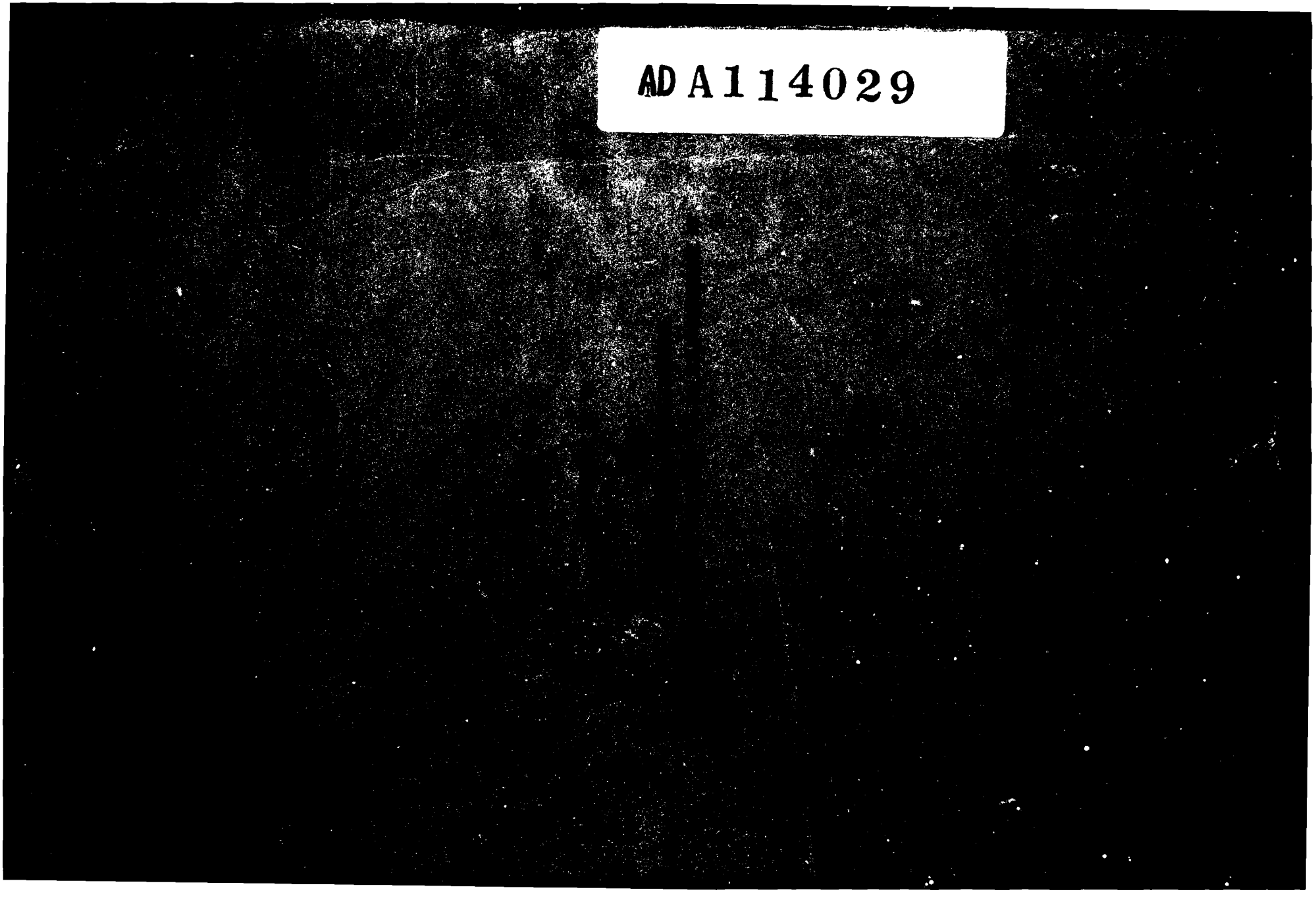
AD-A114 029 ROYAL AIRCRAFT ESTABLISHMENT FARNBOROUGH (ENGLAND) F/G 9/2
AN IMPLEMENTATION OF MIL-STD-1750 AIRBORNE COMPUTER INSTRUCTION--ETC(U)
MAY 81 S J SHRIMPTON
UNCLASSIFIED RAE-FS-403 DRIC-BR-80502 NL


AD A114029

ROYAL AIRCRAFT ESTABLISHMENT

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.

LIST OF CONTENTS

1 INTRODUCTION

1.1 The Mil-Std-1750 standardisation exercise

Mil-Std-1750, the current version of which is Mil-Std-1750A[1], 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 Memorandum[2] 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 devices[3] 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 ER0 to ER15. The registers internal to the 2901 chips are referred to as internal registers and are numbered IR0 to IR15.

The availability now of spare registers prompted the placing of the machine instruction counter in the external register file as ER0. 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 subroutines. 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 repartitioning 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 micro-code 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 microcode 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 ir 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 I7 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 I8 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 (C0 = 1, C1 = 0)
Write cycle (C0 = 0, C1 = 1)
Read-modify-write cycle (C0 = 1, C1 = 1)
Refresh cycle (C0 = 0, C1 = 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, *ie* 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 CI line and a high appears on P4/31, the CO 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.

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, *ie* 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 auxilliary 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 auxilliary 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 × 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 I0, I1, I2, I3, I4, I5, I6, I7, I8, I9, I11, I12 from the bits within the field.

I0 = bit 0
I1 = bit 1
I2 = bit 2
I3 = bit 3
I4 = bit 4

I5 = bit 5
I6 = bit 6
I7 = bit 7
I8 = bit 8
I9 = 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 auxilliary 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:

M51	M15	Type of cycle
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, *ie* 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:

Bit pattern in field	Source for A register address
00	Bits 8 through 11 of the instruction stored in the instruction register (*ie* the GR1 field of the 1750 instruction).
01	Bits 12 through 15 of the instruction stored in the instruction register (*ie* 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 (*ie* 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:

Bit pattern	D-bus source
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 0 - 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 C

AMD 'AMDASM' DEFINITION FILE

```
TITLE MIL-STD-1750 PHASE 1 FORMAT DEFINITION
TITLE MIL STD 1750 PHASE 1 BIG FORMAT DEFINITION
WORD 64
;
;MICROPROGRAM FIELD FORMAT
;------------------------
;
;BIT FUNCTION                                        CONDITION FOR ACTION
;NO.
;-----------------------------------------------------------------------
;M0  ALTERNATIVES:
;M1  (1) STATUS AND SHIFT INSTRUCTION                CK MAC STAT OR CK MICRO
;                                                    STAT. ACTIVE (LOW) OR SHIFT
;                                                    FUNCTION SELECTED BY ALU.
;M2  (2) 12 BIT BRANCH ADDRESS                       BRANCH ADDRESS REQUIRED BY
;                                                    AM 29811
;M3  (3) BITS 0 TO 11 OF 16 BIT DATA FIELD           DATA FIELD SOURCED ONTO
;                                                    D BUS
;M4  (4) AUX CLOCK CONTROL FIELD                     AUX CLK ENABLE ACTIVE.
;M5 M0 - CK DISPLAY
;M6 M1 - CK STATE DISPLAY
;M7 M2 - CK STATUS
;M8 M3 - CK MACH. INT.
;M9
;M10
;M11
;-----------------------------------------------------------------------
;M12  (1) CARRY SELECT FIELD                         ALWAYS AFFECTS INT.CARR LATCH
;M13                                                 MACH.CARR. OR MIC.CARR.
;M14                                                 IF CORRESPONDING ENABLE BITS
;     (2) BITS 12 THROUGH 14 OF DATA FIELD           EFFECTIVE IF DATA FIELD IS
;                                                    ENABLED
;                                                    ARE ACTIVE.
;-----------------------------------------------------------------------
;M15  (1) I/O PORT CONTROL BIT 1                     EFFECTIVE IF CK PORT HIGH
;       (2) BIT 15 OF DATA FIELD                     EFFECTIVE IF DATA FIELD IS
;                                                    SOURCED ONTO D BUS
;-----------------------------------------------------------------------
;M16    NEXT ADDRESS CONTROL                         ALWAYS APPLICABLE
;M17
;M18
;M19
;-----------------------------------------------------------------------
;M20    (1) CONDITION TEST SELECT                    TEST COND. REQUIRED BY AM29811
;M21    (2) NOT ENABLE Z - M20
;M22        NOT ENABLE C - M21                       MACHINE STATUS ENABLED
;M23        NOT ENABLE N - M22
;           NOT ENABLE P - M23
;-----------------------------------------------------------------------
;M24    CONDITION POLARITY TEST                      APPLICABLE IF TEST CONDITION
;                                                    REQUIRED BY AM 29811
;-----------------------------------------------------------------------
;M25    CK INSTRUCTION REG                           ALWAYS APPLICABLE
;-----------------------------------------------------------------------
;M26    ALU SOURCE CONTROL                           ALWAYS APPLICABLE
;M27
;M28
;-----------------------------------------------------------------------
;M29    ALU MS BYTE FUNCTION                         ALWAYS APPLICABLE
;M30
;M31
;-----------------------------------------------------------------------
;M32    ALU LS BYTE FUNCTION
;M33                                                 ALWAYS APPLICABLE
;M34
;-----------------------------------------------------------------------
;M35    ALU DESTINATION CONTROL                      ALWAYS APPLICABLE
;M36
;M37
;-----------------------------------------------------------------------
;M38    EXTERNAL REGISTER SELECT                     APPLICABLE IF EX.REG IS SOURCED
;M39                                                 ONTO D BUS OR 2901 D INPUT
;M40                                                 OR IF REG 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 B SELECT CONTROL        APPLICABLE IF REG B IS USED
;M45    (2) MICRO INT. CONTROL BITS 3 AND 4          APPLICABLE IF CK MIC.INT.
;                                                    IS HIGH
;-----------------------------------------------------------------------
;M46  (1)INTERNAL REGISTER A SELECT MOD.             APPLICABLE IF REG A IS USED
;M47  (2) MAC.INT.CONTROL BITS 1 AND 2               APPLICABLE IF CK MACH.INT.
;                                                    IS HIGH
;-----------------------------------------------------------------------
;M48  (1) INTERNAL REGISTER B SELECT MOD.             APPLICABLE IF REG.B IS USED
;M49  (2) MACH.INT.CONTROL BITS 3 AND 4               APPLICABLE IF CK MACH.INT.
;                                                    IS HIGH
;-----------------------------------------------------------------------
;M50    2901 D INPUT SOURCE SELECT                   APPLICABLE IF D INPUT IS SELECTED
;                                                    AS R OR S SOURCE FOR ALU
;-----------------------------------------------------------------------
;M51      PORT CONTROL BIT 2                         APPLICABLE IF CK PORT IS HIGH
;-----------------------------------------------------------------------
;M52      MACH.STAT. NOT ENABLE                      ALLWAYS APPLICABLE
;-----------------------------------------------------------------------
;M53 MICRO STATUS NOT ENABLE                         ALWAYS APPLICABLE
;M54   D BUS SOURCE CONTROL FIELD                    ALWAYS APPLICABLE
;M55
;M56
;M57
;-----------------------------------------------------------------------
;M58  CK ALU                                  ALWAYS APPLICABLE
;M59  CK REG                                          ALWAYS APPLICABLE
```

```
;M60  CK MIC. INTERRUPT                                  ALWAYS APPLICABLE
;M61  ENABLE AUX CLOCK FIELD                             ALWAYS APPLICABLE
;M62  CK PORT                                            ALWAYS APPLICABLE
;-------------------------------------------------------------------------
;M63  AM29803 CONTROL                                ALWAYS APPLICABLE
;-------------------------------------------------------------------------
;
;
;
TITLE T M E  A M 2 9 0 0  F A M I L Y  M N E M O N I C S
;*********************************************************
;
;
;13 DECEMBER 1976 JBM
;UPDATED SEPT 26, 1977
;
;AM2901 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
;
;AM2901 SOURCE OPERANDS (R S)
;
AQ:EQUQ#0
AB:EQUQ#1
ZQ:EQUQ#2
ZB:EQUQ#3
ZA:EQUQ#4
DA:EQUQ#5
DQ:EQUQ#6
DZ:EQUQ#7
;
;AM2901 ALU FUNCTIONS (R FUNCTION S)
;
ADD:EQUQ#0
SUBR:EQUQ#1
SUBS:EQUQ#2
OR:EQUQ#3
AND:EQUQ#4
NOTRS:EQUQ#5
EXOR:EQUQ#6
EXNOR:EQUQ#7
;
;AM2901 DESTINATION CONTROL
;
QREG:EQUQ#0
NOP:EQUQ#1
RAMA:EQUQ#2
RAMF:EQUQ#3
RAMQD:EQUQ#4
RAMD:EQUQ#5
RAMQU:EQUQ#6
RAMU:EQUQ#7
;
;AM2911 INSTRUCTION SET
;
JZ:EQUH#0;JUMP TO ADDRESS ZERO
CJS:EQUH#1;CONDITIONAL JUMP TO SUBROUTINE WITH JUMP
;ADDRESS IN THE PIPELINE REGISTER
JMAP:EQUH#2;JUMP TO ADDRESS AT MAPPING PROM OUTPUT
CJP:EQUH#3;CONDITIONAL JUMP TO ADDRESS IN PIPELINE
;REGISTER
PUSH:EQUH#4;PUSH STACK AND CONDITIONALLY LOAD COUNTER
JSRP:EQUH#5;JUMP TO SUBROUTINE WITH STARTING ADDRESS
;CONDITIONALLY SELECTED FROM THE AM2911
;R-REGISTER OR PIPELINE ADDRESS
CJV:EQUH#6;CONDITIONAL JUMP TO VECTOR ADDRESS
JRP:EQUH#7;JUMP TO ADDRESS CONDITIONALLY SELECTED FROM
;AM2911 R-REGISTER OR PIPELINE REGISTER
RFCT:EQUH#8;REPEAT LOOP IF COUNTER IS NOT EQUAL TO ZERO
RPCT:EQUH#9;REPEAT PIPELINE ADDRESS IF COUNTER IS NOT
;EQUAL TO ZERO
CRTN:EQUH#A;CONDITIONAL RETURN FROM SUBROUTINE
CJPP:EQUH#B;CONDITIONAL JUMP TO PIPELINE ADDRESS AND POP
;STACK
LDCT:EQUH#C;LOAD COUNTER AND CONTINUE
LOOP:EQUH#D;TEST END OF LOOP
CONT:EQUH#E;CONTINUE TO NEXT ADDRESS
JP:EQUH#F;JUMP TO PIPELINE REGISTER ADDRESS
;
;
;AM2914 INSTRUCTION SET
;
MCLR:EQUH#0;MASTER CLEAR
CLRIN:EQUH#1;CLEAR ALL INTERRUPTS
CLRMB:EQUH#2;CLEAR INTERRUPTS FROM M-BUS
CLRMR:EQUH#3;CLEAR INTERRUPTS FROM MASK REGISTER
CLRVC:EQUH#4;CLEAR INTERRUPT FROM LAST VECTOR READ
