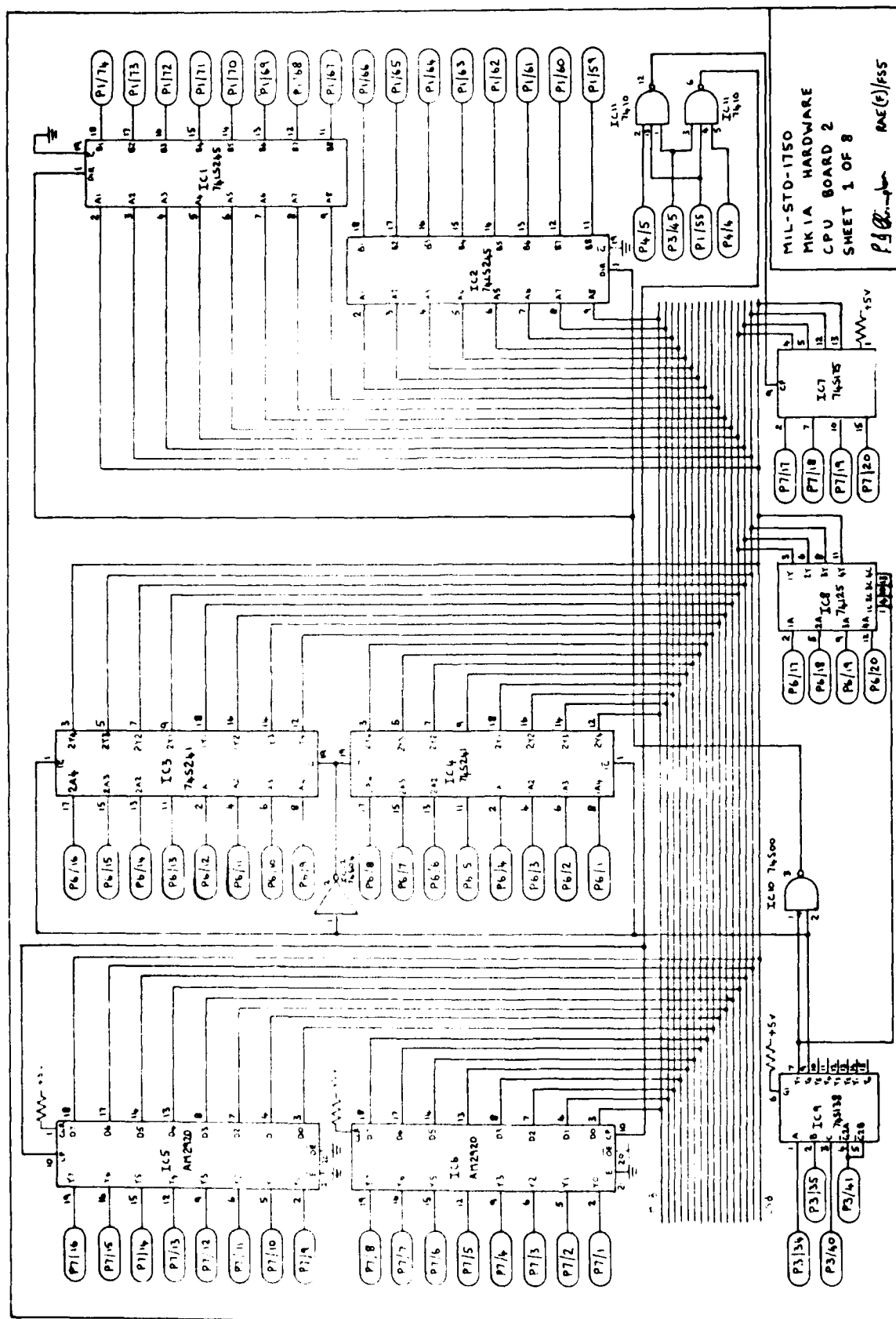


Fig 12

Fig 13

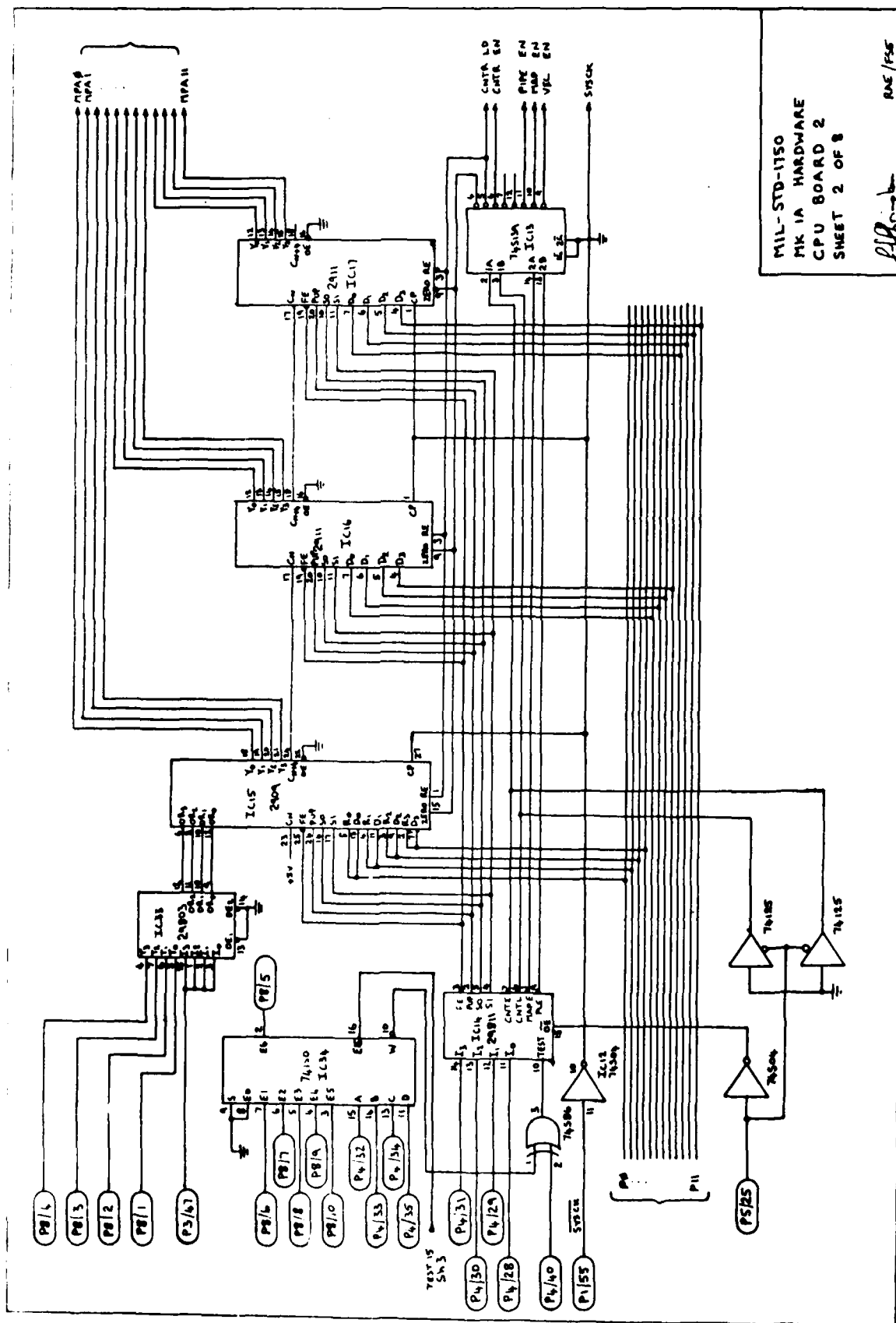