RDVC:EQUH#5;READ VECTOR
RDSTA:EQUH#6;READ STATUS REGISTER
RDM:EQUH#7;READ MASK REGISTER
SETM:EQUH#8;SET MASK REGISTER
```

```
LDSTA:EQUH#9;LOAD STATUS REGISTER
BCLRM:EQUH#A;BIT CLEAR MASK REGISTER
BSETM:EQUH#B;BIT SET MASK REGISTER
CLRM:;QUH#C;CLEAR MASK REGISTER
DISIN:EQUH#D;DISABLE INTERRUPT REQUEST
LDM:EQUH#E;LOAD MASK REGISTER
ENIN:EQUH#F;ENABLE INTERRUPT REQUEST
;
;
;
;INTERNAL REGISTER SELECT CONTROL PATTERNS
;---------------------------------------
;
GR1:    EQU     B#00    ;SELECTS BITS 8 THROUGH 11 OF INSTRUCTION WORD AS
                        ;REGISTER SELECT
GR2:    EQU     B#01    ;SELECTS BITS 12 THROUGH 15
BASE:    EQU     B#10        ;SELECTS BASE REGISTER USING BITS 5 AND 6
RZERO:  EQU     B#11        ;SELECTS REGISTER ZERO
;
;
;CARRY SELECT CONTROL PATTERNS
;---------------------------
;
CONE:   EQU     B#000       ;SELECTS LOGICAL ONE AS CARRY TO BE STORED
CZERO:   EQU     B#001       ;SELECTS LOGICAL ZERO AS CARRY TO BE STORED
CQLSB:  EQU     B#010       ;SELECTS LSB OUTPUT OF Q REGISTER AS CARRY
CRMSB:  EQU     B#011       ;SELECTS MSB OUTPUT OF RAM AS CARRY
CRLSB:  EQU     B#100       ;SELECTS LSB OUTPUT OF RAM AS CARRY
CCAR:   EQU     B#101       ;SELECTS CARRY OUTPUT FROM ALU
CRMID:  EQU     B#110       ;SELECTS RAM MIDPOINT
CQMID:  EQU     B#111      ;SELECTS Q MIDPOINT
;
; NB. THERE ARE THREE CARRY LATCHES IN THE ME ONE IMPLEMENTATION.
;     ONE IS EXTERNAL TO THE 2904 AND IS CLOCKED ON EVERY MICROCYCLE.
;     THE OTHERS ARE THE CARRY BITS IN THE MICRO AND MACHINE STATUS REGISTERS
;     WHICH ARE WITHIN THE 2904 CHIP AND ARE ONLY CLOCKED 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.
;
;
SNONE:  EQU     B#0000      ;DISABLES ALL CPU OUTPUTS TO D BUS
SALU:   EQU     B#0001      ;SOURCES ALU ONTO D BUS
SMEM:   EQU     B#0010       ;SOURCES MEMORY DATA REGISTER ONTO DBUS
SRFG:   EQU     B#0011       ;SOURCES REGISTER FILE ONTO D BUS
SFAULT: EQU     B#0100      ;SOURCES FAULT REGISTER
```

```
SSTATUS:  EQU     B#0101     ;SOURCES MACHINE STATUS REGISTER
SMANDT:  EQU     B#0110      ;SOURCES MANUAL DATA INPUT
SMANRG:  EQU     B#0111      ;SOURCES MANUAL REGISTER SELECT FIELD
SDATA:   EQU     B#1000      ;SOURCES MICROPROGRAM DATA FIELD
SSTATE:  EQU     B#1001      ;SOURCES CPU STATE DISPLAY
STIMA:   EQU     B#1010      ;SOURCES TIMER A
STIMB:   EQU     B#1011      ;SOURCES TIMER B
;
;
;TEST CONDITION SELECT PATTERNS
;-----------------------------
;
; THESE CONTROL THE SOURCE OF THE TEST INPUT TO THE 29811 NEXT ADDRESS
; CONTROL UNIT.
;
TZERO:   EQU     B#0000     ;PUTS LOGICAL ZERO ON TEST INPUT
ZMSB:    EQU     B#0001     ;TESTS FOR ZERO MS BYTE
ZLSB:    EQU     B#0010     ;TESTS FOR ZERO LS BYTE
OVR:     EQU     B#0011     ;TESTS FOR OVERFLOW SET
MUX:     EQU     B#0100     ;TESTS FOR OUTPUT OF AM2904 MULTIPLEXER
INTRPT:  EQU     B#0101     ;TESTS MICROINTERRUPT REQUEST LINE
CARR:    EQU     B#0110     ;TESTS OUTPUT OF EXTERNAL CARRY LATCH
COUNT:   EQU     B#1111     ;TESTS COUTER ZERO
;
;
;TEST POLARITY SELECT PATTERNS
;----------------------------
;
TRUE:    EQU     B#1
FALSE:   EQU     B#0
;
;
;D INPUT TO 2901 SELECT CONTROL PATTERNS
;--------------------------------------
;
ABUS:    EQU     B#0  ;SELECTS THE ADDRESS BUS (OUTPUT OF EXTERNAL REGISTER FILE)
DBUS:    EQU     B#1  ;SELECTS THE INTERNAL TRISTATE DATA BUS
;
;
;FORMAT DEFINITIONS
;------------------
;-------------------------------------------------------------------------------
;
;  BADDRESS
;  ********
;
; 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, 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
;      STATSHFT, DATINSRT, ENAUICK, AUICK
;
BADDRESS:    DEF   52I,12V%:
;
;-------------------------------------------------------------------------
;
;  STATSHFT
;  ********
;
;THIS FORMAT IS USED TO ENABLE THE BINARY PATTERN CONTROLLING THE AM2904
;CHIP TO BE INSERTED INTO THE MICROPROGRAM WORD.
;
; VARIABLE FIELD SUBSTITUTIONS:
;   (1) A 12 BIT BINARY NUMBER FORMING THE CONTROL
;       WORD FOR THE 2904 CHIP.
;             BIT 11 (MSB)  =   I12
;                 10        =   I11
;                 9         =   I0
;                 8         =   I8
;                 ....      =   ....
;                 0         =   I9
;   SEE AM2904 DATA SHEET FOR DEFINITION OF I0 TO I12.
;
;OVERLAY RESTRICTIONS:
;     CANNOT BE OVERLAID WITH:
;        BADDRESS, DATINSRT,ENAUICK,AUICK.
;
;
STATSHFT:    DEF   52I,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 2904 ON EVERY CLOCK CYCLE.
;
;VARIABLE FIELD SUBSTITUTIONS:
;    (1) A THREE BIT BINARY PATTERN APPROPRIATE TO SELECT THE REQUIRED
;        INPUT TO THE CARRY SELECT MULTIPLEXER (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
;       DATINSRT
;
CARRYSEL:    DEF   49I,3V,12I
;
DATINSRT:    DEF   48I,16VX:D0
;-------------------------------------------------------------------------
;
; RFRSH
; *****
;
;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 (CKPORT) IS HELD HIGH.
;
; THERE ARE NO VARIABLE FIELDS.
;
;OVERLAY RESTRICTIONS:
;   MAY NOT BE OVERLAID WITH DATINSRT, READ, WRITE, RMWRT.
;
RFRSH:       DEF   12I,B#0,35I,B#0,15I
;
;-------------------------------------------------------------------------
;
; READ
; ****
;
;THIS FORMAT IS THE SAME AS RFRSH EXCEPT THAT A READ CYCLE IS INITIATED.
;
;OVERLAY RESTRICTIONS:
;     MAY NOT BE OVERLAID WITH
;       DATINSRT, RFRSH, WRITE, RMWRT.
;
READ:        DEF     12I,B#1,35I,B#0,15I
;-------------------------------------------------------------------------
;
; WRITE
; *****
;
;THIS FORMAT IS THE SAME AS RFRSH EXCEPT THAT A WRITE CYCLE IS PRODUCED.
;
;OVERLAY RESTRICTIONS:
;   MAY NOT BE OVERLAID WITH
;        RFRSH, READ,RMWRT.
;
WRITE:       DEF     12I,B#0,35I,B#1,15I
;-------------------------------------------------------------------------
;
; RMWRT
; *****
;
; THIS FORMAT IS THE SAME AS RFRSH EXCEPT THAT A READ/MODIFY/WRITE
; CYCLE IS PRODUCED.
;
; OVERLAY RESTRICTIONS:
;    MUST NOT BE OVERLAID WITH
```

```
;          DATINSRT, RFRSH, READ, WRITE.

RMWRT:     DEF     12I,B#1,3SI,B#1,15I

;-------------------------------------------------------------------------

;  MAC
;  ***

; 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
;          29803 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
;          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 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.
;          (TZERO, ZMSB, ZLSB, OVR, MUX, INTRPT, CARR, COUNT)
;
;    (4) A FOUR BIT BINARY FIELD THAT SPECIFIES THE INSTRUCTION TO
;          THE AM29811 NEXT ADDRESS CONTROL UNIT.
;        THE CONSTANTS DEFINED IN THE AM29811 INSTRUCTION SET CAN BE USED
;          HERE.
;
;   OVERLAY RESTRICTIONS:
;      MUST NOT BE OVERLAID WITH MACSTEN IF A CONDITION TEST IS SPECIFIED.

MAC:       DEF     1VB#0,3TI,B#1,1VI,4TI,4VB#1110,16I

;-------------------------------------------------------------------------

;  MACF
;  ****

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

MACF:      DEF     1VB#0,3TI,B#0,1VI,4VI,4VB#1110,16I

;-------------------------------------------------------------------------

;  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 CLOCKED 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 CLOCKED 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 EXTERNAL 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 OVERLAID WITH ANY OTHER FORMAT.

ALU:       DEF     10I,1VB#1,1VB#1,1I,1VI,12I,3VI,3VI,3VI,3VI,2SI

;-------------------------------------------------------------------------

;  REG
;  ***

; 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 B.

;       (2) A TWO BIT FIELD THE CONTENTS OF WHICH ARE ADDED TO THE BIT PATTERN
```

```
;          BEING USED TO SELECT THE 2901 INTERNAL REGISTER A.
;
;     (3) A TWO BIT FIELD USED TO CONTROL THE SELECTION OF 2901 INTERNAL
;         REGISTER B.   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 CONTOL PATTERNS'.
;
;     (5) A FOUR BIT BINARY PATTERN USED TO SELECT THE SCRATCH (OR EXTERNAL)
;         REGISTER.  THE CONSTANTS DEFINED IN THE SECTION CALLED REGISTER
;         DEFINITIONS (R6,R1 ETC.) CAN BE USED HERE.
;
; OVERLAY RESTRICTIONS:
;        MUST NOT BE OVERLAID WITH
;          MICINT, MACINT.
;
REG:        DEF     16I,2VI,2VI,2VI,2VI,4VI,38I
;-------------------------------------------------------------------------
;
; 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:       DEF     6I,4V,56I
;-------------------------------------------------------------------------
;  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 SISTEM 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     38I,B#1,25I
;
;    CKALU:     CAUSES THE 2901 CHIPS TO BE CLOCKED ON THE CURRENT CYCLE.
;               REGISTERS ARE LOADED ACCORDING TO THE VARIABLE FIELD
;               SUBSTITUTIONS IN THE ALU FORMAT.
;
CKALU:      DEF     5I,B#1,58I
;
;     CKREG:    CAUSES THE EXTERNAL (OR SCRATCH REGISTER FILE TO TO BE
;               CLOCKED.
;
CKREG:      DEF     4I,B#1,59I
;
;               CAUSES THE MICRO INTERUPT CONTROL UNIT TO BE CLOCKED (OR
;               IN OTHER WORDS, THE INSTRUCTION TO THE 2914 IS ENABLED).
;
CKMICINT:   DEF     3I,B#1,60I
;
;    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 PATERN SET UP BY THE RFRSH,READ,WRITE
;               AND RMWRT FORMATS IS CARRIED OUT.
;               SOME OF THESE CYCLES ONLY GO THROUGH PART OF THEEIR
;               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.
;
CKPORT:     DEF     1I,B#1,62I
;
;  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     2I,B#1,61I
;
;-------------------------------------------------------------------------
;
; 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 STATE 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
;       BADDRESS, STATSHFT, DATINSRT.
;
AUICK:      DEF    60I,4V
;-------------------------------------------------------------------
;
;  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 18I,4V,42I
;
;-------------------------------------------------------------------
;
; MACINT
; ******
;
;THIS FORMAT IS THE SAME AS MICINT EXCEPT THAT IT APPLIES TTO THE 2914
;CONTROLLING THE MACHINE INTERRUPT STRUCTURE.
;
MACINT:      DEF      14I,4V,46I
;
;-------------------------------------------------------------------
;
;  MACSTEN
;  *******
;
; 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 Z
;              1    NOT ENABLE C
;              2    NOT ENABLE N
;              3    NOT ENABLE P   (THIS IS USED AS OVR IN THE SPEC. FOR
;                                   AM 2904 BUT IS USED AS P IN THIS MACHINE)
;         (SEE 2904 DATA SHEET)
;
;  OVERLAY RESTRICTIONS:
;     MUST NOT BE OVERLAID WITH MAC OR MACF UNLESS THE CONDITION SELECT
;     VARIABLE FIELD IS LEFT TO DEFAULT IN MAC OR MACF.
;
MACSTEN:      DEF      48I,4VB#0000,28I
;
;
END
```

Appendix D

MICROCODE SOURCE TEXT

```
WORD 64
;
;MICROPROGRAM FIELD FORMAT
;-------------------------
;
;BIT FUNCTION                                  CONDITION FOR ACTION
;NO.
;-------------------------------------------------------------------
;M0  ALTERNATIVES:
;M1  (1) STATUS AND SHIFT INSTRUCTION            CK MAC STAT OR CK MICRO
;                                               STAT. ACTIVE (LOW) OR SHIFT
;                                               FUNCTION SELECTED BY ALU.
;M2  (2) 12 BIT BRANCH ADDRESS                   BRANCH ADDRESS REQUIRED BY
;                                               AM 29811
;M3  (3) BITS 0 TO 11 OF 16 BIT DATA FIELD       DATA FIELD SOURCED ONTO
;                                               D BUS
;M4  (4) AUX CLOCK CONTROL FIELD                 AUX CLK ENABLE ACTIVE.
;M5 M0 - CK DISPLAY
;M6 M1 - CK STATE DISPLAY
;M7 M2 - CK STATUS
;M8 M3 - CK MACH. INT.
;M9
;M10
;M11
;-------------------------------------------------------------------
;M12  (1) CARRY SELECT FIELD                     ALWAYS AFFECTS EXT.CARR LATCH
;M13                                             MACH.CARR. OR MIC.CARR.
;M14                                             IF CORRESPONDING ENABLE BITS
;     (2) BITS 12 THROUGH 14 OF DATA FIELD       EFFECTIVE IF DATA FIELD IS
;                                               ENABLED
;                                               ARE ACTIVE.
;-------------------------------------------------------------------
;M15  (1) I/O PORT CONTROL BIT 1               EFFECTIVE IF CK PORT HIGH
;       (2) BIT 15 OF DATA FIELD                 EFFECTIVE IF DATA FIELD IS
;                                               SOURCED ONTO D BUS
;-------------------------------------------------------------------
;M16    NEXT ADDRESS CONTROL                     ALWAYS APPLICABLE
;M17
;M18
;M19
;-------------------------------------------------------------------
;M20    (1) CONDITION TEST SELECT                TEST COND. REQUIRED BY AM29811
;M21    (2) NOT ENABLE I - M20
;M22        NOT ENABLE C - M21                   MACHINE STATUS ENABLED
;M23        NOT ENABLE N - M22
;          NOT ENABLE P - M23
```



```
;-------------------------------------------------------------------
;M24    CONDITION POLARITY TEST                  APPLICABLE IF TEST CONDITION
;                                                REQUIRED BY AM 29811
;-------------------------------------------------------------------
;M25    CK INSTRUCTION REG                       ALWAYS APPLICABLE
;-------------------------------------------------------------------
;M26    ALU SOURCE CONTROL                       ALWAYS APPLICABLE
;M27
;M28
;-------------------------------------------------------------------
;M29    ALU MS BYTE FUNCTION                     ALWAYS APPLICABLE
;M30
;M31
;-------------------------------------------------------------------
;M32    ALU LS BYTE FUNCTION
;M33                                          ALWAYS APPLICABLE
;M34
;-------------------------------------------------------------------
;M35    ALU DESTINATION CONTROL                  ALWAYS APPLICABLE
;M36
;M37
;-------------------------------------------------------------------
;M38    EXTERNAL REGISTER SELECT                 APPLICABLE IF EX.REG IS SOURCED
;M39                                              ONTO D BUS OR 2901 D INPUT
;M40                                                OR IF REG 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 B SELECT CONTROL      APPLICABLE IF REG B IS USED
;M45    (2) MICRO INT. CONTROL BITS 3 AND 4           APPLICABLE IF CK MIC.INT.
;                                                       IS HIGH
;-------------------------------------------------------------------
;M46   (1)INTERNAL REGISTER A SELECT MOD.         APPLICABLE IF REG A IS USED
;M47   (2) MAC.INT.CONTROL BITS 1 AND 2             APPLICABLE IF CK MACH.INT.
;                                                       IS HIGH
;-------------------------------------------------------------------
;M48  (1) INTERNAL REGISTER B SELECT MOD.          APPLICABLE IF REG.B IS USED
;M49  (2) MACH.INT.CONTROL BITS 3 AND 4            APPLICABLE IF CK MACH.INT.
;                                                       IS HIGH
;-------------------------------------------------------------------
;M50    2901 D INPUT SOURCE SELECT             APPLICABLE IF D INPUT IS SELECTED
```

```
AMDOS/29 AMDASM MICRO ASSEMBLER, V1.2
MIL STD 1750 PHASE 1 BIC FORMAT DEFINITION

;                                                    AS R OR S SOURCE FOR ALU
;-------------------------------------------------------------------------------
;M51     PORT CONTROL BIT 2                        APPLICABLE IF CK PORT IS HIGH
;-------------------------------------------------------------------------------
;M52     MACH.STAT. NOT ENABLE                      ALLWAYS APPLICABLE
;-------------------------------------------------------------------------------
;M53 MICRO STATUS NOT ENABLE                        ALWAYS APPLICABLE
;M54   D BUS SOURCE CONTROL FIELD                   ALWAYS APPLICABLE
;M55
;M56
;M57
;-------------------------------------------------------------------------------
;M58  CK ALU                                ALWAYS APPLICABLE
;M59  CK REG                                        ALWAYS APPLICABLE
;M60  CK MIC. INTERRUPT                             ALWAYS APPLICABLE
;M61  ENABLE AUX CLOCK FIELD                        ALWAYS APPLICABLE
;M62  CK PORT                                       ALWAYS APPLICABLE
;-------------------------------------------------------------------------------
;M63  AM29803 CONTROL                               ALWAYS APPLICABLE
;-------------------------------------------------------------------------------
;
;
TITLE   T H E   A M 2 9 0 0   F A M I L Y   M N E M O N I C S
;       ************************************************
;
;
;13 DECEMPBER 1976 JRM
;UPDATED SEPT 28, 1977
;
;AM29#  INSTRUCTION SET
;
;REGISTER DEFINITIONS
```



```
AMDOS/29 AMDASM MICRO ASSEMBLER, V1.2
T H E   A M 2 9 0 0   F A M I L Y   M N E M O N I C S

R0:     EQU     H#0
R1:     EQU     H#1
R2:     EQU     H#2
R3:     EQU     H#3
R4:     EQU     H#4
R5:     EQU     H#5
R6:     EQU     H#6
R7:     EQU     H#7
R8:     EQU     H#8
R9:     EQU     H#9
R10:    EQU     H#A
R11:    EQU     H#B
R12:    EQU     H#C
R13:    EQU     H#D
R14:    EQU     H#E
R15:    EQU     H#F
;
;AM2901 SOURCE OPERANDS (R S)
;
AQ:     EQU     Q#0
AB:     EQU     Q#1
ZQ:     EQU     Q#2
ZB:     EQU     Q#3
ZA:     EQU     Q#4
DA:     EQU     Q#5
DQ:     EQU     Q#6
DZ:     EQU     Q#7
;
;AM2901 ALU FUNCTIONS (R FUNCTION S)
;
ADD:    EQU     Q#0
SUBR:   EQU     Q#1
SUBS:   EQU     Q#2
OR:     EQU     Q#3
AND:    EQU     Q#4
NOTRS:  EQU     Q#5
EXOR:   EQU     Q#6
EXNOR:  EQU     Q#7
;
;AM2901 DESTINATION CONTROL
;
QREG:   EQU     Q#0
NOP:    EQU     Q#1
RAMA:   EQU     Q#2
RAMF:   EQU     Q#3
RAMQD:  EQU     Q#4
```

```
RAMD:   EQU     Q#5
RAMQD:  EQU     Q#6
RAMU:   EQU     Q#7
;
;AM29811 INSTRUCTION SET
;
JZ:     EQU     H#0     ;JUMP TO ADDRESS ZERO
CJS:    EQU     H#1     ;CONDITIONAL JUMP TO SUBROUTINE WITH JUMP
                        ;ADDRESS IN THE PIPELINE REGISTER
JMAP:   EQU     H#2     ;JUMP TO ADDRESS AT MAPPING PROM OUTPUT
CJP:    EQU     H#3     ;CONDITIONAL JUMP TO ADDRESS IN PIPELINE
                        ;REGISTER
PUSH:   EQU     H#4     ;PUSH STACK AND CONDITIONALLY LOAD COUNTER
JSRP:   EQU     H#5     ;JUMP TO SUBROUTINE WITH STARTING ADDRESS
                        ;CONDITIONALLY SELECTED FROM THE AM2911
                        ;R-REGISTER OR PIPELINE ADDRESS
CJV:    EQU     H#6     ;CONDITIONAL JUMP TO VECTOR ADDRESS
JRP:    EQU     H#7     ;JUMP TO ADDRESS CONDITIONALLY SELECTED FROM
                        ;AM2911 R-REGISTER OR PIPELINE REGISTER
RFCT:   EQU     H#8     ;REPEAT LOOP IF COUNTER IS NOT EQUAL TO ZERO
RPCT:   EQU     H#9     ;REPEAT PIPELINE ADDRESS IF COUNTER IS NOT
                        ;EQUAL TO ZERO
CRTN:   EQU     H#A     ;CONDITIONAL RETURN FROM SUBROUTINE
CJPP:   EQU     H#B     ;CONDITIONAL JUMP TO PIPELINE ADDRESS AND POP
                        ;STACK
LDCT:   EQU     H#C     ;LOAD COUNTER AND CONTINUE
LOOP:   EQU     H#D     ;TEST END OF LOOP
CONT:   EQU     H#E     ;CONTINUE TO NEXT ADDRESS
JP:     EQU     H#F     ;JUMP TO PIPELINE REGISTER ADDRESS
;
;
;AM2914 INSTRUCTION SET
;
MCLR:   EQU     H#0     ;MASTER CLEAR
CLRIN:  EQU     H#1     ;CLEAR ALL INTERRUPTS
CLRMB:  EQU     H#2     ;CLEAR INTERRUPTS FROM M-BUS
CLRMR:  EQU     H#3     ;CLEAR INTERRUPTS FROM MASK REGISTER
CLRVC:  EQU     H#4     ;CLEAR INTERRUPT FROM LAST VECTOR READ
RDVC:   EQU     H#5     ;READ VECTOR
RDSTA:  EQU     H#6     ;READ STATUS REGISTER
RDM:    EQU     H#7     ;READ MASK REGISTER
SETM:   EQU     H#8     ;SET MASK REGISTER
LDSTA:  EQU     H#9     ;LOAD STATUS REGISTER
BCLRM:  EQU     H#A     ;BIT CLEAR MASK REGISTER
BSETM:  EQU     H#B     ;BIT SET MASK REGISTER
CLRM:   EQU     H#C     ;CLEAR MASK REGISTER
```



```
DISIN:  EQU     H#D     ;DISABLE INTERRUPT REQUEST
LDM:    EQU     H#E     ;LOAD MASK REGISTER
ENIN:   EQU     H#F     ;ENABLE INTERRUPT REQUEST
;
;
;
;INTERNAL REGISTER SELECT CONTROL PATTERNS
;----------------------------------------
;
GP1:    EQU     B#00    ;SELECTS BITS 3 THROUGH 11 OF INSTRUCTION WORD AS
                        ;REGISTER SELECT
GP2:    EQU     B#01    ;SELECTS BITS 12 THROUGH 15
BASE:   EQU     B#10    ;SELECTS BASE REGISTER USING BITS 5 AND 6
RZERO:  EQU     B#11    ;SELECTS REGISTER ZERO
;
;
;CARRY SELECT CONTROL PATTERNS
;----------------------------
;
CONE:   EQU     B#000   ;SELECTS LOGICAL ONE AS CARRY TO BE STORED
CZERO:  EQU     B#001   ;SELECTS LOGICAL ZERO AS CARRY TO BE STORED
CQLSB:  EQU     B#010   ;SELECTS LSB OUTPUT OF Q REGISTER AS CARRY
CRMSB:  EQU     B#011   ;SELECTS MSB OUTPUT OF RAM AS CARRY
CRLSP:  EQU     B#100   ;SELECTS LSB OUTPUT OF RAM AS CARRY
CCAR:   EQU     B#101   ;SELECTS CARRY OUTPUT FROM ALU
CRMID:  EQU     B#110   ;SELECTS RAM MIDPOINT
CQMID:  EQU     B#111   ;SELECTS Q MIDPOINT
;
; NB. THERE ARE THREE CARRY LATCHES IN THE MK ONE IMPLEMENTATION.
;     ONE IS EXTERNAL TO THE 2904 AND IS CLOCKED ON EVERY MICROCYCLE.
;     THE OTHERS ARE THE CARRY BITS IN THE MICRO AND MACHINE STATUS REGISTERS
;     WHICH ARE WITHIN THE 2904 CHIP AND ARE ONLY CLOCKED 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.
;
;
SNONE:  EQU     B#0000  ;DISABLES ALL CPU OUTPUTS TO D BUS
```

```
AMDOS/29 AMDASM MICRO ASSEMPLER, V1.2
THE AM2900 FAMILY MNEMONICS

 SALU:    EQU    D#0001     ;SOURCES ALU ONTO D BUS
 SMEM:    EQU    D#0010      ;SOURCES MEMORY DATA REGISTER ONTO DBUS
 SREG:    EQU    D#0011      ;SOURCES REGISTER FILE ONTO D BUS
 SFAULT:  EQU    D#0100     ;SOURCES FAULT REGISTER
 SSTATUS:  EQU    D#0101    ;SOURCES MACHINE STATUS REGISTER
 SMANDT:  EQU    D#0110     ;SOURCES MANUAL DATA INPUT
 SMANRG:  EQU    D#0111     ;SOURCES MANUAL REGISTER SELECT FIELD
 SDATA:   EQU    D#1000     ;SOURCES MICROPROGRAM DATA FIELD
 SSTATE:  EQU    D#1001     ;SOURCES CPU STATE DISPLAY
 STIMA:   EQU    D#1010     ;SOURCES TIMER A
 STIMB:   EQU    D#1011     ;SOURCES TIMER B
 ;
 ;
 ;TEST CONDITION SELECT PATTERNS
 ;-----------------------------
 ;
 ; THESE CONTROL THE SOURCE OF THE TEST INPUT TO THE 29811 NEXT ADDRESS
 ; CONTROL UNIT.
 ;
 ;
 TZERO:   EQU    D#0000     ;PUTS LOGICAL ZERO ON TEST INPUT
 ZMSB:    EQU    D#0001     ;TESTS FOR ZERO MS BYTE
 ZLSB:    EQU    D#0010     ;TESTS FOR ZERO LS BYTE
 OVR:     EQU    D#0011     ;TESTS FOR OVERFLOW SET
 MUX:     EQU    D#0100     ;TESTS FOR OUTPUT OF AM2904 MULTIPLEXER
 INTRPT:  EQU    D#0101     ;TESTS MICROINTERRUPT REQUEST LINE
 CARR:    EQU    D#0110     ;TESTS OUTPUT OF EXTERNAL CARRY LATCH
 COUNT:   EQU    D#1111     ;TESTS COUTER ZERO
 ;
 ;
 ;TEST POLARITY SELECT PATTERNS
 ;----------------------------
 ;
 TRUE:    EQU    D#1
 FALSE:   EQU    D#0
 ;
 ;
 ;D INPUT TO 2901 SELECT CONTROL PATTERNS
 ;--------------------------------------
 ;
 ABUS:    EQU    D#0  ;SELECTS THE ADDRESS PUS (OUPPUT OF EXTERNAL REGISTER FILE)
 DBUS:    EQU    D#1  ;SELECTS THE INTERNAL TRISTATE DATA BUS
 ;
 ;
 ;FORMAT DEFINITIONS
```



```
AMDOS/29 AMDASM MICRO ASSEMPLER, V1.2
THE AM2900 FAMILY MNEMONICS

 ;------------------
 ;-------------------------------------------------------------------------------
 ;
 ;  BADDRESS
 ;  ********
 ;
 ; 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, 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
 ;      STATSHFT, DATINSRT, ENAUICK, AUICK
 ;
 BADDRESS:    DEF   52X,12VX:
 ;
 ;-------------------------------------------------------------------------------
 ;
 ;  STATSHFT
 ;  ********
 ;
 ;THIS FORMAT IS USED TO ENABLE THE BINARY PATTERN CONTROLLING THE AM2904
 ;CHIP TO BE INSERTED INTO THE MICROPROGRAM WORD.
 ;
 ; VARIABLE FIELD SUBSTITUTIONS:
 ;   (1) A 12 BIT BINARY NUMBER FORMING THE CONTROL
 ;       WORD FOR THE 2904 CHIP.
 ;             BIT 11 (MSB)  =   I12
 ;                 10        =   I11
 ;                  9        =   I9
 ;                  8        =    I8
 ;                  ....     =   ....
 ;                  0        =    I0
 ;   SEE AM2904 DATA SHEET FOR DEFINITION OF I0 TO I12.
 ;
 ;OVERLAY RESTRICTIONS:
 ;     CANNOT BE OVERLAID WITH:
 ;        BADDRESS, DATINSRT,ENAUICK,AUICK.
 ;
 ;
 STATSHFT:    DFF   52X,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 2904 ON EVERY CLOCK CYCLE.
;
;VARIABLE FIELD SUBSTITUTIONS:
;    (1) A THREE BIT BINARY PATTERN APPROPRIATE TO SELECT THE REQUIRED
;        INPUT TO THE CARRY SELECT MULTIPLEXER (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
;       DATINSRT
;
CARRYSEL:    DFF   49I,3V,12I
;
DATINSRT:    DEF   49I,16VI:D#
;-------------------------------------------------------------------------
;
; RFRSH
; *****
;
;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 (CKPORT) IS HELD HIGH.
;
; THERE ARE NO VARIABLE FIELDS.
;
;OVERLAY RESTRICTIONS:
;   MA  NOT BE OVERLAID WITH DATINSRT, READ, WRITE, RMWRT.
;
RFRSH:       DEF   12I,B#0,35I,B#0,15I
;
;-------------------------------------------------------------------------
;
; READ
; ****
;
;THIS FORMAT IS THE SAME AS RFRSH EXCEPT THAT A READ CYCLE IS INITIATED.
```



```
;
;OVERLAY RESTRICTIONS:
;      AY NOT BE OVERLAID WITH
;       DATINSRT, RFRSH, WRITE, RMWRT.
;
READ:        DEF     12I,B#1,35I,B#0,15I
;-------------------------------------------------------------------------
;
; WRITE
; *****
;
;THIS FORMAT IS THE SAME AS RFRSH EXCEPT THAT A WRITE CYCLE IS PRODUCED.
;
;OVERLAY RESTRICTIONS:
;   MAY NOT BE OVERLAID WITH
;        RFRSH, READ,RMWRT.
;
WRITE:       DEF     12I,B#0,35I,B#1,15I
;-------------------------------------------------------------------------
;
; RMWRT
; *****
;
; THIS FORMAT IS THE SAME AS RFRSH EXCEPT THAT A READ/MODIFY/WRITE
; CYCLE IS PRODUCED.
;
; OVERLAY RESTRICTIONS:
;    MUST NOT BE OVERLAID WITH
;         DATINSRT, RFRSH, READ, WRITE.
;
RMWRT:       DEF     12I,B#1,35I,B#1,15I
;
;-------------------------------------------------------------------------
;
;  MAC
;  ***
;
; 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
;          29803 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
```

```
;            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 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.
;            (TZERO, ZMSB, ZLSB, OVR, MUX, INTRPT, CARP, COUNT)
;
;    (4) A FOUR BIT BINARY FIELD THAT SPECIFIES THE INSTRUCTION TO
;            THE AM29811 NEXT ADDRESS CONTROL UNIT.
;        THE CONSTANTS DEFINED IN THE AM29811 INSTRUCTION SET CAN BE USED
;            HERE.
;
;  OVERLAY RESTRICTIONS:
;    MUST NOT BE OVERLAID WITH MACSTEN IF A CONDITION TEST IS SPECIFIED.
;
MAC:        DEF     1VB#0,3?I,B#1,1VI,4VI,4VB#1110,16I
;-------------------------------------------------------------------------
;
; MACF
; ****
;
; THIS IS THE SAME AS MAC EXCEPT THAT BIT 25 IS SET TO 0 RATHER THAN 1.
; THIS CAUSES THE INSTRUCTION REGISTER TO BE CLOCKED EITHER ON THE CURRENT
; CYCLE IN THE PIPELINED ARCHITECTURE OR ON THE NEXT CYCLE IN THE NON
; PIPELINED HARDWARE.
;
MACF:       DEF     1VB#0,3?I,B#0,1VI,4VI,4VB#1110,16I
;-------------------------------------------------------------------------
;
; ALU
; ***
;
; THIS FORMAT ENABLES THE BIT PATERNS 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 CLOCKED 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 CLOCKED 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 EXTERNAL 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 OVERLAID WITH ANY OTHER FORMAT.
;
;
ALU:        DEF     10I,1VB#1,1VB#1,1I,1VI,12I,3VI,3VI,3VI,3VI,2?I
;-------------------------------------------------------------------------
;
; REG
; ***
;
; 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 B.
;
;       (2) A TWO BIT FIELD THE CONTENTS OF WHICH ARE ADDED TO THE BIT PATTERN
;           BEING USED TO SELECT THE 2901 INTERNAL REGISTER A.
;
```

```
;       (3) A TWO BIT FIELD USED TO CONTROL THE SELECTION OF 2901 INTERNAL
;           REGISTER B.  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 CONTOL 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,2FX,2EX,2DX,2CX,4BX,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:       DEF     6X,4F,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
```



```
; 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,B#1,25X
;
;    CKALU:     CAUSES THE 2901 CHIPS TO BE CLOCKED ON THE CURRENT CYCLE.
;               REGISTERS ARE LOADED ACCORDING TO THE VARIABLE FIELD
;               SUBSTITUTIONS IN THE ALU FORMAT.
;
CKALU:      DEF     5X,B#1,58X
;
;     CKREG:    CAUSES THE EXTERNAL (OR SCRATCH REGISTER FILE TO TO BE
;               CLOCKED.
;
CKREG:      DEF     4X,B#1,59X
;
;               CAUSES THE MICRO INTERUPT CONTROL UNIT TO BE CLOCKED (OR
;               IN OTHER WORDS, THE INSTRUCTION TO THE 2914 IS ENABLED).
;
CKMICINT:   DEF     3X,B#1,60X
;
;    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 REFRSH,READ,WRITE
;               AND RMWRT 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.
;
CKPORT:     DEF     1X,B#1,62X
;
;  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,B#1,61X
;
```

```
;-------------------------------------------------------------------------------
;
; AUXCK
; *****
;
; 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 STATE 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
;      BADDRESS, STATSHFT, DATINSRT.
;
AUXCK:      DEF    60I,4V
;-------------------------------------------------------------------------------
;
; 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 18I,4V,42I
;
;-------------------------------------------------------------------------------
;
; MACINT
; ******
```



```
;
;THIS FORMAT IS THE SAME AS MICINT EXCEPT THAT IT APPLIES TTO THE 2914
;CONTROLLING THE MACHINE INTERRUPT STRUCTURE.
;
MACINT:       DEF      14I,4V,46I
;
;-------------------------------------------------------------------------------
;
; MACSTEN
; *******
;
; 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 Z
;              1    NOT ENABLE C
;              2    NOT ENABLE N
;              3    NOT ENABLE P   (THIS IS USED AS OVR IN THE SPEC. FOR
;                                   AM 2904 BUT IS USED AS P IN THIS MACHINE)
;       (SEE 2904 DATA SHEET)
;
;  OVERLAY RESTRICTIONS:
;     MUST NOT BE OVERLAID WITH NAC OR NACP UNLESS THE CONDITION SELECT
;     VARIABLE FIELD IS LEFT TO DEFAULT IN NAC OR NACP.
;
MACSTEN:      DEF      40I,4VP#0000,2WI
;
;
END

TOTAL PHASE 1 ERRORS =   0
```

```
      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 BIG MK1A
      ;
      ; FIRMWARE VERSION 1A.0
      ;
      ; DATE OF RELEASE -- JUNE 1, 1980   S.J.SHRIMPTON   RAE(F)/FS5
      ;
```



```
      P: EQU H#FFFF
      ;-------------------------------------------------------------------------
      ;
      ; SECOND K MICROCODE ADDRESSES
      ; ****************************
 FFFF ;
 050E PLGR1GR2: EQU H#50E
 05E9 REGSELBIT: EQU H#5E9
 05E1 SELBIT: EQU H#5E1
 04FA SHFTND16: EQU H#4FA
 0504 SHFTND32: EQU H#504
 04A6 SHFTNM: EQU H#4A6
 054B XOPCD400: EQU H#54B
 057E XOPCD402: EQU H#57E
 0407 XOPCD404: EQU H#407
 0418 XOPCD405: EQU H#418
 045A XOPCD40C: EQU H#45A
 0493 XOPCD40F: EQU H#493
 048A XOPCD40F: EQU H#48A
 040F XOPCD4A1: EQU H#40F
 0420 XOPCD4A2: EQU H#420
 0429 XOPCD4A4: EQU H#429
 043C XOPCD4A8: EQU H#43C
 0497 XOPCD4A7: EQU H#497
 048E XOPCD4A6: EQU H#48E
 049C XOPCD4A9: EQU H#49C
 0462 XOPCD4AA: EQU H#462
      XOPCD4AB: EQU H#4A3
      ;
      ;-------------------------------------------------------------------------
      ;
      ;
      ; THIS FIRMWARE CONTROLS ALL THE BASIC MACHINE FUNCTIONS EXCLUDING THE
      ; ACTUAL CARRYING OUT OF INSTRUCTIONS.
      ;
      ;
      ;  THE FUNCTIONS LOAD EXTERNAL REGISTER, DISPLAY EXTERNAL REGISTER AND
      ;  OBEY TEST INSTRUCTION ARE NOT INCLUDED.
      ;
 04A3 ;
      ORG H#0
      ;
 0000 ;RESET SEQUENCE START
 0000 RESET:    NAC ,,,JP
```