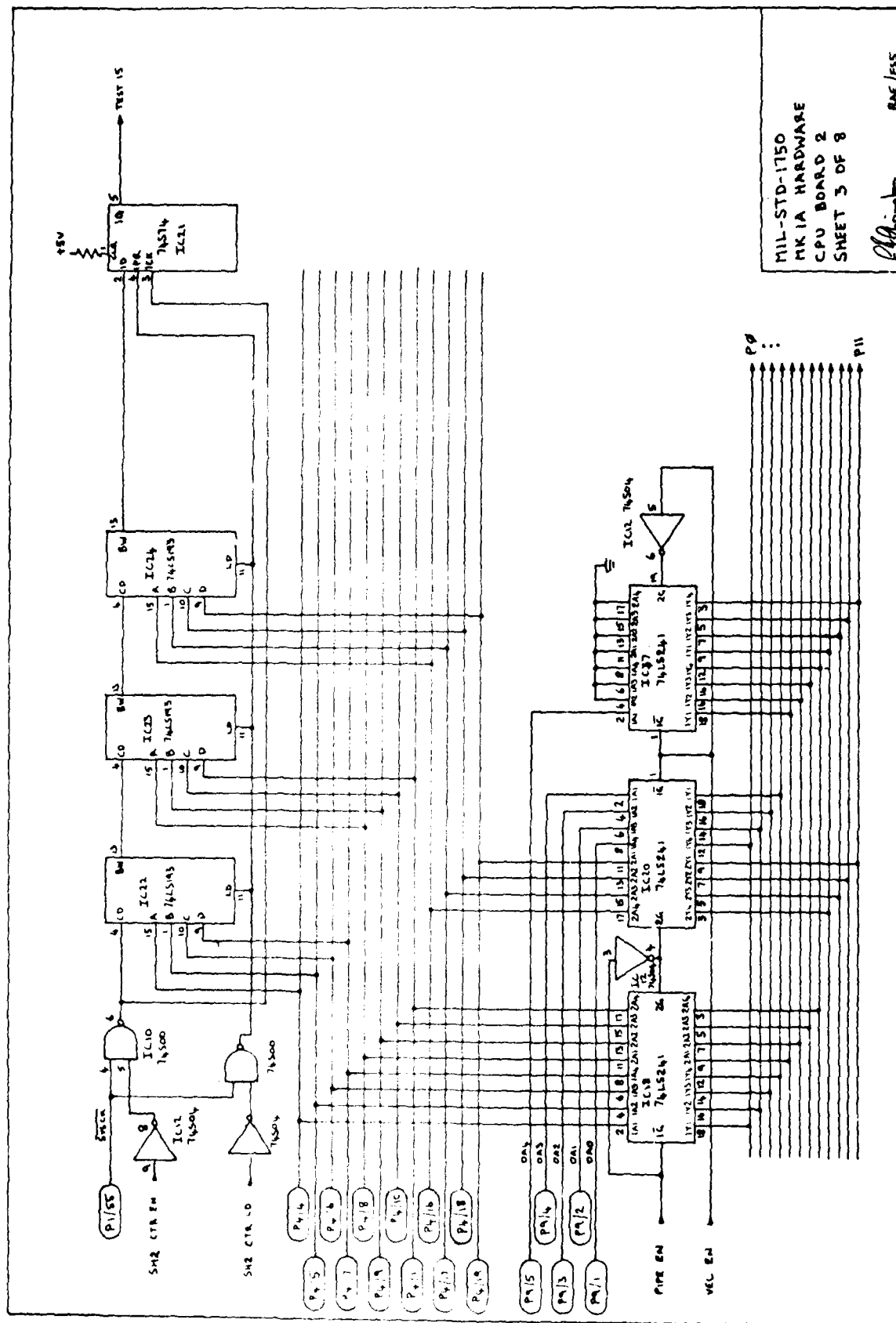
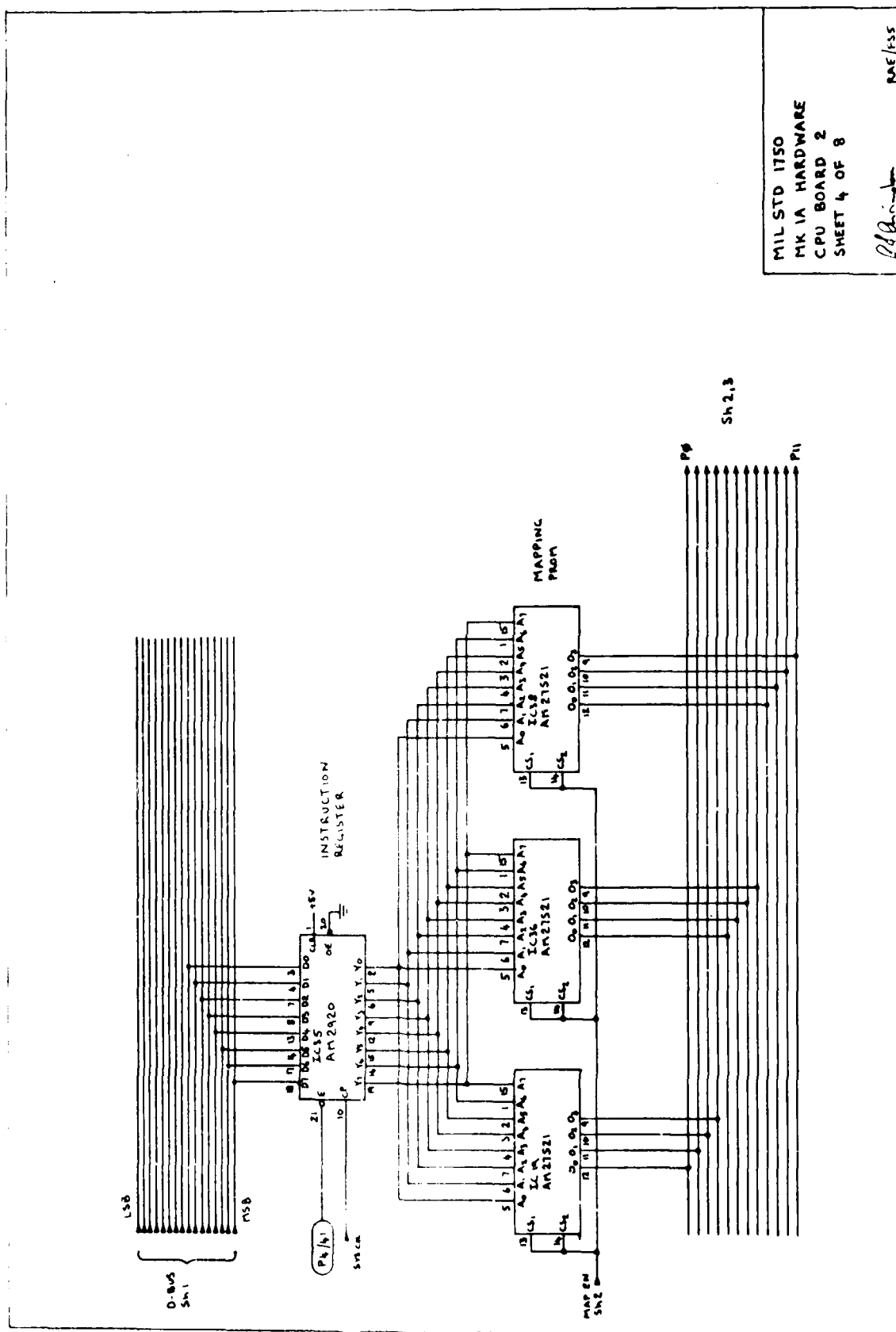