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

      / &         BADDRESS RESET
      / &         ALU
      ;
      ;INTERUPT A SEQUENCE START
 0001 SINTRPTA:    NAC ,,,JP
      / &         BADDRESS INTRPTA
      / &         ALU
      ;
      ;INTERUPT B SEQUENCE START
 0002 SINTRPTB:    NAC ,,,JP
      / &         BADDRESS INTRPTB
      / &         ALU
      ;
      ;SPARE SEQUENCE START
 0003 SFAULTSV: NAC ,,,JP
      / &         BADDRESS FAULTSV
      / &         ALU
      ;
      ;SPARE SEQUENCE START
 0004 SSPARE4:     NAC ,,,JP
      / &         BADDRESS HALTEST
      / &         ALU
      ;
      ;FRONT PANEL SERVICE SEQUENCE START
 0005 SPANEL:      NAC ,,,JP
      / &         BADDRESS PANEL
      / &         ALU
      ;
      ;SPARE SEQUENCE START
 0006 SSPARE5:     NAC ,,,JP
      / &         BADDRESS HALTEST
      / &         ALU
      ;
      ;REFRESH SEQUENCE START (DUMMY)
 0007 SREFRESH:    NAC ,,,JP
      / &         BADDRESS HALTEST
      / &         ALU
      ;
      ;
      ;
      ;-------------------------------------------------------------------------
      ;
      ;FRONT PANEL SERVICE SEQUENCE STARTS
      ;
      ;-----------------------------------
      ORG H#10
```



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

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

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

 0019 SSPAREP4:  MAC ...JP
      / &        ALU
      / &        BADDRESS HALTEST
      ;
      ;SPARE FUNCTION START
      ;--------------------
 001A SSPAREP5:  MAC ...JP
      / &        ALU
      / &        BADDRESS HALTEST
      ;
      ;SPARE FUNCTION START
      ;--------------------
 001B SSPAREP6:  MAC ...JP
      / &        ALU
      / &        BADDRESS HALTEST
      ;
      ;
      ; LOAD REGISTER
      ; -------------
 001C SLOADREG:  MAC ...JP & BADDRESS LOADREG & ALU
      ;
      ; DISPLAY PROGRAM COUNTER
      ; ----------------------
 001D SDISPPC:   MAC ...JP & BADDRESS DISPPC & ALU
      ;
      ; DISPLAY MEMORY
      ; --------------
 001E SDISPMEM:  MAC ...JP & BADDRESS DISPMEM & ALU
      ;
      ; DISPLAY REGISTER
      ; ----------------
 001F SDISPREG:  MAC ...JP & BADDRESS DISPREG & ALU
      ;
      ;-------------------------------------------------------------------------
      ;
      ;FRONT PANEL SERVICE FUNCTIONS
      ;-----------------------------
      ;
      ORG H'20
      ;
      ;-------------------------------------------------------------------------
      ;
      ;RUN FUNCTION
      ;------------
      ;
```



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

      ;(1) THE STATUS WORD IS LOADED INTO THE QREGISTER FROM EXTERNAL REGISTER
 0020 ;NUMBER 4
 0020 RUN:       MAC ...CONT
      / &        ALU ..ABUS,QREG,OR,OR,DZ
      / &        RFG ....R4
      / &         CKALU
      ;
      ;(2) BIT 15 OF THE STATUS WORD IS SET TO ZERO TO INDICATE MACHINE HALT
 0021            MAC ...CONT
      / &        ALU ..DBUS,QREG,OR,OR,DQ
      / &        DBUS SDATA
      / &        DATINSRT D#32768 ;MASK 1000000000000000
      / &        CKALU
      ;
      ;(3) THE NEW STATUS WORD IS LOADED INTO THE MACHINE STATE DISPLAY LATCH
      ;AND EXTERNAL REGISTER NUMBER 4.
 0022            MAC ...CONT
      / &        ALU ...QREG,OR,OR,ZQ
      / &        REG ....R4
      / &        DBUS SALU
      / &        ENAUICK
      / &        AUICK B#0010
      / &        CKREG
      ;
      ;(4)RETURN TO HALTEST SEQUENCE.
 0023            MAC ...JP
      / &        BADDRESS HALTEST
      / &        ALU
      ;
      ;-------------------------------------------------------------------------
      ;
      ;
      ;HALT FUNCTION
      ;-------------
      ;
      ;(1) THE STATUS WORD IS LOADED INTO THE Q REGISTER FROM
      ;EXTERNAL REGISTER #4
 0024 HALT:      MAC ...CONT
      / &        ALU ..ABUS,QREG,OR,OR,DZ
      / &        RFG ....R4
      / &        CKALU
      ;
      ;(2) BIT 15 OF THE STATUS WORD IS SET TO ZERO TO INDICATE MACHINE HALT.
 0025            MAC ...CONT
      / &        ALU ..DBUS,QREG,AND,AND,DQ
```

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

      / &         DBUS SDATA
      / &         DATINSRT D#32767 ; MASK @1111111111111111
      / &         CKALU
      ;
      ;(3) THE NEW MACHINE STATUS WORD IS LOADED INTO THE MACHINE STATE DISPLAY LATCH
      ;AND EXTERNAL REGISTER #4
0026              MAC ,,,CONT
      / &         ALU ,,,QREG,OR,OR,ZQ
      / &         REG ,,,R4
      / &         DBUS SALU
      / &         EMAUICK
      / &         AUICK B#0010
      / &         CKREG
      ;
      ;(4) RETURN TO HALTEST SEQUENCE
0027              MAC ,,,JP
      / &         BADDRESS HALTEST
      / &         ALU
      ;
      ;-------------------------------------------------------------------------------
      ;LOAD REGISTER
      ;-------------
      ;
      ;(1)REGISTER SELECT TAKES PLACE
0028 LOADREG:     MAC ,FALSE,TZERO,CJS
      / &         ALU
      / &         BADDRESS MANREGST
      ;
      ;(2) THE CONTENTS OF THE SWITCH FIELD ARE LOADED INTO THE SELECTED REGISTER.
0029              MAC ,,,JP
      / &         ALU ,,DBUS,RAMF,OR,OR,DZ
      / &         REG ,,GR2
      / &         DBUS SMANDT
      / &         BADDRESS HALTEST
      / &         CKALU
      ;
      ;-------------------------------------------------------------------------------
      ;
      ;DISPLAY REGISTER
      ;----------------
      ;
      ;(1)REGISTER SELECT TAKES PLACE
002A DISPREG:     MAC ,FALSE,TZERO,CJS
      / &         ALU
      / &         BADDRESS MANREGST
      ;
```



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

      ;(2)
      ;THE CONTENTS OF THE SELECTED REGISTER ARE PLACED IN THE DISPLAY LATCH.
002B              MAC ,,,CONT
      / &         ALU ,,,NOP,OR,OR,ZA
      / &         REG ,,,GR2
      / &         DBUS SALU
      / &         EMAUICK
      / &         AUICK B#0001
      ;
      ;(3) JUMP TO HALTEST TAKES PLACE
002C              MAC ,,,JP
      / &         BADDRESS HALTEST
      / &         ALU
      ;
      ;-------------------------------------------------------------------------------
      ;REGISTER SELECT ROUTINE
      ;-----------------------
      ;
      ;(1)THE CONTENTS OF INTERNAL REGISTER ZERO ARE TEMPORARILY STORED IN THE
      ;Q REGISTER.
002D MANREGST:    MAC ,,,CONT
      / &         ALU ,,,QREG,OR,OR,ZA
      / &         REG ,,,RZERO
      / &         CKALU
      ;
      ;(2)THE REGISTER SELECT FIELD IS LOADED INTO IR0 AND CIRCULARLY UPSHIFTED.
002E              MAC ,,,CONT
      / &         ALU ,,DBUS,RAMU,OR,OR,DZ
      / &         STATSHFT B#0010000000
      / &         RFG ,,RZERO
      / &         DBUS SMANRG
      / &         CKALU
      ;
      ;(3)THE CONTENTS OF IR0 ARE UPSHIFTED
002F              MAC ,,,CONT
      / &         ALU ,,,RAMU,OR,OR,ZB
      / &         REG ,,,RZERO
      / &         STATSHFT B#0010000000
      / &         DBUS SALU
      / &         CKALU
      ;
      ;(4)THE CONTENTS OF IR0 ARE UPSHIFTED
0030              MAC ,,,CONT
      / &         ALU ,,,RAMU,OR,OR,ZB
      / &         RFG ,,RZERO
      / &         STATSHFT B#0010000000
```

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

      / &         DBUS SALU
      / &         CKALU
      ;
      ;(5)THE CONTENTS OF IR0 ARE UPSHIFTED.
 0031             MAC ,,,CONT
      / &         ALU ,,RAMU,OR,OR,ZB
      / &         REG ,,RZERO
      / &         STATSHFT B#0010100000000
      / &         CKALU
      ;
      ;(6) THE CONTENTS OF IR0 ARE LOADED INTO THE INSTRUCTION REGISTER AND THE
      ;ORIGINAL CONTENTS OF IR0 ARE RESTORED.
 0032             MACF ,FALSE,TZERO,CRTN
      / &         ALU ,,,RAMA,OR,OR,ZQ
      / &         REG ,,RZERO,RZERO
      / &         DBUS SALU
      / &         CKALU
      ;
      ;-------------------------------------------------------------------------
      ;LOAD PROGRAM COUNTER
      ;--------------------
      ;
      ;(1)THE CONTENTS OF THE FRONT PANEL DATA SWITCHFIELD ARE LOADED INTO
      ;ER0 WHICH IS THE PROGRAM COUNTER.
 0033 LOADPC:     MAC ,,,JP
      / &         ALU
      / &         ~BUS SMANDT
      / &         REG ,,,,R0
      / &         CKREG
      / &         BADDRESS HALTEST
      ;
      ;-------------------------------------------------------------------------
      ;DISPLAY PROGRAM COUNTER
      ;-----------------------
      ;
      ;(1)THE CONTENTS OF ER0 ARE DISPLAYED ON THE DATA DISPLAY FIELD.
 0034 DISPPC:     MAC ,,,CONT
      / &         ALU
      / &         REG ,,,,R0
      / &         DBUS SREG
      / &         ENAUICK
      / &         AUICK B#0001
      ;
      ;(2) JUMP TAKES PLACE TO HALTEST.
 0035             MAC ,,,JP
      / &         BADDRESS HALTEST
```



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

      / &         ALU
      ;
      ;-------------------------------------------------------------------------
      ;DISPLAY MEMORY
      ;--------------
      ;
      ;THE CONTENTS OF THE MEMORY LOCATION CURRENTLY POINTED TO BY THE PC ARE
      ;DISPLAYED ON THE FRONT PANEL LED DISPLAY.
      ; THE PROGRAM COUNTER IS INCREMENTED.
      ;
      ;(1)THE MEMORY IS ACCESSED
 0036 DISPMEM:    MAC ,,,CONT
      / &         ALU ,,ABUS,QREG,ADD,ADD,DZ
      / &         READ
      / &         STATSHFT B#0100000000000
      / &         CKPORT & CKALU
      / &         REG ,,,,R0
      ;
      ;(2)THE DATA FROM THE MEMORY IS LOADED INTO THE LED DISPLAY LATCHES.
 0037             MAC ,,,CONT
      / &         ALU
      / &         DBUS SMEM
      / &         ENAUICK
      / &         AUICK B#0001
      ;
      ;(3) JUMP TO HALTEST TAKES PLACE.
      ;    THE NEW PROGRAM COUNTER CONTENTS ARE LOADED INTO ER0.
 0038             MAC ,,,JP
      / &         ALU ,,,NOP,OR,OR,ZQ
      / &         DBUS SALU
      / &         REG ,,,,R0 & CKREG
      / &         BADDRESS HALTEST
      ;
      ;-------------------------------------------------------------------------
      ;LOAD MEMORY
      ;-----------
      ;
      ;THE CONTENTS OF THE FRONT PANEL DATA SWITCH FIELD ARE LOADED INTO THE
      ;MEMORY LOCATION CURRENTLY POINTED TO BY THE PROGRAM COUNTER.
      ; THE PROGRAM COUNTER IS INCREMENTED AND THE DATA JUST WRITTEN TO MEMORY IS
      ; DISPLAYED ON THE LED DATA DISPLAY FIELD.
      ;JUMP TAKES PLACE TO HALTEST.
 0039 LOADMEM:    MAC ,,,CONT
      / &         ALU ,,ABUS,QREG,ADD,ADD,DZ
      / &         REG ,,,,R0
      / &         WRITE
```

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

      / &         DBUS SMANDT
      / &         CKPORT & CKALU
      / &         ENAUICK & STATSHFT B#010000000001
      ;
      ;(2) THE INCREMENTED PC CONTENTS ARE PLACED IN ER0 AND JUMP TAKES PLACE
      ; TO HALTEST.
 003A             NAC ,,,JP & ALU ,,,NOP,OR,OR,ZQ & CKREG & REG ....R0
      / &         DBUS SALU & BADDRESS HALTEST
      ;-------------------------------------------------------------------------
      ;
      ; DISPLAY STATUS
      ; ##############
      ;
 003B DISPSTAT:   NAC ,,,CONT & ALU & DBUS SSTATUS & ENAUICK & STATSHFT B#000000100001
      ;
 003C             NAC ,,,JP & ALU  & BADDRESS HALTEST
      ;
      ;-------------------------------------------------------------------------
      ;
      ;
      ;SINGLE STEP
      ;-----------
      ;
      ;A SINGLE MACHINE INSTRUCTION IS EXECUTED AND THE PROGRAM COUNTER IS
      ;INCREMENTED.
      ;
      ;(1) THE INSTRUCTION FROM MEMORY IS FETCHED
 003D SINGSTP:    NAC ,,,CONT
      / &         ALU
      / &         REG ....R0
      / &         CKPORT
      / &         READ
      ;
      ;(2)THE INSTRUCTION IS LOADED INTO THE INSTRUCTION REGISTER AND ER1.
      ;   JUMP SUB TAKES PLACE TO  EXECUTE .
 003E             NACP ,FALSE,TZERO,CJS
      / &         BADDRESS EXECUTE
      / &         ALU
      / &         DBUS SMEM
      / &         REG ....R1
      / &         CKREG
      ;
      ;
      ;(4)JUMP TO HALTEST TAKES PLACE.
 003F             NAC ,,,JP
      / &         BADDRESS HALTEST
```



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

      / &         ALU
      ;
      ;
      ;-------------------------------------------------------------------------
      ; FRONT PANEL SERVICE SEQUENCE
      ; ----------------------------
      ;
      ;(1)
      ;THE MICROINTERRUPT THAT CAUSED ENTRY INTO THIS AREA IS CLEARED.
 0040 PANEL:      NAC ,,,CONT
      / &         ALU
      / &         MICINT CLRVC
      / &         CKMICINT
      ;
      ;(2) THE STATUS REGISTER IN THE MICROINTERUPT UNIT IS LOADED WITH ZERO.
 0041             NAC ,,,CONT
      / &         ALU
      / &         DBUS SDATA
      / &         DATINSRT D#0
      / &         MICINT LDSTA
      / &         CKMICINT
      ;
      ;(3) A VECTOR JUMP IS MADE TO THE APPROPRIATE FRONT PANEL SERVICE SEQUENCE.
 0042             NAC ,FALSE,TZERO,CJV & ALU
      ;
      ;-------------------------------------------------------------------------
      ;
      ;   HALTEST SEQUENCE
      ;   ----------------
      ;
      ;
      ;(3) A CHECK IS STARTED TO SEE IF HALT HAS BEEN SET.
      ;THE MACHINE STATE WORD IS PASSED THROUGH THE ALU AND SHIFTED UP SO THAT THE
      ;MOST SIGNIFICANT BIT GOES INTO THE CARRY. THE ALU IS NOT CLOCKED SO THAT NO
      ;REGISTERS ARE LOADED.
 0043 HALTEST:    NAC ,,,CONT
      / &         ALU ,,ABUS,RAMU,OR,OR,DZ
      / &         STATSHFT B#000010000000
      / &         CARRYSEL CRMSB
      / &         DBUS SALU
      / &         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.
 0044             NAC ,TRUE,CARR,CJP
      / &         BADDRESS FETCH
```

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

      / &           ALU
      ;
      ;
      ;(5) THE MICROINTERRUPT IS TESTED TO SEE IF AN INTERRUPT HAS OCCURED.
0045 HALTED:       NAC ,FALSE,INTRPT,CJV
      / &           ALU
      / &           MICINT RDVC & CKMICINT
      ;
      ;(6) AN UNCONDITIONAL JUMP IS MADE TO HALTED.
0046               NAC ,,,JP
      / &           BADDRESS HALTED
      / &            ALU
      ;
      ;
      ;-------------------------------------------------------------------------
      ;FETCH SEQUENCE
      ;--------------
      ;
      ;(1) THE PROGRAM COUNTER VALUE IN Q IS PLACED IN THE PC (ER0). JUMP TAKES
      ;PLACE TO THE EXECUTE ROUTINE.
0047 FETCH1:       NAC ,FALSE,ZERO,CJS
      / &           ALU
      / &           BADDRESS EXECUTE
      ;
      ;(2)THIS IS THE ENTRY POINT TO THE FETCH SEQUENCE.
      ;A CONDITIONAL JUMP OCCURS TO THE MICROINTERRUPT SERVICE ROUTINE IF THE
      ;INTERRUPT REQUEST LINE IS LOW.
0048 FETCH:         NAC ,FALSE,INTRPT,CJV
      / &           MICINT RDVC & CKMICINT
      / &           ALU & REG ,,,,R0
      / &           CKPORT & READ
      ;
      ;
      ;(3) THE INSTRUCTION FROM MEMORY IS PLACED IN THE INSTRUCTION REGISTER
      ;AND EXTERNAL REGISTER #1.
0049               NACF ,,,JP
      / &           BADDRESS FETCH1
      / &           ALU
      / &           DBUS SMEM
      / &           REG ,,,,R1
      / &           CKREG
      ;
      ;
      ;-------------------------------------------------------------------------
      ;RESET SEQUENCE
      ;--------------
```



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

      ;
      ; IN THE NEXT TWO INSTRUCTIONS THE STATSHFT FORMAT IS USED TO SET UP
      ; AUXILIARY CLOCK BITS BEYOND THE RANGE OF AUICK.  THIS IS BECAUSE THE
      ; AUXILIARY CLOCK FIELD WAS EXTENDED AND CHANGING AUICK WOULD HAVE MEANT
      ; MAKING CHANGES TO THE WHOLE OF THE MICROCODE.
      ;
      ;
      ;(1) MASTER CLEAR OF BOTH MACHINE INTERRUPT CHIPS IS EXECUTED.
004A RESET:        NAC ,,,CONT & ALU &  MICINT MCLR
      / &           ENAUICK & STATSHFT B#0000000011000
      ;
      ;(2) THE MASK REGISTER OF BOTH MACHINE INTERRUPT CHIPS IS LOADED SO THAT ALL
      ; INTERRUPTS ARE ENABLED.
004B               NAC ,,,CONT & ALU &  MICINT CLRM
      / &           ENAUICK & STATSHFT B#0000000011000
      ;
      ;
      ;(3) MASTER CLEAR OF THE MICRO-INTERRUPT UNIT IS EXECUTED.
004C               NAC ,,,CONT & ALU & MICINT MCLR & CKMICINT
      ;
      ;(4) THE MICRO-INTERRUPT MASK REGISTER IS LOADED SO THAT ALL INTERRUPTS
      ; ARE ACCEPTED.
004D               NAC ,,,CONT & ALU & MICINT LDM & DBUS SDATA & DATINSRT D#0
      / &           CKMICINT
      ;
      ;(5) THE INSTRUCTION COUNTER IS CLEARED
004E               NAC ,,,CONT & ALU ,,,NOP,AND,AND,DZ & DBUS SALU & CKREG & REG ,,,,R0
      ;
      ;(6) THE FAULT FLAG REGISTER, THE LEAST SIG. 12 BITS OF MAC. STATUS, THE TIMERS,
      ;  THE FAULT REGISTER AND THE PENDING INTERRUPT REGISTER ARE CLEARED.
004F               NAC ,,,CONT & ALU ,,,NOP,AND,AND,DZ & ENAUICK
      / &           STATSHFT B#0000111111111## & DBUS SALU & MACINT CLRM
      / &           REG ,,,,R3 & CKREG
      ;
      ;(7) THE MOST SIGNIFICANT FOUR BITS OF THE MACINT STATUS REGISTER ARE CLEARED.
0050               NAC ,,,CONT & ALU ,B##,,NOP,AND,AND,DZ & MACSTEN B#0000
      / &           STATSHFT B#0000000000000 & DBUS SALU
      ;
      ;(9) THE TEMPORARY MASK STORAGE REGISTER (ER15) IS LOADED WITH ZEROS.
0051               NAC ,,,CONT & ALU ,,,NOP,AND,AND,DZ & REG ,,,,R15 & DBUS SALU & CKREG
      ;
      ;(10) THE QREG IS LOADED WITH A MASK WITH BITS SET TO ONE EVERYWHERE EXCEPT
      ;     IN THE MOST SIG 3 BITS.
0052               NAC ,,,CONT & ALU ,,DBUS,QREG,OR,OR,DZ & CKALU & DBUS SDATA
      / &           DATINSRT D#16383
      ;
```

```
AMDOS/29 AMDASM 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 BUT
      ; THE POWER DOWN INTERRUPT AND THE MACHINE ERROR INTERRUPT ARE DISABLED.
0053          NAC ,,,CONT & ALU ,,,NOP,OR,OR,ZQ & DBUS SALU & MACINT LDM
      / &     STATSHFT B#0000000011000 & ENAUICK
      ;(12) THE Q REGISTER IS LOADED WITH 1000110000000000
0054          NAC ,,,CONT & ALU ,,DBUS,QREG,OR,OR,DZ & CKALU & DBUS SDATA
      / &     DATINSRT D#35840
      ;
      ;(13) THE STATE WORD IS LOADED SO THAT:
      ;         THE 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
0055          NAC ,,,CONT & ALU ,,,NOP,OR,OR,ZQ & DBUS SALU & REG ,,,,R4 & CKREG
      / &     ENAUICK & STATSHFT B#0000000000010
      ;
      ;(14) JUMP TAKES PLACE TO A DEFINED LOCATION IN THE SECOND K OF ROM SO THAT
      ; ADDITIONAL RESET PROCEDURES CAN BE ADDED.
0056          NAC ,,,JP & ALU &  BADDRESS HALTEST
      ;
      ;-------------------------------------------------------------------------
      ;-------------------------------------------------------------------------
      ; EXECUTE SEQUENCE
      ;-----------------
      ;
      ;(1) THE PROGRAM COUNTER IS INCREMENTED. JUMP MAP TAKES PLACE TO THE START
      ;  OF THE APPROPRIATE EMULATION SEQUENCE.
0057 EXECUTE:   NAC ,,,JMAP
      / &       ALU ,,ABUS,NOP,ADD,ADD,DZ & DBUS SALU & REG ,,,,R0 & CKREG
      / &       STATSHFT B#0100000000000
      ;
      ;-------------------------------------------------------------------------
      ;
      ; 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 ER3 AND CAUSES
      ; A MACHINE ERROR INTERRUPT -- NUMBER 1.
```



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

      ;
0058 FAULTSV:   NAC ,,,CONT & ALU ,,DBUS,QREG,OR,OR,DZ & CKALU & DBUS SFAULT
      ;
      ;(2) THE FAULT FLAG REGISTER IS CLEARED.
      ;    THE CONTENTS OF THE FAULT REGISTER ARE ORED WITH THE CONTENTS OF THE
      ;     QREGISTER AND THE RESULTS PLACED IN THE FAULT REGISTER ER3.
0059          NAC ,,,CONT & ALU ,,ABUS,NOP,OR,OR,DQ & REG ,,,,R3 & CKREG
      / &     DBUS SALU & ENAUICK & STATSHFT B#0000001000000
      ;
      ;(3) THE MICRO INTERRUPT IS CLEARED.
005A          NAC ,,,CONT & ALU &  CKMICINT & MICINT CLRVC
      ;
      ;(4) THE BINARY PATTERN 0100000000000000 IS LOADED INTO THE QREG.
005B FAULTINT:  NAC ,,,CONT & ALU ,,DBUS,QREG,OR,OR,DZ & DBUS SDATA & CKALU
      / &     DATINSRT D#16384
      ;
      ;(5) THE CONTENTS OF THE QREGISTER ARE ORED WITH AND LOADED INTO THE
      ;PENDING INTERRUPT REGISTER.
005C          NAC ,,,CONT & ALU ,,,NOP,OR,OR,ZQ & DBUS SALU & ENAUICK
      / &     STATSHFT B#0010000000000
      ;
      ;(6) JUMP TO HALTEST TAKES PLACE.
005D          NAC ,,,JP & ALU &  BADDRESS HALTEST
      ;
      ;
      ;-------------------------------------------------------------------------
      ;
      ;
      ; INTERRUPT '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 2914 IS READ AND STORED IN ER8.
005E INTRPTA:   NAC ,,,CONT & ALU &  ENAUICK & CKREG & REG ,,,,R8
      / &     STATSHFT B#0000000001000 & MACINT RDVC
      ;
      ;(1A) THE CONTENTS OF ER0 ARE SWOPPED WITH THE CONTENTS OF IR0.
005F          NAC ,,,CONT & ALU ,,ABUS,RAMA,OR,OR,DZ & REG ,,RZERO,RZERO,R8
      / &     DBUS SALU & CKREG & CKALU
      ;
      ;(1B) THE CONTENTS OF IR0 ARE MASKED BY (0000000000000111).
0060          NAC ,,,CONT & ALU ,,DBUS,RAMF,AND,AND,DA & REG ,,RZERO,RZERO
```

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

      / &         DBUS SDATA & DATINSRT D#7 & CKALU
      ;
      ;(1C) THE CONTENTS OF IR0 ARE SHIFTED ONE PLACE UPWARDS.
0061              MAC ,,,CONT & ALU ,,,RAMU,OR,OR,ZA & REG ,,RZERO,RZERO & CKALU
      / &          STATSHFT B#0000010000000
      ;
      ;(2) THE CONTENTS OF ER8 AND IR0 ARE SWOPPED BACK.
0062              MAC ,,,CONT & ALU ,,ABUS,RAMA,OR,OR,DZ & REG ,,RZERO,RZERO,R8
      / &         DBUS SALU & CKALU & CKREG
      ;
      ;(2A) THE MACHINE INTERRUPT IS CLEARED.
0063              MAC ,,,CONT & ALU &  ENAUXCK & MACINT CLRVC
      / &         STATSHFT B#0000000001000
      ;
      ;(3) THE MACHINE INTERRUPT STATUS IS ZEROED.
0064              MAC ,,,CONT & ALU ,,ABUS,NOP,AND,AND,DZ & DBUS SALU & ENAUXCK
      / &         MACINT LDSTA & STATSHFT B#0000000001000
      ;
      ;(4) THE MICRO-INTERRUPT IS CLEARED.  JUMP TO THE SECOND PART OF THE
      ;  INTERRUPT SEQUENCE TAKES PLACE.
0065              MAC ,,,JP & ALU &  CKMICINT & MICINT CLRVC
      / &         BADDRESS INTRPT2
      ;
      ;-------------------------------------------------------------------------
      ;
      ;  THIS IS THE FIRST PART OF THE MACHINE INTERRUPT SERVICE SEQUENCE FOR AN
      ; INTERRUPT DETECTED BY THE 'B' GROUP OF INTERRUPTS. IE. THE MOST SIGNIFICANT
      ;  GROUP.
      ;
      ;(1) THE VECTOR IS READ FROM THE 'B' 2914.
0066 INTRPTB:     MAC ,,,CONT & ALU &  ENAUXCK & CKREG
      / &         STATSHFT B#0000000010000 & REG ,,,R8 & MACINT RDVC
      ;
      ;(2) THE QREG IS LOADED WITH D#8.
0067              MAC ,,,CONT & ALU ,,DBUS,QREG,OR,OR,DZ & DBUS SDATA & CKALU
      / &         DATINSRT D#8
      ;
      ;(3) THE CONTENTS OF ER8 ARE SWOPPED WITH THE CONTENTS OF IR0.
0068              MAC ,,,CONT & ALU ,,ABUS,RAMA,OR,OR,DZ & REG ,,RZERO,RZERO,R8
      / &         DBUS SALU & CKREG & CKALU
      ;
      ;
      ;(3A) THE CONTENTS OF IR0 ARE MASKED BY (0000000000000111).
0069              MAC ,,,CONT & ALU ,,DBUS,RAMF,AND,AND,DA & REG ,,RZERO,RZERO
      / &         DBUS SDATA & DATINSRT D#7 & CKALU
      ;(4) THE CONTENTS OF IR0 ARE ORED WITH D#8 AND SHIFTED UP ONE PLACE.
```



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

006A              MAC ,,,CONT & ALU ,,RAMU,OR,OR,AQ & REG ,,RZERO,RZERO & CKALU
      / &         STATSHFT B#0000010000000
      ;
      ;(5) THE CONTENTS OF IR0 AND ER8 ARE SWOPPED BACK.
006B              MAC ,,,CONT & ALU ,,ABUS,RAMA,OR,OR,ZA & REG ,,RZERO,RZERO,R8
      / &         CKALU & CKREG & DBUS SALU
      ;
      ;(5A) THE MACHINE INTERRUPT IS CLEARED.
006C              MAC ,,,CONT & ALU &  MACINT CLRVC & ENAUXCK
      / &         STATSHFT B#0000000010000
      ;
      ;(6) THE MACHINE INTERRUPT STATUS IS ZEROED.
006D              MAC ,,,CONT & ALU ,,ABUS,NOP,AND,AND,DZ & DBUS SALU & MACINT LDSTA
      / &         STATSHFT B#0000000010000 & ENAUXCK
      ;
      ;(7) THE MICRO-INTERRUPT IS CLEARED. JUMP TO THE SECOND PART OF THE INTERRUPT
      ; SERVICE SEQUENCE TAKES PLACE.
006E              MAC ,,,JP & ALU &  MICINT CLRVC
      / &         CKMICINT
      / &         BADDRESS INTRPT2
      ;
      ;-------------------------------------------------------------------------
      ;
      ;(1) THE MICRO-INTERRUPT STATUS IS LOADED WITH ZERO.
006F INTRPT2:     MAC ,,,CONT & ALU & DBUS SDATA & DATINSRT D#0 & CKMICINT
      / &         MICINT LDSTA
      ;
      ;(2) THE ADDRESS OF WHERE TO STORE THE MACHINE STATUS IS FETCHED FROM MEMORY.
0070              MAC ,,,CONT & ALU ,,ABUS,NOP,ADD,ADD,DZ & REG ,,,R8 & CKREG
      / &         CKPORT & READ & STATSHFT B#0100000000000 & DBUS SALU
      ;
      ;(2A) THE ADDRESS IS PLACED IN ER2.
0071              MAC ,,,CONT & ALU & DBUS SMEM & REG ,,,,R2 & CKREG
      ;
      ;(2A) THE MACHINE INTERRUPT MASK FROM ER15 IS INVERTED AND LOADED INTO THE QREG.
0072              MAC ,,,CONT & ALU ,,ABUS,QREG,EINOR,EINOR,DZ & CKALU & REG ,,,,R15
      ;
      ;(3) THE MACHINE INTERRUPT MASK WORD IS STORED.
0073              MAC ,,,CONT & ALU ,,,NOP,OR,OR,ZQ & REG ,,,,R2 & CKPORT & WRITE
      / &         DBUS SALU
      ;
      ;(4) THE CONTENTS OF ER2 ARE INCREMENTED.
0074              MAC ,,,CONT & ALU ,,ABUS,NOP,ADD,ADD,DZ & DBUS SALU & REG ,,,,R2
      / &         CKREG & STATSHFT B#0100000000000
      ;
```