Fig 14





MIL STD 1750
MK 1A HARDWARE
CPU BOARD 2
SHEET 4 OF 8

RAE/HSS

Fig 16

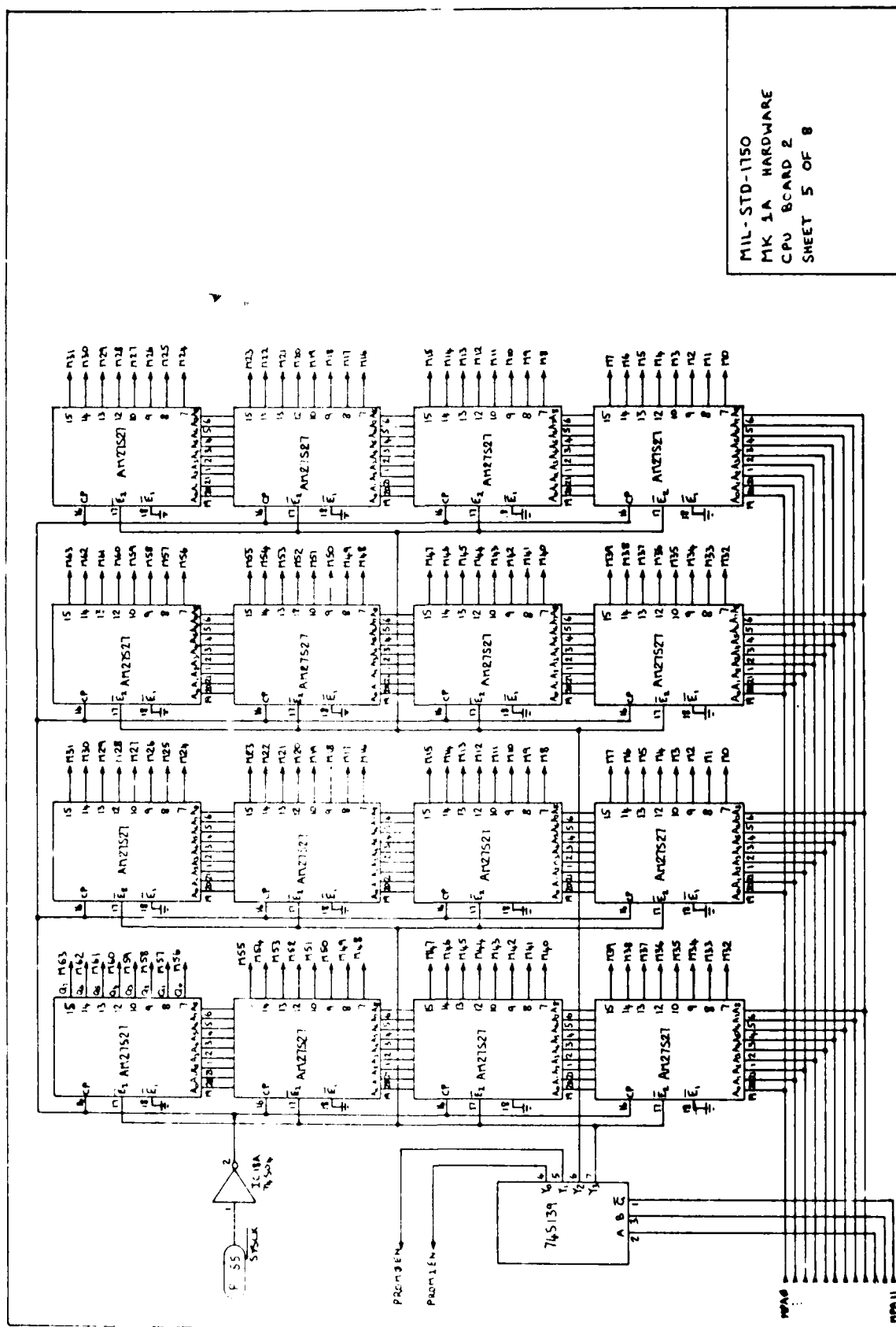
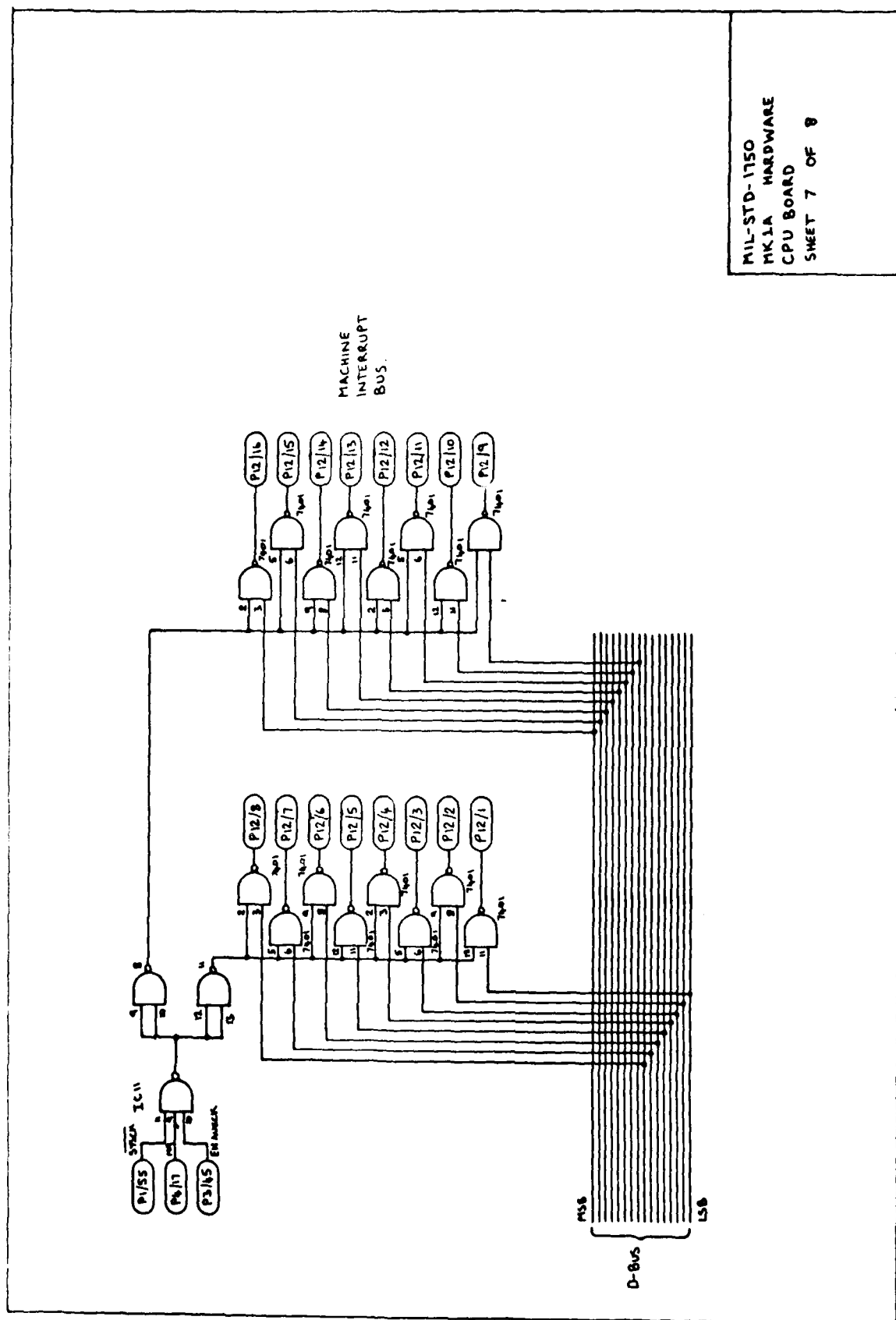


Fig 16

Fig 17

MIL-STD-1750
MK1A HARDWARE
CPU BOARD
SHEET 7 OF 8



MIL-STD-1750
MK 1A HARDWARE
CPU BOARD 2
SHEET 8 OF 8

TM PS 403

1919

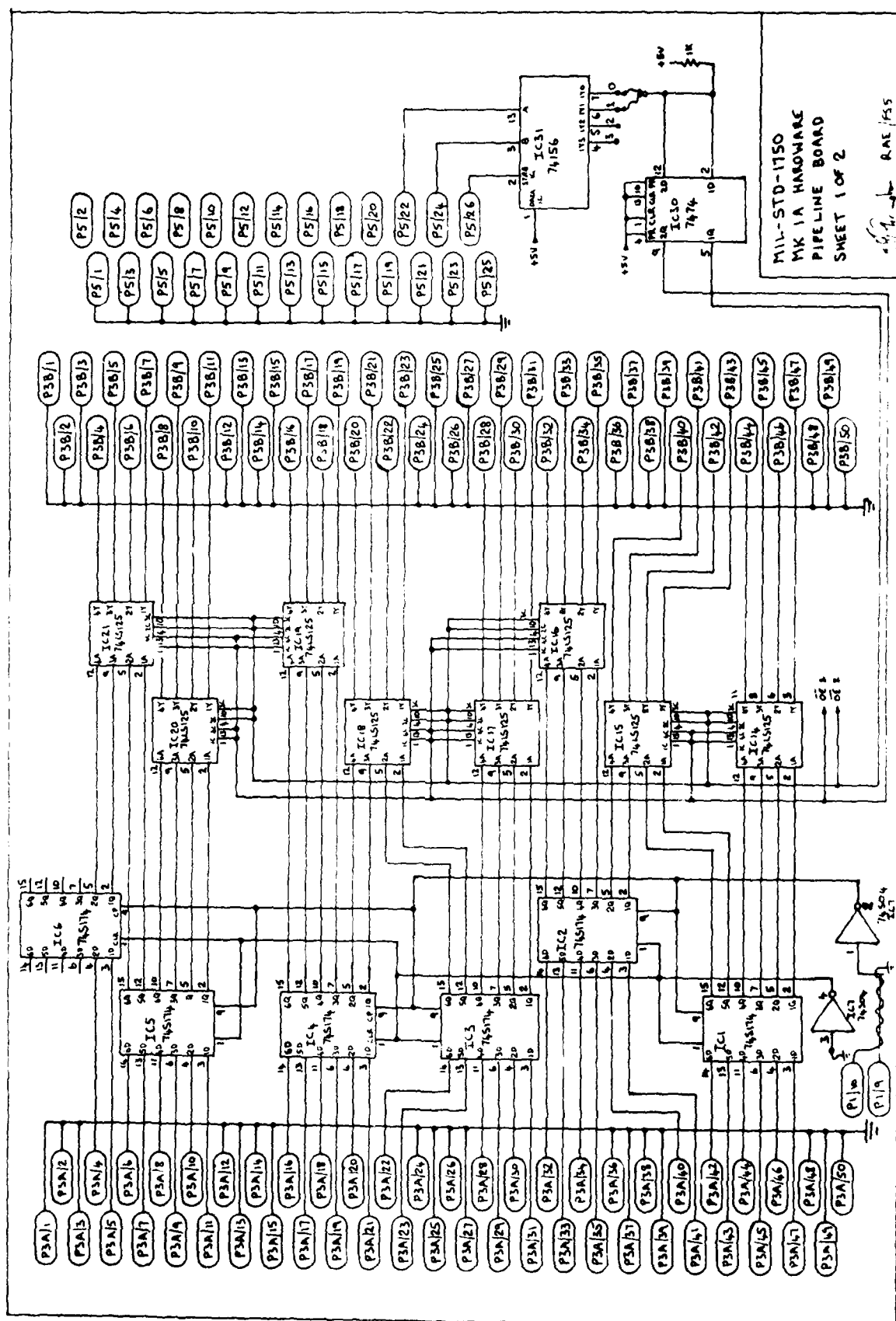


Fig 20

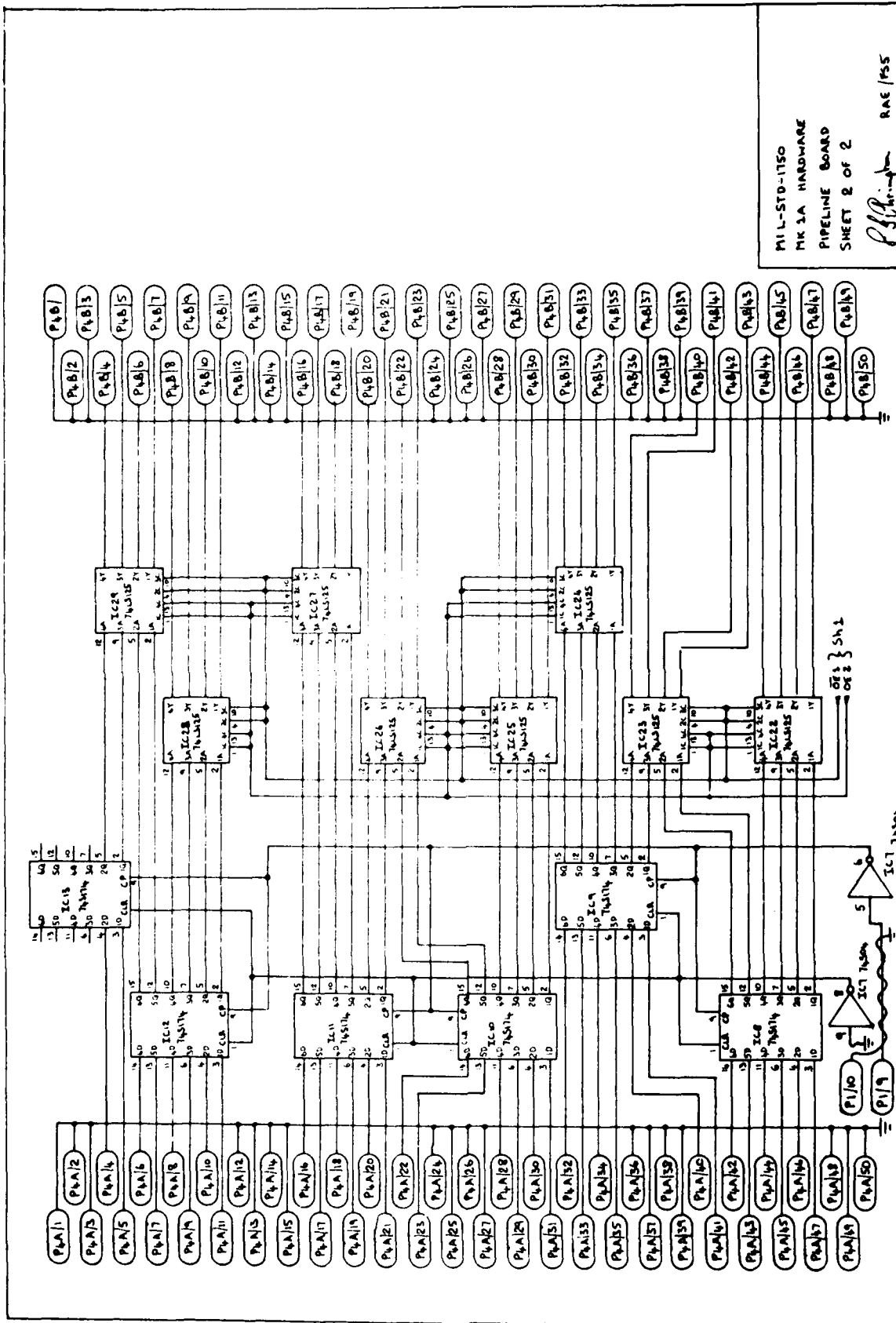


Fig 20



Fig 22

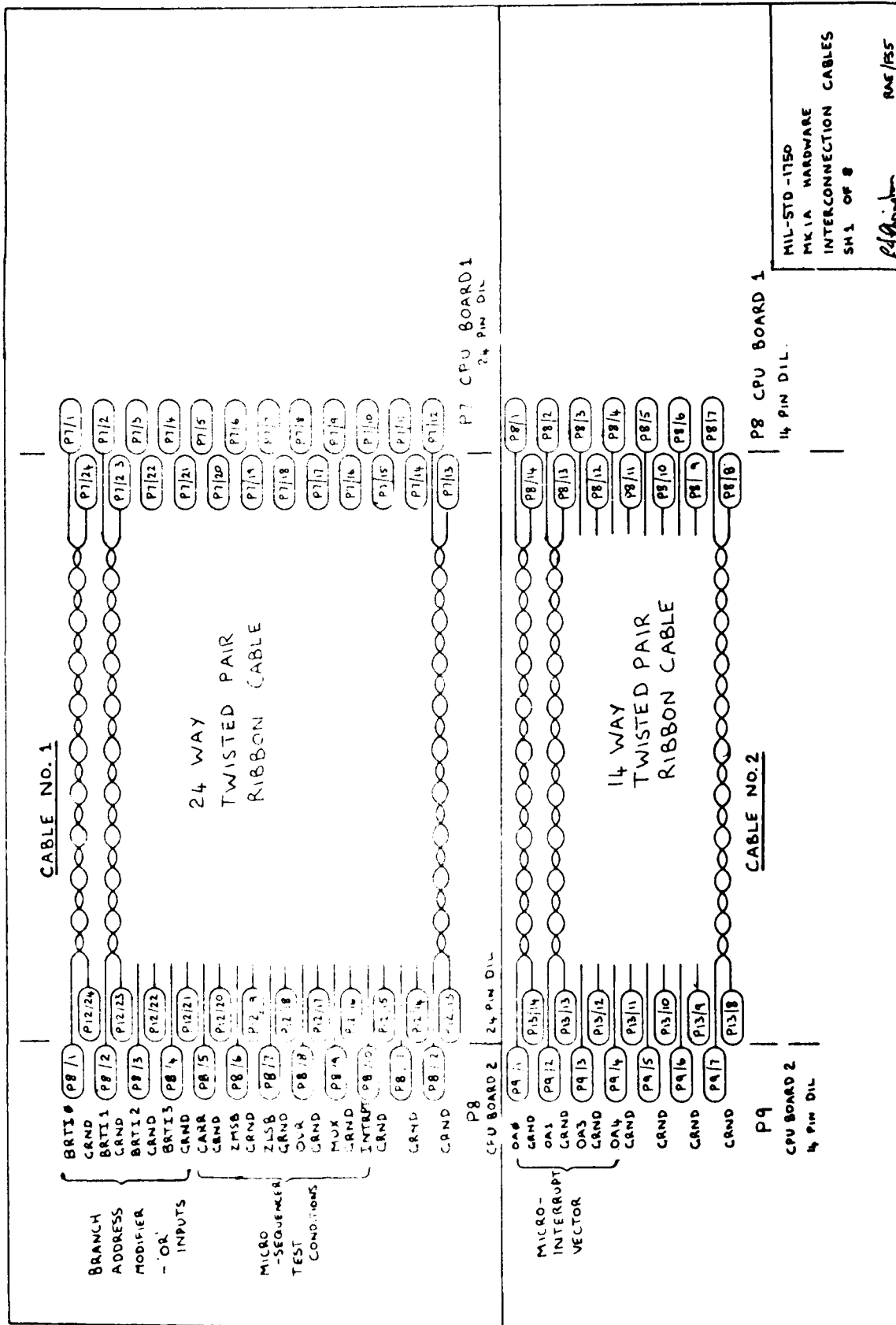


Fig 22

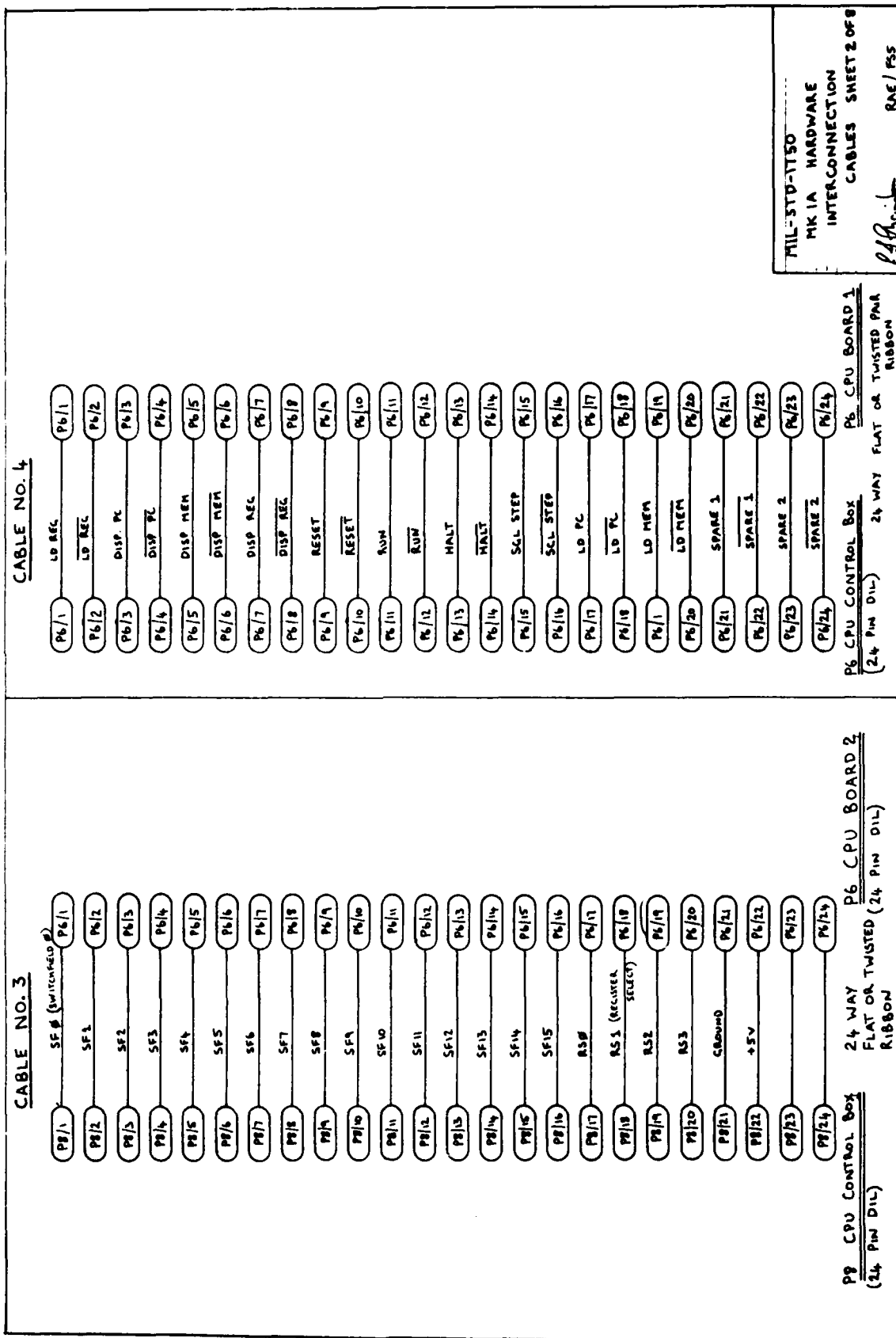


Fig 23

Fig 24

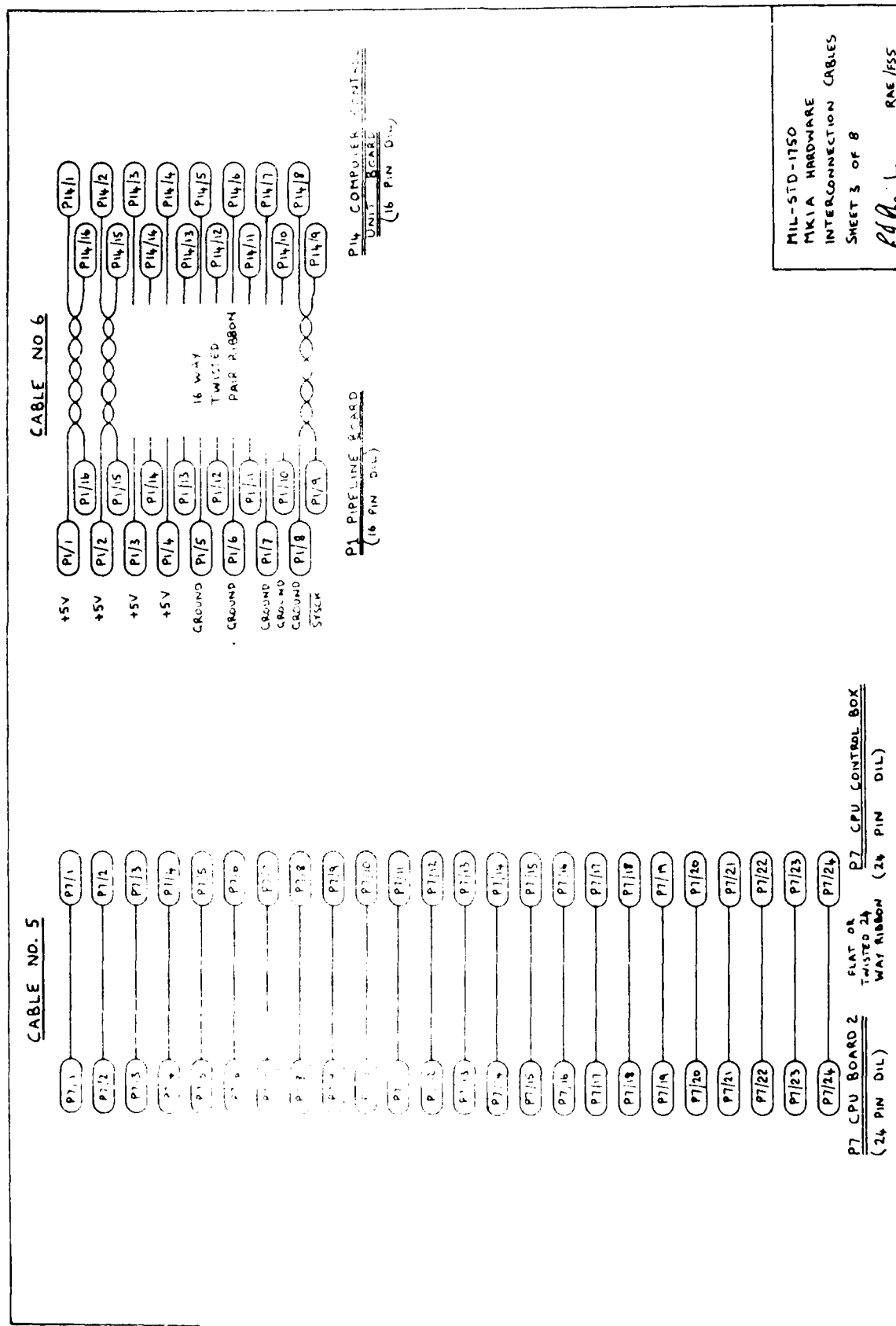


Fig 24

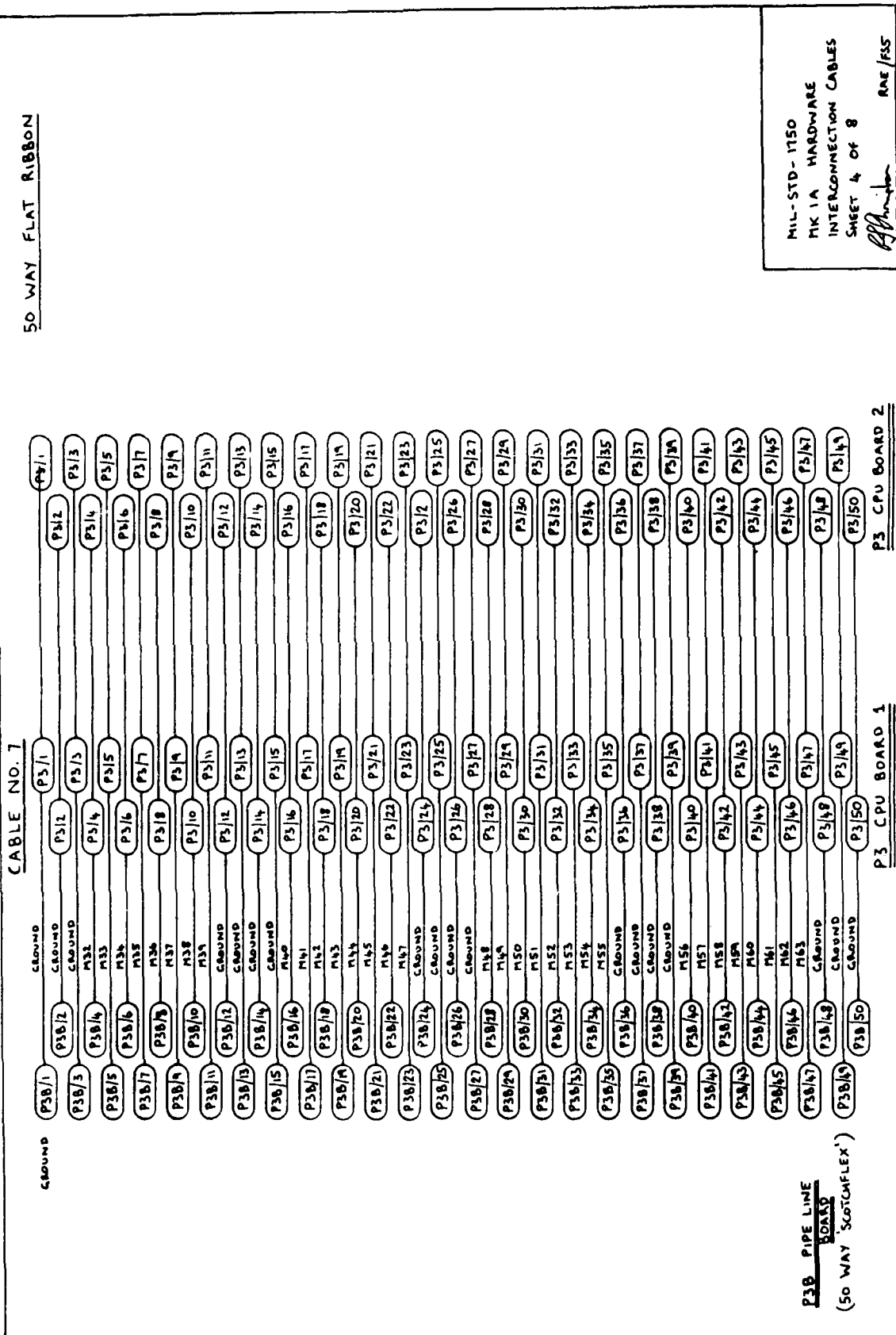


Fig 25

Fig 26

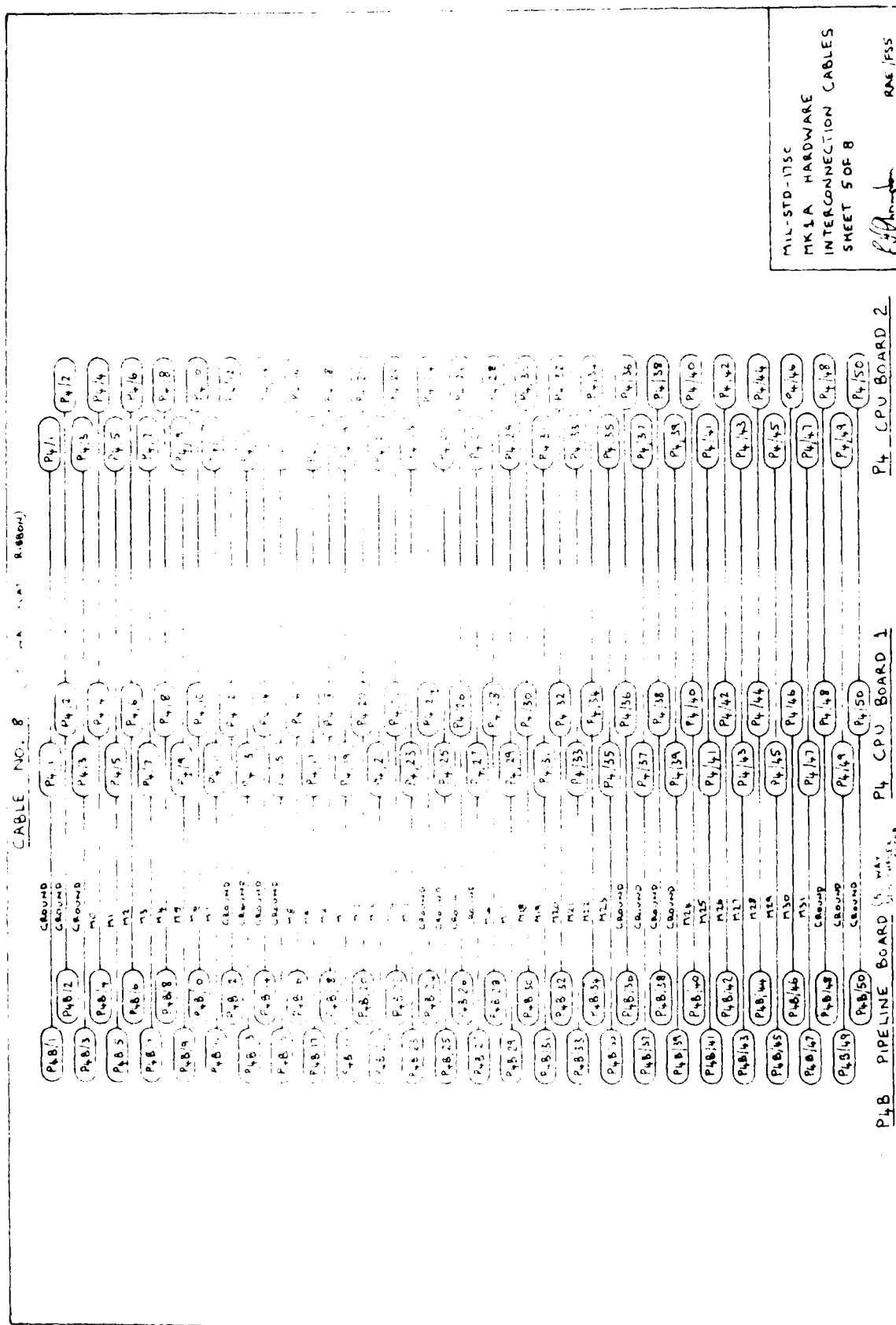


Fig 26