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

      ;(5) THE MACHINE STATUS WORD IS STORED.
0075          NAC ,,,CONT & ALU & REG ,,,,R2 & CKPORT & WRITE & DBUS SSTATUS
      / &     STATSHFT B#0000001000000
      ;
      ;(6) THE CONTENTS OF ER2 ARE INCREMENTED.
0076          NAC ,,,CONT & ALU ,,ABUS,NOP,ADD,ADD,DZ & DBUS SALU
      / &     REG ,,,,R2 & CKREG & STATSHFT B#0100000200000
      ;
      ;(7) THE PC IS PLACED IN QREG.
0077          NAC ,,,CONT & ALU ,,ABUS,QREG,OR,OR,DZ & REG ,,,,R0 & CKALU
      ;
      ;(8) THE CONTENTS OF THE Q REG (PC) ARE STORED.
0078          NAC ,,,CONT & ALU ,,,NOP,OR,OR,ZQ & CKPORT & WRITE & REG ,,,,R2
      / &     DBUS SALU
      ;
      ;(9) THE ADDRESS FROM WHERE TO GET THE NEW MACHINE STATUS IS FETCHED.
0079          NAC ,,,CONT & ALU & REG ,,,,R0 & CKPORT & READ
      ;
      ;(10) THE ADDRESS IS PLACED IN ER2.
007A          NAC ,,,CONT & ALU & REG ,,,,R2 & CKREG & DBUS SMEM
      ;
      ;(11) THE MACHINE INTERRUPT MASK IS FETCHED, THE CONTENTS OF ER2 ARE
      ;  INCREMENTED.
007B          NAC ,,,CONT & ALU ,,ABUS,NOP,ADD,ADD,DZ & DBUS SALU & CKREG & CKPORT
      / &     READ & REG ,,,,R2 & STATSHFT B#0100000000000
      ;
      ;(12) THE MACHINE INTERRUPT MASK IS LOADED, INVERTED, INTO THE Q REG.
007C          NAC ,,,CONT & ALU ,,DPUS,QREG,EXNOR,EXNOR,DZ
      / &     DBUS SMEM & CKALU
      ;
      ;(12A) THE MACHINE INTERRUPT MASK IS LOADED INTO ER15.
007D          NAC ,,,CONT & ALU ,,,NOP,OR,OR,ZQ & DBUS SALU
      / &     REG ,,,,R15 & CKREG
      ;
      ;(12B) JUMP SUB TAKES PLACE TO MIMICINT.
007E          NAC ,FALSE,TZERO,CJS & ALU  & BADDRESS MIMICINT
      ;
      ;(13) THE MACHINE STATUS WORD IS FETCHED, THE CONTENTS OF ER2 ARE INCREMENTED.
007F          NAC ,,,CONT & ALU ,,ABUS,NOP,ADD,ADD,DZ & DBUS SALU & CKREG
      / &     CKPORT & READ & REG ,,,,R2 & STATSHFT B#0100000000000
      ;
      ;(14) THE MACHINE STATUS WORD IS LOADED.
0080          NAC ,,,CONT & ALU & DBUS SMEM & ENAUICK & AUICK B#0100
      ;
      ;(14A) THE FOUR MOST SIGNIFICANT BITS OF THE MACHINE STATUS WORD ARE LOADED.
0081          NAC ,,,CONT & ALU ,B#0 & DBUS SMEM & MACSTEN B#0000
```



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

      / &     STATSHFT B#0000000000000
      ;(15) THE PROGRAM COUNTER CONTENTS ARE FETCHED.
0082          NAC ,,,CONT & ALU &  CKPORT & READ & REG ,,,,R2
      ;
      ;(16) THE PROGRAM COUNTER IS LOADED, JUMP TO HALTEST TAKES PLACE.
0083          NAC ,,,JP & ALU & DBUS SMEM & REG ,,,,R0 & CKREG & BADDRESS HALTEST
      ;-------------------------------------------------------------------------
      ;
      ;
      ;
      ORG B#A0
      ;
      ;
      ;REGISTER USAGE
      ;--------------
      ;
      ; INTERNAL (2901) REGISTERS ARE AS DEFINED IN MIL-STD-1750
      ;
      ; EXTERNAL (29705) REGISTERS:
      ;
      ; ER0 PROGRAM COUNTER
      ; ER1 HOLDS MACHINE INSTRUCTION IN ADDITION TO MAIN INSTRUCTION
      ;       REGISTER
      ; ER2 HOLDS SECOND WORD OF TWO WORD INSTRUCTION
      ; ER3 HOLDS FAULT REGISTER
      ; ER4 HOLDS MACHINE STATE INDICATOR WORD
      ; ER5 HOLDS DERIVED OPERAND WORD 1
      ; ER6 HOLDS DERIVED OPERAND WORD 2
      ; ER7 HOLDS DERIVED OPERAND WORD 3
      ; ER8 MICROPROG. SCRATCH SPACE
      ; ER9   DITTO
      ; ER10   DITTO
      ; ER11   DITTO
      ; ER12
      ; ER13
      ; ER14
      ; ER15 HOLDS INTERRUPT MASK TEMPORARILY WHEN INTERRUPTS ARE DISABLED
      ;*************************************************************************
      ;
      ;                       SUBROUTINES
      ;                       ***********
      ;
      ;*************************************************************************
      ;
```

```
                ;
                ; OPERAND FETCH ROUTINES
                ; ----------------------
                ;
                ; OFDI1:
                ;       FETCHES A SINGLE WORD DIRECT OR DIRECT INDEXED OPERAND AND PLACES
                ;       IT IN ER5.   DIRECT OR DIRECT INDEXED IS CHOSEN ACCORDING TO THE
                ;       CONTENTS OF BITS 12 THRU 15 OF THE MACHINE INSTRUCTION.
                ;       THE PROGRAM COUNTER IS AUTOMATICALLY INCREMENTED ONCE TO TAKE
                ;       ACCOUNT OF THE LONG INSTRUCTION FORMAT.
                ;       THIS SUBROUTINE CAN THEREFORE ONLY BE USED ONCE WITHIN A
                ;       MACHINE INSTRUCTION EMULATION SEQUENCE.
                ;       THE ADDRESS OF THE DERIVED OPERAND (AFTER INDEXING) IS LEFT IN ER2.
                ;
                ; OFDI2:
                ;       FETCHES A DOUBLE WORD DIRECT OR DIRECT INDEXED OPERAND AND PLACES
                ;       IT IN ER5 AND ER6.   DIRECT OR DIRECT INDEXED IS CHOSEN ACCORDING
                ;       TO THE CONTENTS OF BITS 12 THRU 15 OF THE MACHINE INSTRUCTION.
                ;       THE PROGRAM COUNTER IS AUTOMATICALLY INCREMENTED TO TAKE ACCOUNT OF
                ;       THE LONG INSTRUCTION FORMAT.   THIS SUBROUTINE CAN THEREFORE BE
00A0            ;       USED ONLY ONCE IN A MACHINE INSTRUCTION EMULATION SEQUENCE.
                ;
                ; OFDI3:
                ;       FETCHES A TRIPLE WORD DIRECT OR DIRECT INDEXED OPERAND AND PLACES
                ;       IT IN ER5 AND ER6 AND ER7.   DIRECT OR DIRECT INDEXED IS CHOSEN
                ;       ACCORDING TO THE CONTENTS OF BITS 12 THRU 15 OF THE MACHINE
                ;       INSTRUCTION.
                ;       THE PROGRAM COUNTER IS AUTOMATICALLY INCREMENTED TO TAKE ACCOUNT OF THE
                ;       LONG INSTRUCTION FORMAT.   THIS SUBROUTINE CAN THEREFORE BE USED
                ;       ONLY ONCE DURING A MACHINE INSTRUCTION EMULATION SEQUENCE.
                ;
                ; OFBI1:
                ;       FETCHES A SINGLE WORD BASE RELATIVE OPERAND AND PLACES IT IN ER5.
                ;       THE ADDRESS OF THE DERIVED OPERAND (IE CONTENTS OF SPECIFIED BASE
                ;       REGISTER PLUS THE DISPLACEMENT) IS LEFT IN ER2.
                ;
                ; OFB2:
                ;       FETCHES A DOUBLE WORD BASE RELATIVE OPERAND AND PLACES IT IN ER5
                ;       AND ER6.
                ;
                ; OFB3:  FETCHES A TRIPLE WORD BASE RELATIVE OPERAND AND PLACES IT IN ER5,
                ;       ER6 AND ER7.
                ;
                ; OFBI1: FETCHES A SINGLE WORD BASE RELATIVE INDEXED OPERAND AND PLACES IT
```



```
                ;       IN ER5. THE ADDRESS OF THE DERIVED OPERAND (IE THE CONTENTS OF THE BASE
                ;       PLUS THE CONTENTS OF THE INDEX REGISTER) IS LEFT IN ER2.
                ;
                ; OFBI2: FETCHES A DOUBLE WORD BASE RELATIVE INDEXED OPERAND AND PLACES IT
                ;       IN ER5 AND ER6.
                ;
                ; OFBI3:
                ;       FETCHES A TRIPLE WORD BASE RELATIVE INDEXED OPERAND AND PLACES IT IN
                ;       ER5, ER6 AND ER7.
                ;
                ; OFISP:
                ;       FETCHES AN IMMEDIATE 16 BIT POSITIVE OPERAND.
                ;
                ; OFIM:
                ;       FETCHES AN IMMEDIATE OPERAND.  THE PROGRAM COUNTER IS AUTOMATICALLY
                ;       INCREMENTED TO TAKE ACCOUNT OF THE LONG INSTRUCTION FORMAT.
                ;
                ;
                ;*****************************************************************************
                ;
                ;          MICROCODE FOR OPERAND FETCH ROUTINES
                ;
                ;*****************************************************************************
                ;
                ;(1) MEMORY ACCESS IN READ MODE TAKES PLACE.  THE PROGRAM COUNTER
                ; CONTENTS ARE INCREMENTED AND PLACED BACK IN ER0.
00A0 OFDI1:     NAC ,,,CONT & ALU ,,ABUS,NUP,ADD,AND,DZ & REG ,,,,R0 & CKREG & CKPORT
                / &         READ & STATSHFT B#0100000000000 & DBUS SALU
                ;
                ;(2) THE CONTENTS OF IR0 ARE PLACED IN ER2, ZERO IS PLACED IN IR0.
00A1                NAC ,,,CONT & ALU ,,,RAMA,AND,AND,ZA & REG ,,RZERO,RZERO,R2
                / &         DBUS SALU & CKALU & CKREG
                ;
                ;(3)THE CONTENTS OF GR2 ARE ADDED TO THE CONTENTS OF THE MEMORY DATA
                ; REGISTER AND THE RESULT PLACED IN IR0.
00A2                NAC ,,,CONT & ALU ,,DBUS,RAMF,ADD,ADD,DA & REG ,,RZERO,GR2 & CKALU
                / &         DBUS SMEM & STATSHFT B#0000000000000
                ;(4) THE CONTENTS OF ER2 ARE RESTORED TO IR0, CONTENTS OF IR0 ARE PLACED IN
                ;   ER2.
00A3                NAC ,,,CONT & ALU ,,ABUS,RAMA,OR,OR,DZ & REG ,,RZERO,RZERO,R2
                / &         DBUS SALU & CKALU & CKREG
                ;
                ;(5) THE DERIVED OPERAND IS FETCHED FROM MEMORY.
```

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

00A4              NAC ,,,CONT & ALU & READ & CKPORT & REG ,,,,R2
                  ;
                  ;(6) THE DERIVED OPERAND IN THE MEMORY DATA REGISTER IS PLACED IN ER5
00A5              NAC ,FALSE,TZERO,CRTN & ALU & DBUS SMEM & REG ,,,,R5 & CKREG
                  ;
                  ;-------------------------------------------------------------------------
                  ;
                  ; OPDX2 -- FETCHES TWO WORD DERIVED OPERAND AND PLACES THE RESULT IN ER 5 AND 6.
                  ;
                  ;(1) JUMP SUB TO OPDX1 -- THIS FETCHES THE FIRST WORD.
00A6 OPDX2:       NAC ,FALSE,TZERO,CJS & ALU &  BADDRESS OPDX1
                  ;
                  ;(2) THE CONTENTS OF ER2 ARE INCREMENTED.
00A7              NAC ,,,CONT & ALU ,,ABUS,NOP,ADD,ADD,DZ & REG ,,,,R2 & DBUS SALU
                  / &       CKREG & STATSHFT B#0100000000000
                  ;
                  ;(3) MEMORY ACCESS IN READ MODE TAKES PLACE.
00A8              NAC ,,,CONT & ALU &  CKPORT & READ & REG ,,,,R2
                  ;
                  ;(4) THE DATA FROM MEMORY IS PLACED IN ER6.  RETURN TAKES PLACE.
00A9              NAC ,FALSE,TZERO,CRTN & ALU & DBUS SMEM & REG ,,,,R6 & CKREG
                  ;
                  ;-------------------------------------------------------------------------
                  ;
                  ;  OPDX3 FETCHES A 3WORD DIRECT OPERAND FROM MEMORY AND PLACES IT IN
                  ; ER5,6 AND 7.
                  ;
                  ;(1) JUMP SUB TO OPDX1, THIS OBTAINS THE FIRST WORD OF THE DERIVED OPERAND
                  ; AND PLACES IT IN ER5.
00AA OPDX3:       NAC ,FALSE,TZERO,CJS & ALU &  BADDRESS OPDX1
                  ;
                  ;(2) THE CONTENTS OF ER2 ARE INCREMENTED.
00AB               NAC ,,,CONT & ALU ,,ABUS,NOP,ADD,ADD,DZ & CKREG & DBUS SALU
                  / &        REG ,,,,R2 & STATSHFT B#0100000000000
                  ;
                  ;(3) MEMORY ACCESS IN READ MODE TAKES PLACE, THE CONTENTS OF ER2 ARE
                  ; INCREMENTED.
00AC              NAC ,,,CONT & ALU ,,ABUS,NOP,ADD,ADD,DZ & DBUS SALU & REG ,,,,R2
                  / &       CKREG & CKPORT & READ & STATSHFT B#0100000000000
                  ;
                  ;(4) THE CONTENTS OF THE MEMORY DATA REGISTER ARE PLACED IN ER6.
00AD              NAC ,,,CONT & ALU & REG ,,,,R6 & DBUS SMEM & CKREG
                  ;
                  ;(5) THE MEMORY IS ACCESSED IN READ MODE.
00AE              NAC ,,,CONT & ALU & CKPORT & READ & REG ,,,,R2
```



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

                  ;
                  ;(6) THE MEMORY DATA IS PLACED IN ER7, RETURN TAKES PLACE.
00AF              NAC ,FALSE,TZERO,CRTN & ALU & REG ,,,,R7 & DBUS SMEM & CKREG
                  ;
                  ;-------------------------------------------------------------------------
                  ;
                  ;OPB1 -- FETCHES A SINGLE WORD BASE RELATIVE OPERAND.
                  ;
                  ;(1) THE INSTRUCTION IN ER1 IS MASKED TO LEAVE ONLY THE LEAST SIGNIFICANT
                  ; 8 BITS AND PLACED IN THE Q REG.
00B0 OPB1:        NAC ,,,CONT & ALU ,,ABUS,QREG,OR,AND,DZ & REG ,,,,R1 & CKALU
                  ;
                  ;(2) THE CONTENTS OF THE SPECIFIED BASE REGISTER ARE ADDED TO THE CONTENTS
                  ; OF THE QREGISTER AND THE RESULT PLACED IN ER2.
00B1              NAC ,,,CONT & ALU ,,,NOP,ADD,ADD,AQ & REG ,,BASE,R2 & DBUS SALU
                  / &       CKREG & STATSHFT B#0000000000000
                  ;
                  ;(3) THE DERIVED OPERAND IS FETCHED FROM MEMORY.
00B2              NAC ,,,CONT & ALU & READ & CKPORT &  REG ,,,,R2
                  ;
                  ;(4) THE DERIVED OPERAND IS PLACED IN ER5, RETURN TAKES PLACE.
00B3              NAC ,FALSE,TZERO,CRTN & ALU & DBUS SMEM & CKREG & REG ,,,,R5
                  ;
                  ;-------------------------------------------------------------------------
                  ;
                  ;
                  ; OPBX1 -- FETCHES BASE RELATIVE INDEXED OPERAND.
                  ;
                  ;(1)THE CONTENTS OF IR0 ARE PLACED IN ER5, ZERO IS PLACED IN IR0.
00B4 OPBX1:       NAC ,,,CONT & ALU ,,,RAM4,AND,AND,ZA & REG ,,RZERO,RZERO,R5 & CKALU
                  / &       CKREG & DBUS SALU
                  ;
                  ;(2) THE CONTENTS OF THE SPECIFIED BASE REGISTER ARE ADDED TO THE CONTENTS
                  ; OF THE SPECIFIED INDEX REGISTER AND THE RESULT PLACED IN ER2
00B5              NAC ,,,CONT & ALU ,,,NOP,ADD,ADD,AB & REG ,,BASE,GR2,R2 & CKREG
                  / &       DBUS SALU & STATSHFT B#0000000000000
                  ;
                  ;(3) THE MEMORY IS ACCESSED TO OBTAIN THE DERIVED OPERAND.
00B6              NAC ,,,CONT & ALU & REG ,,,,R2 & CKPORT & READ
                  ;
                  ;(4) THE DERIVED OPERAND IS PLACED IN ER5 AND THE CONTENTS OF ER5 RESTORED
                  ; TO IR0.
00B7              NAC ,FALSE,TZERO,CRTN & ALU ,,ABUS,RAM5,OR,OR,DZ & REG ,,RZERO,,R5
                  / &       DBUS SMEM & CKALU & CKREG
```

```
      ;
      ;-------------------------------------------------------------------------------
      ;
      ;
      ; OFIX1 -- FETCHES AN INDIRECT OR INDIRECT-INDEXED OPERAND.
      ;
      ;(1) JUMP SUB IS TAKEN PLACE TO OFDX1 TO GET THE ADDRESS OF THE INDIRECT
      ;   OPERAND.
@@B8 OFIX1:    NAC ,FALSE,TZERO,CJS & ALU &  BADDRESS OFDX1
      ;
      ;(2) THE INDIRECT OPERAND IS OBTAINED FROM MEMORY USING THE DERIVED OPERAND
      ;   IN ER5 AS THE ADDRESS.
      ;   THE ADDRESS IS ALSO PLACED IN QREG
@@B9          NAC ,,,CONT & ALU ,,ABUS,QREG,OR,OR,DZ & CKPORT & READ & CKALU
      / &      REG ,,,,R5
      ;
      ;(2A) THE CONTENTS OF THE QREG ARE PLACED IN ER2.
@@BA          NAC ,,,CONT & ALU ,,,NOP,OR,OR,ZQ & DBUS SALU & REG ,,,,R2
      / &      CKREG
      ;
      ;(3) THE INDIRECT OPERAND IN THE MEMORY DATA REGISTER IS PLACED IN ER5.
      ; RETURN TAKES PLACE.
@@BB          NAC ,FALSE,TZERO,CRTN & ALU & DBUS SMEM & CKREG & REG ,,,,R5
      ;
      ;-------------------------------------------------------------------------------
      ;
      ;
      ; OFIX3 -- FETCHES A THREE WORD INDIRECT OPERAND
      ;
      ;(1) JUMP SUB TO OFDX1 TO GET THE ADDRESS OF THE INDIRECT OPERAND.
@@BC OFIX3:    NAC ,FALSE,TZERO,CJS & ALU &  BADDRESS OFDX1
      ;
      ;(2) THE CONTENTS OF ER5 ARE PLACED IN QREG.
@@BD          NAC ,,,CONT & ALU ,,ABUS,QREG,OR,OR,DZ & REG ,,,,R5 & CKALU
      ;(3) THE CONTENTS OF THE QREG ARE PLACED IN ER2.
@@BE          NAC ,,,CONT & ALU ,,NOP,OR,OR,ZQ & REG ,,,,R2 & DBUS SALU & CKREG
      ;
      ;(4) THE FIRST WORD OF THE DERIVED OPERAND IS FETCHED FROM MEMORY.
      ;    THE CONTENTS OF ER2 ARE INCREMENTED.
@@BF          NAC ,,,CONT & ALU ,,ABUS,NOP,ADD,ADD,DZ & REG ,,,,R2 & CKPORT & READ
      / &      DBUS SALU & CKREG & STATSHFT B#2100000000000
      ;
      ;(5) THE CONTENTS OF THE MEMORY DATA REGISTER ARE PLACED IN ER5.
@@C0          NAC ,,,CONT & ALU & DBUS SMEM & REG ,,,,R5 & CKREG
      ;
      ;(6) THE SECOND WORD IS FETCHED FROM MEMORY.  THE CONTENTS OF ER2 ARE
```



```
      ; INCREMENTED.
@@C1          NAC ,,,CONT & ALU ,,ABUS,NOP,ADD,ADD,DZ & REG ,,,,R2 & DBUS SALU
      / &      CKPORT & READ & CKREG & STATSHFT B#2100000000000
      ;
      ;(7) THE CONTENTS OF THE MEMORY DATA REGISTER ARE PLACED IN ER6.
@@C2          NAC ,,,CONT & ALU & DBUS SMEM & REG ,,,,R6 & CKREG
      ;
      ;(8) THE THIRD WORD IS FETCHED FROM MEMORY.
@@C3          NAC ,,,CONT & ALU & REG ,,,,R2 & CKPORT & READ
      ;
      ;(9) THE CONTENTS OF THE MDR ARE PLACED IN ER7.  RETURN TAKES PLACE.
@@C4          NAC ,FALSE,TZERO,CRTN & ALU & DBUS SMEM & REG ,,,,R7 & CKREG
      ;
      ;-------------------------------------------------------------------------------
      ;
      ;
@@C5 OFISP:    NAC ,,,CONT
      / &      ALU ,,ABUS,QREG,OR,OR,DZ
      / &      REG ,,,,R1
      / &      CKALU
      / &      DBUS SREG
      ;
      ;
@@C6          NAC ,,,CONT
      / &      ALU ,,DBUS,QREG,AND,AND,DQ
      / &      DBUS SDATA
      / &      DATINSRT D#15
      / &      CKALU
      ;
      ;
@@C7          NAC ,,,CONT
      / &      ALU ,,DBUS,QREG,ADD,ADD,DQ
      / &      DBUS SDATA
      / &      DATINSRT D#1
      / &      CKALU
      ;
      ;
@@C8          NAC ,FALSE,TZERO,CRTN
      / &      ALU ,,,NOP,OR,OR,ZQ
      / &      DBUS SALU
      / &      REG ,,,,R5
      / &      CKREG
      ;
      ;
@@C9 OFISN:    NAC ,FALSE,TZERO,CJS
```