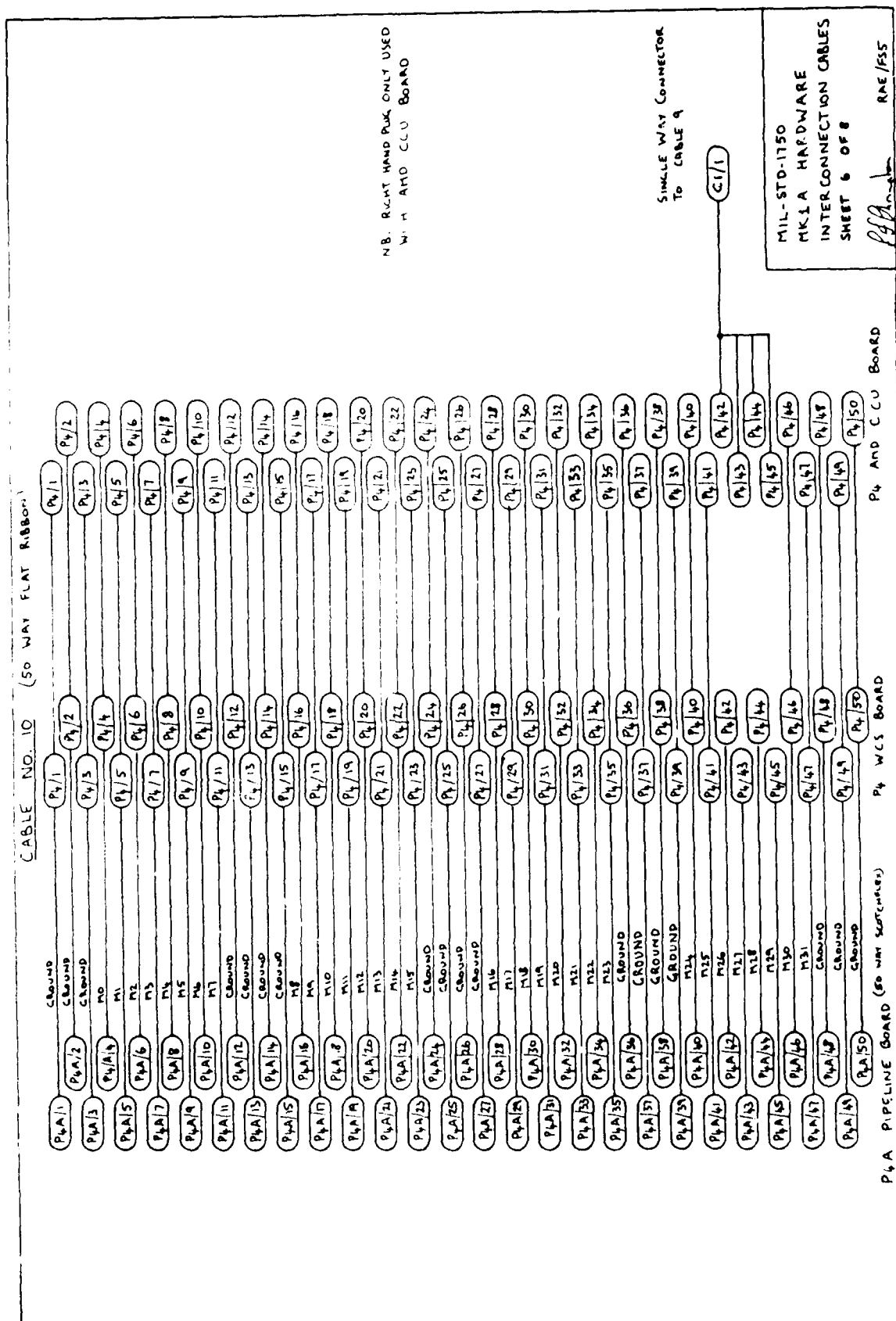


Fig 28

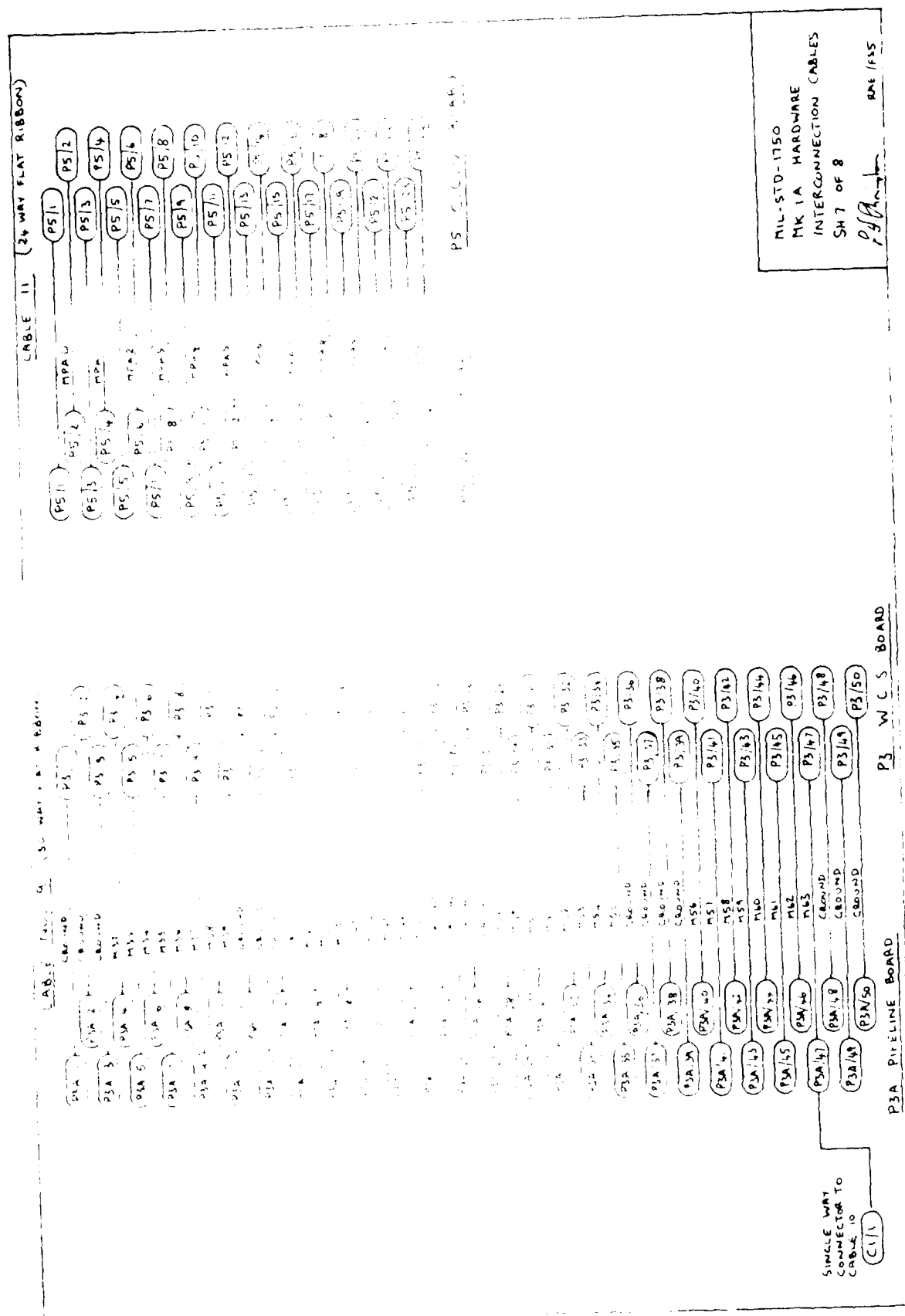


fig 28

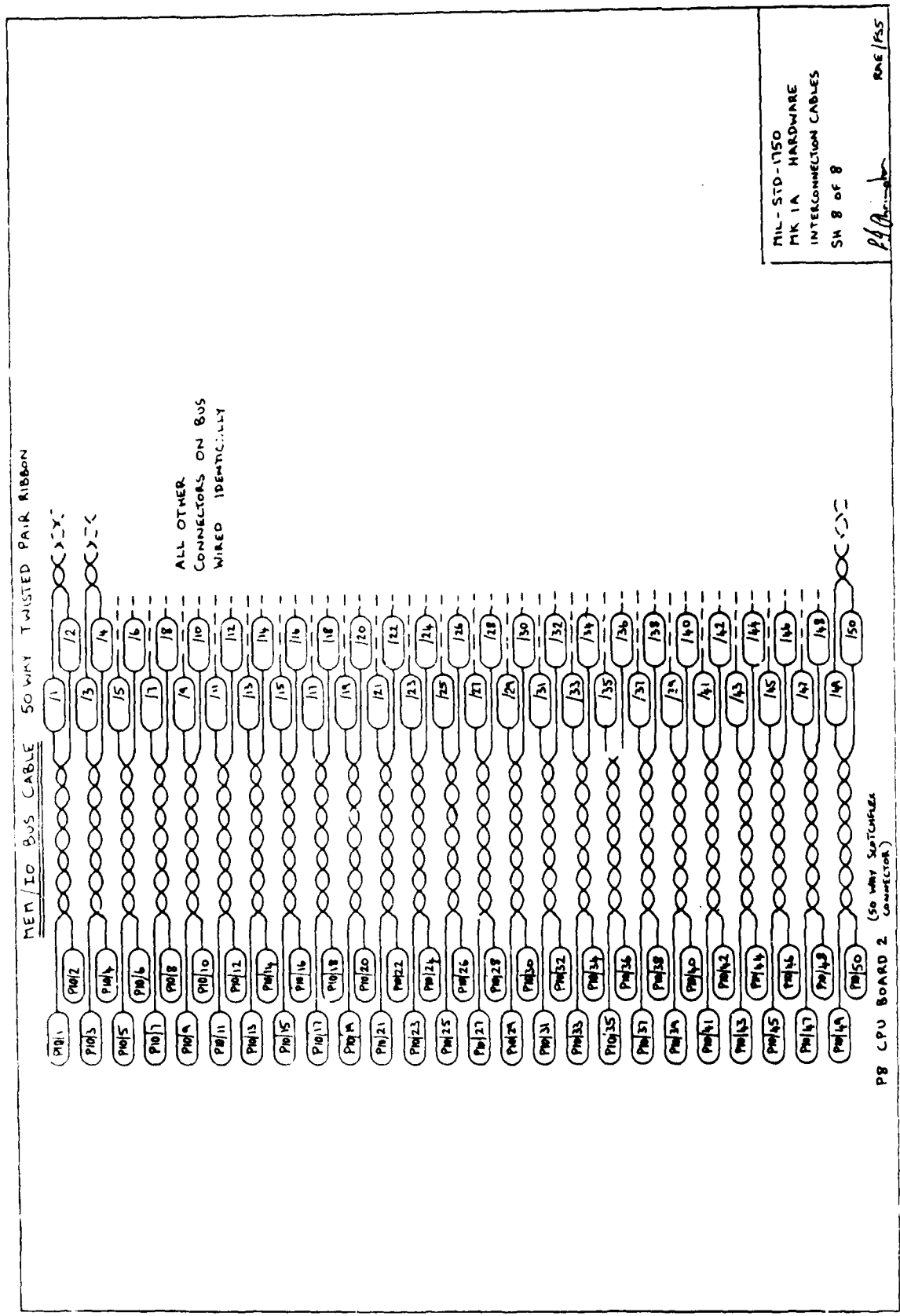


Fig 29

REPORT DOCUMENTATION PAGE

Overall security classification of this page

UNCLASSIFIED

As far as possible this page should contain only unclassified information. If it is necessary to enter classified information, the box above must be marked to indicate the classification, e.g. Restricted, Confidential or Secret.

| | | | | | |
|--|--|--|---|--------------------------|--------------|
| 1. DRIC Reference
(to be added by DRIC) | 2. Originator's Reference
RAE TM FS 403 | 3. Agency Reference
N/A | 4. Report Security Classification/Marking
UNCLASSIFIED | | |
| 5. DRIC Code for Originator
7673000W | | 6. Originator (Corporate Author) Name and Location
Royal Aircraft Establishment, Farnborough, Hants, UK | | | |
| 5a. Sponsoring Agency's Code
N/A | | 6a. Sponsoring Agency (Contract Authority) Name and Location
N/A | | | |
| 7. Title An implementation of Mil-Std-1750 airborne computer instruction set architecture | | | | | |
| 7a. (For Translations) Title in Foreign Language | | | | | |
| 7b. (For Conference Papers) Title, Place and Date of Conference | | | | | |
| 8. Author 1. Surname, Initials
Shrimpton, S.J. | 9a. Author 2 | 9b. Authors 3, 4 | | 10. Date
May 1981 | Pages
149 |
| | | | | Refs.
4 | |
| 11. Contract Number
N/A | 12. Period
N/A | 13. Project | | 14. Other Reference Nos. | |
| 15. Distribution statement
(a) Controlled by –
(b) Special limitations (if any) – | | | | | |
| 16. Descriptors (Keywords) (Descriptors marked * are selected from TEST)
Bit slice. Mil-Std-1750. | | | | | |
| 17. Abstract

This Memorandum describes the design of a processor implementing the Mil-Std-1750 Airborne Computer Instruction Set Architecture, using Advanced Micro Devices 2901 bit-slice microprocessor devices. The aspects of the hardware design and microcode specific to Mil-Std-1750 are discussed and reviewed in the light of the experience gained. A full listing of the AMD 'AMDASM' micro assembler definition file and microcode source text is included, together with full hardware documentation. | | | | | |

FS910/