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

      / &       BADDRESS OPISP
      / &       ALU
      ;
      ;
 00CA           NAC ,FALSE,TZERO,CRTN
      / &       ALU ,,ABUS,NOP,SUBR,SUBR,DZ
      / &       REG ,,,,R5
      / &       DBUS SALU
      / &       CKREG
      / &       STATSHFT B#0180000000000
      ;
      ; OPIM -- FETCHES AN IMMEDIATE OPERAND AND PLACES IT IN ER5.
      ;
      ;(1) THE OPERAND IS FETCHED FROM MEMORY AND THE PC VALUE INCREMENTED.
 00CB OPIM:     NAC ,,,CONT & ALU ,,ABUS,NOP,ADD,ADD,DZ & DBUS SALU & REG ,,,,R0
      / &       STATSHFT B#0180000000000 & CKREG & CKPORT & READ
      ;
      ;(2) THE DATA FROM MEMORY IS PLACED IN ER5. RETURN TAKES PLACE.
 00CC           NAC ,FALSE,TZERO,CRTN & ALU & REG ,,,,R5 & CKREG & DBUS SMEM
      ;
      ;-----------------------------------------------------------------------
      ;
      ;
      ;OPGENPUR --- THIS SUBROUTINE FETCHES AN OPERAND FROM THE ADDRESS IN
      ; MEMORY POINTED TO BY ER2.  THE CONTENTS OF ER2 ARE NOT CHANGED AND NEITHER
      ; ARE THOSE OF THE PC (ER0).  THE OPERAND IS PLACED IN ER5.
      ;
      ;(1) THE OPERAND IS FETCHED FROM MEMORY.
 00CD OPGENPUR:  NAC ,,,CONT & ALU &  CKPORT & READ & REG ,,,,R2
      ;
      ;(2) THE DERIVED OPERAND IS PLACED IN ER5. RETURN TAKES PLACE.
 00CE           NAC ,FALSE,TZERO,CRTN & ALU & DBUS SMEM & CKREG & REG ,,,,R5
      ;
      ;-----------------------------------------------------------------------
      ;
      ;
      ;
      ;
      ;
      ;-----------------------------------------------------------------------
      ;
      ; OPERAND SEND ROUTINES
      ; *********************
      ;
      ; OSDI1 -- THIS ROUTINE SENDS A DIRECT OR DIRECT INDEXED OPERAND OF SINGLE
      ; TO ITS DESTINATION.
```



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

      ;
      ;(1) MEMORY ACCESS IN READ MODE TAKES PLACE. THE PROGRAM COUNTER VALUE IS
      ;   INCREMENTED AND PLACED BACK IN ER0
 00CF OSDI1:    NAC ,,,CONT & ALU ,,ABUS,NOP,ADD,ADD,DZ & REG ,,,,R0 & CKREG & CKPORT
      / &       READ & STATSHFT B#0122000000000 & DBUS SALU
      ;
      ;(2) THE CONTENTS OF IR0 ARE PLACED IN ER2, ZERO IS PLACED IN IR0.
 00D0           NAC ,,,CONT & ALU ,,,RAMA,AND,AND,ZA & REG ,,RZERO,RZERO,R2
      / &       DBUS SALU & CKALU & CKREG
      ;
      ;(3) THE CONTENTS OF GR2 ARE ADDED TO THE CONTENTS OF THE MEMORY DATA REGISTER
      ;   AND THE RESULT PLACED IN IR0.
 00D1           NAC ,,,CONT & ALU ,,DBUS,RAMF,ADD,ADD,DA & REG ,,RZERO,GR2 & CKALU
      / &       DBUS SMEM & STATSHFT B#0000000000000
      ;
      ;(4) THE CONTENTS OF ER2 ARE RESTORED TO IR0, THE CONTENTS OF IR0 ARE PLACED
      ; IN ER2.
 00D2           NAC ,,,CONT & ALU ,,ABUS,RAMA,OR,OR,DZ & REG ,,RZERO,RZERO,R2
      / &       DBUS SALU & CKALU & CKREG
      ;
      ;(4A) THE OPERAND IN ER5 IS PLACED IN THE QREG.
 00D3           NAC ,,,CONT & ALU ,,ABUS,QREG,OR,OR,DZ & REG ,,,,R5 & CKALU
      ;(5) THE OPERAND IS SENT TO MEMORY.
      ;  RETURN TAKES PLACE.
 00D4           NAC ,FALSE,TZERO,CRTN & ALU ,,,NOP,OR,OR,ZQ & DBUS SALU & CKPORT
      / &       WRITE & REG ,,,,R2
      ;
      ;
      ;-----------------------------------------------------------------------
      ;
      ; OSDI -- THIS ROUTINE SENDS A BASE RELATIVE OPERAND TO ITS DESTINATION.
      ;
      ;(1) THE INSTRUCTION IN ER1 IS MASKED TO LEAVE ONLY THE LEAST SIGNIFICANT 8
      ; BITS AND PLACED IN THE QREG.
 00D5 OSDI:     NAC ,,,CONT & ALU ,,ABUS,QREG,OR,AND,DZ & REG ,,,,R1 & CKALU
      ;(2) THE CONTENTS OF THE SPECIFIED BASE REGISTER ARE ADDED TO THE CONTENTS OF
      ; THE QREGISTER AND THE RESULT PLACED IN ER2.
 00D6           NAC ,,,CONT & ALU ,,,NOP,ADD,ADD,AQ & REG ,,BASE,R2 & DBUS SALU
      / &       CKREG & STATSHFT B#0000000000000
      ;
      ;(2A) THE OPERAND IS PLACED IN QREG FROM ER5.
 00D7           NAC ,,,CONT & ALU ,,ABUS,QREG,OR,OR,DZ & REG ,,,,R5 & CKALU
      ;(3) THE OPERAND IS SENT TO MEMORY, RETURN TAKES PLACE.
 00D8           NAC ,FALSE,TZERO,CRTN & ALU ,,,NOP,OR,OR,ZQ & DBUS SALU & REG ,,,,R2
      / &       CKPORT & WRITE
```

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

```
                ;-------------------------------------------------------------------------
                ;
                ;
                ; OSBX1 -- THIS ROUTINE SENDS A BASE RELATIVE INDEXED OPERAND TO ITS DESTINATION
                ; *****
                ;
                ;(1) THE CONTENTS OF IR0 ARE PLACED IN ER8, ZERO IS PLACED IN IR0.
00D9 OSBX1:     NAC ,,,CONT & ALU ,,,RAMA,AND,AND,ZA & REG ,,RZERO,RZERO,R8
                / &         CKALU & CKREG & DBUS SALU
                ;
                ;(2) THE CONTENTS OF THE SPECIFIED BASE REGISTER ARE ADDED TO THE CONTENTS
                ;  OF THE SPECIFIED INDEX REGISTER AND THE RESULT PLACED IN ER2
00DA            NAC ,,,CONT & ALU ,,,QREG,ADD,ADD,AB & REG ,,BASE,BR2,R2 & CKREG
                / &         DBUS SALU & STATSHFT B#000000000000
                ;
                ;(3) THE OPERAND IN ER5 IS PLACED IN QREG.
00DB            NAC ,,,CONT & ALU ,,ABUS,QREG,OR,OR,DZ & REG ,,,,R5 & CKALU
                ;
                ;(4) THE OPERAND, NOW IN THE QREG IS SENT TO MEMORY, ER2 BEING USED TO SUPPLY
                ; THE ADDRESS.
00DC            NAC ,,,CONT & ALU ,,,NOP,OR,OR,ZQ & CKALU
                / &         REG ,,RZERO,RZERO,R2 & DBUS SALU & CKPORT & WRITE
                ;
                ;(5) THE CONTENTS OF IR0 ARE RESTORED, RETURN TAKES PLACE.
00DD            NAC ,FALSE,TZERO,CRTN & ALU ,,ABUS,RAMF,OR,OR,DZ & REG ,,RZERO,,R8
                / &         CKALU
                ;
                ;-------------------------------------------------------------------------
                ;
                ; OSIX1 -- THIS SUBROUTINE SENDS AN INDIRECT OR INDIRECT INDEXED OPERAND TO
                ;   ITS DESTINATION.
                ;
                ;(1) JUMP SUB TO OFDX1, THIS FETCHES THE ADDRESS OF THE INDIRECT OPERAND TO
                ; ER5.  THE CONTENTS OF ER5 ARE PLACED IN QREG.
00DE OSIX1:     NAC ,FALSE,TZERO,CJS & ALU ,,ABUS,QREG,OR,OR,DZ & REG ,,,,R5
                / &         CKALU &  BADDRESS OFDX1
                ;(2) THE OPERAND, NOW IN THE QREG IS SENT TO MEMORY USING THE CONTENTS OF
                ; ER5 AS ADDRESS.   RETURN TAKES PLACE.
00DF            NAC ,FALSE,TZERO,CRTN & ALU ,,,NOP,OR,OR,ZQ & DBUS SALU
                / &         CKPORT & WRITE & REG ,,,,R5
                ;
                ;-------------------------------------------------------------------------
                ;
                ;
                ;-------------------------------------------------------------------------
```


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

```
                ;
                ; OSGENPUR -- 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 OFDX1,OFDX2 OR OFDX3.  NB THE EXECUTION OF OSGENPUR
                ; DOES NOT RESULT IN A CHANGE OF THE PC VALUE IN ER0.
                ;
                ;(1) THE CONTENTS OF ER5 ARE PLACED IN QREG.
00E0 OSGENPUR:  NAC ,,,CONT & ALU ,,ABUS,QREG,OR,OR,DZ & REG ,,,,R5 & CKALU
                ;(2) THE OPERAND, NOW IN THE QREG IS SENT TO THE ADDRESS IN MEMORY POINTED TO
                ; BY ER2.
                ;  RETURN TAKES PLACE.
00E1            NAC ,FALSE,TZERO,CRTN & ALU ,,,NOP,OR,OR,ZQ & REG ,,,,R2 & CKPORT
                / &         WRITE & DBUS SALU
                ;
                ;-------------------------------------------------------------------------
                ;
                ;
                ;-------------------------------------------------------------------------
                ;
                ; SIGN DETERMINE SUBROUTINE
                ; *************************
                ;
                ; THIS SUBROUTINE DETERMINES THE SIGN OF THE PRODUCT OR QUOTIENT IN A MULTIPLY
                ; OR DIVIDE OPERATION.
                ;  THE MULTIPLICAND (OR DIVIDEND) IS ASSUMED TO BE IN GR1 AND THE MULTIPLIER
                ;  (OR DIVISOR) IN ER5.  THE MS BIT OF ER8 IS SET IF THE PRODUCT (OR QUOTIENT)
                ;   IS NEGATIVE AND CLEARED IF IT IS POSITIVE.  THE REST OF ER8 IS CLEARED.
                ;   THE CONTENTS OF THE Q REGISTER ARE LOST DURING THIS ROUTINE.
                ;
00E2 ASSIGN:    NAC ,,,CONT
                / &         ALU ,,ABUS,QREG,OR,OR,DZ
                / &         REG ,,,,R5 & CKALU
                ;
                ;
                ; A SIGN BIT ONLY MASK IS PLACED IN ER8.
00E3            NAC ,,,CONT
                / &         ALU
                / &         DBUS SDATA & DATINSRT D#32768
                / &         REG ,,,,R8 & CKREG
                ;
                ;
                ;(2) THE CONTENTS OF THE QREG (MULTIPLIER) ARE MASKED BY THE CONTENTS OF
                ; ER8 SO THAT ONLY THE SIGN BIT OF THE MULTIPLIER IS LEFT.
00E4            NAC ,,,CONT
                / &         ALU ,,ABUS,NOP,AND,AND,DQ
```

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

      / &         DBUS SALU & REG ....R8
      ;
      ;(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             NAC ,TRUE,ZMSB,CJP
      / &         BADDRESS ASSIGN1 & 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             NAC ,,,CONT
      / &         ALU ,,DBUS,QREG,OR,OR,DZ
      / &         DBUS SDATA & DATINSRT D#32768
      ;
      ;(5) THE MULTIPLIER IN ER5 IS NEGATED AND LEFT IN ER5.
 00E7             NAC ,,,CONT
      / &         ALU ,,ABUS,NOP,SUBR,SUBR,DZ
      / &         REG ,,,R5 & CKREG & DBUS SALU
      / &         STATSHFT B#01000000000
      ;
      ;(6) THE MASK IN QREG IS NOW ANDED WITH THE NOT OF THE MULTIPLICAND IN GR1.
      ;  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             NAC ,,,CONT
      / &         ALU ,,,NOP,NOTRS,NOTRS,AQ
      / &         DBUS SALU & CKREG
      / &         REG ,,GR1,GR1,R8
      ;
      ;(7) A RETURN OCCURS IF THE OUTPUT OF THE ALU HAD MS BYTE NOT ZERO. IE THE
      ; SIGN BIT WAS SET INDICATING THAT THE MUTLIPLICAND WAS POSITIVE.
 00E9             NAC ,FALSE,ZMSB,CRTN
      / &         ALU
      ;
      ;(8) SINCE IT HAS NOW BEEN SHOWN THAT THE MULTIPLICAND IS NEGATIVE, IT IS
      ; NEGATED.
 00EA             NAC ,FALSE,TZERO,CRTN
      / &         ALU ,,,RAMF,SUBS,SUBS,ZA
      / &         REG ,,GR1,GR1
      / &         CKALU
      / &         STATSHFT B#01000000000
      ;
      ;(8) THIS IS THE CONTINUATION WHEN IT HAS BEEN SHOWN THAT THE MULTIPLIER
      ; IS POSITIVE.
 00EB ASSIGN1:    NAC ,,,CONT
      / &         ALU ,,ABUS,NOP,AND,AND,DA
      / &         REG ,,GR1,GR1,R8
      / &         DBUS SALU & CKREG
```



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

      ;
      ;(9)  RETURN OCCURS IF THE SIGN BIT WAS ZERO.
 00EC             NAC ,TRUE,ZMSB,CRTN
      / &         ALU
      ;
      ;(10) THE MULTIPLICAND IN GR1 IS NEGATED AND RETURN TAKES PLACE.
 00ED             NAC ,FALSE,TZERO,CRTN
      / &         ALU ,,,RAMF,SUBS,SUBS,ZA
      / &         REG ,,GR1,GR1
      / &         CKALU
      / &         STATSHFT B#01000000000
      ;
      ;-------------------------------------------------------------------------------
      ; THIS SUBROUTINE MULTIPLIES A SINGLE PRECISION POSITIVE INTEGER IN GR1 BY
      ; ANOTHER SINGLE PRECISION POSITIVE INTEGER IN ER5.  THE RESULT, A DOUBLE
      ;PRECISION POSITIVE INTEGER IS LEFT IN GR1 (MS WORD) AND QREG (LS WORD).
      ;
      ;  THE MULTIPIER IN ER5 IS PLACED IN QREG.
 00EE ASMULT:     NAC ,,,CONT
      / &         ALU ,,ABUS,QREG,OR,OR,DZ
      / &         REG ,,,R5
      / &         CKALU
      ;
      ;   THE MULTIPLICAND IS PLACED IN ER9
 00EF             NAC ,,,CONT
      / &         ALU ,,,NOP,OR,OR,ZA
      / &         REG ,,GR1,GR1,R9
      / &         DBUS SALU & CKREG
      ;
      ;
      ;
      ;(20) THE MULTIPLICATION PROCESS BEGINS.
      ;     GR1 IS CLEARED, GR1 AND QREG ARE DOWNSHIFTED. ZERO IS PUT IN AT THE TOP
      ;      OF GR1. THE MICRO-PROGRAM COUNTER VALUE PLUS ONE IS PUSHED INTO THE
      ;      MICRO-PROGRAM SEQUENCER STACK (2911 & 2909).
      ;      THE NUMBER D#15 IS LOADED INTO THE COUNTER IN THE CCU.
      ;      THE EXTERNAL CARRY LATCH IS LOADED WITH THE LS BIT FROM THE QREG.
 00F0 SPIM2:      NAC ,FALSE,TZERO,PUSH
      / &         ALU ,,,RAMQD,AND,AND,ZA
      / &         REG ,,GR1,GR1
      / &         CKALU
      / &         CARRYSEL CQLSB
      / &         BADDRESS B#0000000001110
      ;
      ;
```

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

      ;(21) JUMP TO SPIM3 TAKES PLACE IF THE CARRY LATCH WAS SET MEANING THAT THE
      ; LS BIT OF THE Q REGISTER WAS ONE.
00F1          NAC ,TRUE,CARR,CJP
      / &     BADDRESS SPIM3
      / &     ALU
      ;
      ;(22) GR1 AND QREG ARE DOWNSHIFTED. THE LS BIT OF GR1 GOES INTO THE MS
      ;  BIT OF QREG.  CARRY IS SET ACCORDING TO THE LS BIT OF THE QREG.
      ;  IF THE COUNTER IS ZERO CONTINUE OCCURS.
      ;  IF THE COUNTER IS NOT ZERO, IT IS DECREMENTED AND JUMP TAKES PLACE TO
      ;  THE ADDRESS HELD IN THE STACK WHICH IS (21) ABOVE.
00F2          NAC ,FALSE,COUNT,RFCT
      / &     ALU ,,,RAMQD,OR,OR,ZA
      / &     REG ,,GR1,GR1
      / &     CARRYSEL CQLSB
      / &     STATSHFT B#0001100000000
      / &     CKALU
      ;
      ;
      ; (23) THE MULTIPLICATION IS NOW COMPLETE AND AN UNCONDITIONAL RETURN TAKES PLACE
00F3          NAC ,FALSE,TZERO,CRTN
      / &     ALU
      ;
      ;
      ;(24) THE MULTIPLICAND IN ER9 IS ADDED TO THE PARTIAL PRODUCT IN GR1.
      ;   THE CONTENTS OF GR1 AND QREG ARE DOWNSHIFTED.
      ;    THE LS BIT OF GR1 GOES INTO THE MS BIT OF THE QREG.
      ;   LS BIT OF QREG GOES INTO THE FX CARRY LATCH.
      ;   IF THE COUNTER IS ZERO CONTINUE TAKES PLACE.
      ;   IF THE CONTER IS NOT ZERO, THE COUNTER IS DECREMENTED AND JUMP
      ;    TO THE ADRESS IN THE STACK TAKES PLACE WHICH IS TO
      ;    (21) ABOVE.
00F4 SPIM3:   NAC ,FALSE,COUNT,RFCT
      / &     ALU .,DBUS,RAMQD,ADD,ADD,DA
      / &     REG .,GR1,GR1,R9
      / &     CARRYSEL CQLSB
      / &     STATSHFT B#0001100000000
      / &     CKALU
      / &     DBUS SREG
      ;
      ;
      ;(25) THE MULTIPLICATION IS NOW COMPLETE AND UNCONDITIONAL RETURN TAKES PLACE
00F5          NAC ,FALSE,TZERO,CRTN
      / &     ALU
      ;
```



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

      ;-------------------------------------------------------------------------
      ;
      ;-------------------------------------------------------------------------
      ;
      ; 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.
00F6 ASDLSTG2:  NAC ,,,CONT
      / &       ALU ,B#0,,NOP,OR,OR,ZA & REG ,,,GR2
      / &       CARRYSEL CZERO
      / &       STATSHFT B#0000000010000
      ;
      ;
      ;(2) THE CONTENTS OF THE Q REG ARE LOADED INTO GR2+1
      ;     THE MICRO STATUS REGISTER IS LOADED ACCORDINGLY.
00F7          NAC ,,,CONT
      / &       ALU B#0,,,RAMF,OR,OR,ZQ
      / &       CKALU & REG P#01,,,GR2
      / &       STATSHFT B#0000000010000
      ;
      ;(3) RETURN TAKES PLACE IF THE PREVIOUS LOADING OF THE MICRO STATUS REGISTER
      ;  SET MICRO Z IMPLYING THAT THE LS WORD OF THE PAIR WAS ZERO.
00F8          NAC ,TRUE,MUX,CRTN
      / &       ALU ,,,NOP,OR,OR,ZA
      / &       REG ,,,GR2 & STATSHFT B#0000000000100
      ;
      ;(4) RETURN TAKES PLACE IF THE ZERO BIT OF THE MACHINE STATUS REGISTER IS
      ; NOT SET.
00F9          NAC ,FALSE,MUX,CRTN
      / &       ALU &  STATSHFT B#0000000100100
      ;
      ;(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 MZ AND MP ENABLED.
00FA          NAC ,,,CONT
      / &       ALU ,B#0 &  MACSTEN B#0110
      / &       STATSHFT B#0000000000101
      ;
      ;(6) RETURN TAKES PLACE.
00FB          NAC ,FALSE,TZERO,CRTN & ALU
      ;-------------------------------------------------------------------------
```

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

        ;
        ; OVERFLOW ROUTINE
        ; ****************
        ;
        ;(1) THE PATTERN 0000100002222020 IS LOADED INTO THE ER0
 00FC OVERFLOW:  NAC ,,,CONT & ALU & DBUS SDATA & CKREG
        / &       DATINSRT D#2048 & REG ....R0
        ;
        ;(2) THE CONTENTS OF THE ER0 ARE ORED WITH AND LOADED INTO THE PENDING
        ; INTERRUPT REGISTER.
 00FD           NAC ,FALSE,TZERO,CRTN & ALU & DBUS SREG & RW3 ....R0
        / &       FNAUCK & STATSHFT B#0010000000000
        ;
        ;-----------------------------------------------------------------------
        ;
        ; ENTRY POINTS TO EXTENTED OPCODE HANDLERS
        ; ****************************************
        ;
        ; MAP JUMP TO ANY OF THE ENTRY POINTS BELOW RESULTS IN A MODIFIED JUMP
        ; PIPELINE TO LOCATIONS DEFINED IN MEMORY AT MULTIPLES OF 16. THE JUMP
        ; ADDRESS IS ORED WITH THE CONTENTS OF GR1 OR SR2 FIELD OF THE INSTRUCTION
        ; RESULTING IN A JUMP LOCATION DEPENDENT ON THE CONTENTS OF THE RELEVANT
        ; OPCODE EXTENSION FIELD.  AT THE JUMP LOCATION, AN UNCONDITIONAL JUMP
        ; OCCURS TO THE START OF THE APPROPRIATE EMULATION SEQUENCE WITH A LABEL
        ; OF THE FORM XOPCDXXY, WHERE XX IS THE OPCODE AND Y IS THE EXTENSION.
        ;
        ;
 00FE OPCODE40:: NAC P#1,,,JP & BADDRESS SWITCH40 & ALU  & REG ...GR1
        ;
 00FF OPCODE4A:: NAC P#1,,,JP & BADDRESS SWITCH4A & ALU  &  REG ...GR1
        ;
        ;-----------------------------------------------------------------------
        ;
        ;
        ; FETDIADD -- THIS ROUTINE FETCHES THE CONTENTS OF THE MEMORY LOCATION
        ; FOLLOWING THAT OF THE CURRENT LOCATION.  IE THE SECOND WORD OF A TWO WORD
        ; INSTRUCTION. THE WORD IS PLACED IN ER2 AND THE CONTENTS OF THE SPECIFIED
        ; INDEX REGISTER IS ADDED TO IT (IF ANY).  THE CONTENTS OF ER2 THEN REPRESENTS
        ; THE CORRECT ADDRESS FOR THE FIRST WORD (OR ONLY WORD) OF A DIRECT OR
        ; DIRECT-INDEXED DERIVED OPERAND.
        ; HAVING CARRIED OUT THIS PROCESS, THE PROGRAM COUNTER IS INCREMENTED TO
        ; TAKE ACCOUNT OF THE FACT THAT THE NEXT INSTRUCTION WILL BE FOUND NEXT BUT
```



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

        ; ONE TO THE PRESENT INSTRUCTION.
        ;
        ;(1) MEMORY ACCESS IN THE READ MODE TAKES PLACE, THE PROGRAM COUNTER VALUE
        ;  IS INCREMENTED.
 0100 FETDIADD: NAC ,,,CONT & ALU ,,APUS,NOP,ADD,ADD,DZ & REG ....RW & CKPORT
        / &       READ & STATSHFT B#0100000000000 & DBUS SALU
        / &       CKREG
        ;
        ;(2) THE CONTENTS OF RW IN ARE PLACED IN ER2, ZERO IS PLACED IN IR0.
 0101           NAC ,,,CONT & ALU ,,RAMA,AND,AND,DA & REG ,,RZERO,RZERO,R2
        / &       OR ,, SALU & CKALU & CKREG
        ;
        ;(3) THE CONTENTS OF SR2 ARE ADDED TO THE CONTENTS OF THE MEMORY DATA REGISTER
        ; AND THE RESULT PLACED IN IR0.
 0102           NAC ,,,CONT & ALU ,,DPUS,RAMP,ADD,ADD,DA & REG ,,RZERO,GR2 & CKALU
        / &       DBUS SMEM & STATSHFT B#0000000000000
        ;
        ;(4) THE CONTENTS OF IR0 ARE SWOPPED WITH THE CONTENTS OF ER2 SO THAT
        ; THE OPERAND ADDRESS IS LEFT IN ER2 AND THE ORIGINAL CONTENTS OF IR0 ARE
        ; RESTORED.
 0103           NAC ,FALSE,TZERO,CRTN & ALU ,,APUS,RAMA,OR,OR,DZ & CKALU & CKREG
        / &       DBUS SALU & REG ,,RZERO,RZERO,R2
        ;
        ;-----------------------------------------------------------------------
        ;
        ;
        ;-----------------------------------------------------------------------
        ;
        ; EXTENDED OPCODE HANDLER
        ;
        ALIGN 16
 0110 ;
 0110 SWITCH40:  NAC ,,,JP & ALU &  BADDRESS XOPCD400
 0111            NAC ,,,JP & ALU &  BADDRESS ILLEGOP1
 0112            NAC ,,,JP & ALU &  BADDRESS XOPCD402
 0113            NAC ,,,JP & ALU &  BADDRESS ILLEGOP1
 0114            NAC ,,,JP & ALU &  BADDRESS XOPCD404
 0115            NAC ,,,JP & ALU &  BADDRESS XOPCD405
 0116            NAC ,,,JP & ALU &  BADDRESS ILLEGOP1
 0117            NAC ,,,JP & ALU &  BADDRESS ILLEGOP1
 0118            NAC ,,,JP & ALU &  BADDRESS ILLEGOP1
 0119            NAC ,,,JP & ALU &  BADDRESS ILLEGOP1
 011A            NAC ,,,JP & ALU &  BADDRESS ILLEGOP1
 011B            NAC ,,,JP & ALU &  BADDRESS ILLEGOP1
```

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

 011C            NAC ...JP & ALU &  BADDRESS XOPCD40C
 011D            NAC ...JP & ALU &  BADDRESS ILLEGOP1
 011E            NAC ...JP & ALU &  BADDRESS XOPCD40E
 011F            NAC ...JP & ALU &  BADDRESS XOPCD40F
      ;
      ;
      ;
 0120 SWITCH4A:  NAC ...JP & ALU &  BADDRESS ILLEGOP
 0121            NAC ...JP & ALU &  BADDRESS XOPCD4A1
 0122            NAC ...JP & ALU &  BADDRESS XOPCD4A2
 0123            NAC ...JP & ALU &  BADDRESS ILLEGOP1
 0124            NAC ...JP & ALU &  BADDRESS XOPCD4A4
 0125            NAC ...JP & ALU &  BADDRESS ILLEGOP1
 0126            NAC ...JP & ALU &  BADDRESS XOPCD4A6
 0127            NAC ...JP & ALU &  BADDRESS XOPCD4A7
 0128            NAC ...JP & ALU &  BADDRESS XOPCD4A8
 0129            NAC ...JP & ALU &  BADDRESS XOPCD4A9
 012A            NAC ...JP & ALU &  BADDRESS XOPCD4AA
 012B            NAC ...JP & ALU &  BADDRESS XOPCD4AB
 012C            NAC ...JP & ALU &  BADDRESS ILLEGOP1
 012D            NAC ...JP & ALU &  BADDRESS ILLEGOP1
 012E            NAC ...JP & ALU &  BADDRESS ILLEGOP1
 012F            NAC ...JP & ALU &  BADDRESS ILLEGOP1
      ;-----------------------------------------------------------------------
      ;-----------------------------------------------------------------------
      ;-----------------------------------------------------------------------
      ;
      ; INPUT/OUTPUT INSTRUCTION
      ; ************************
      ;
      ALIGN 16
      ;
      ;(1) JUMP SUB TO FETDIADD TO FETCH THE DIRECT/DIRECT-INDEXED ADDRESS WHICH WILL
 0130 ; BE USED AS THE I/O COMMAND.
 0130 OPCODE48:: NAC ,FALSE,TZERO,CJS & ALU &  BADDRESS FETDIADD
      ;
      ;(2)THE CONTENTS OF REGISTER GR1 ARE TRANSFERRED TO ER5.
 0131            NAC ...CONT & ALU ...NOP,OR,OR,ZA & REG ...SR1,R5 & DBUS SALU
      / &        CKREG
      ;
      ;(3) GR1 IS LOADED WITH THE I/O COMMAND FROM ER2.
 0132            NAC ,FALSE,TZERO,PUSH & ALU ..ABUS,RAMF,OR,OR,DZ & REG .,GR1,.R2
      / &        CKALU & DATINSRT D#2 & DBUS SDATA
      ;
      ;(4) THE CONTENTS OF GR1 ARE UPSHIFTED 4 TIMES.
```



```
AMDOS/29 AMDASM MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EMSL
 0133            NAC ,FALSE,COUNT,RPCT & ALU ...RAMU,OR,OR,ZA & REG .,GR1,GR1 & CKALU
      / &        STATSHFT D#0010000000
      ;
      ;(5) THE INSTRUCTION REGISTER IS LOADED WITH THE CONTENTS OF GR1, GR1 IS
      ; LOADED WITH ITS ORIGINAL CONTENTS FROM ER5.
 0134            NACP ...CONT & ALU ..ABUS,RAMA,OR,OR,DZ & REG .,GR1,GR1,R5 & CKALU
      / &        DBUS SALU
      ;
      ;(6) IN ORDER TO BRANCH TO AN ADDRESS DEPENDENT ON THE MOST SIGNIFICANT FOUR
      ;  BIT OF TH 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            NAC B#1,..JP & ALU & REG ...GR2 &  BADDRESS IOBRNCH1
      ;-----------------------------------------------------------------------
      ;
      ; ILLEGAL OPCODE ROUTINE
      ; **********************
      ;
      ;(1) THE QREGISTER IS LOADED WITH A WORD WHICH HAS A ONE IN THE ILLEGAL OPCODE
      ; BIT POSITION FOR THE FAULT REGISTER AND ZERO EVERYWHERE ELSE.
 0136 ILLEGOP::   NAC ...CONT & ALU ..DBUS,QREG,OR,OR,DZ & DBUS SDATA
      / &        DATINSRT D#1024 & CKALU
      ;
      ;(2) THE CONTENTS OF THE FAULT REGISTER (ER3) IS ORED WITH THE WORD IN THE QREG
      ;  SO THAT THE ILLEGAL OPCODE BIT IS SET IN THE FAULT REG.
 0137 ILLEGOP2:  NAC ...CONT & ALU ..ABUS,NOP,OR,OR,DQ & DBUS SALU & REG ....R3
      / &        CKREG
      ;
      ;(3) JUMP TAKES PLACE TO FAULTINT. THE MICROSTACK IS POPPED.
 0138            NAC ,FALSE,TZERO,CJPP & ALU &  BADDRESS FAULTINT
      ;
      ;-----------------------------------------------------------------------
      ;
      ;
      ; THE NEXT 4 INSTRUCTIONS ELLIMINATE I/O COMMANDS OTHER THAN IOOI.
      ; 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 BRCH1@@X:  NACP ...CONT & ALU ..ABUS,QREG,OR,OR,DZ & CKALU & DBUS SREG
```

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

      / &         REG ,,,,R2
      ;
      ; THE CONTENTS OF THE Q REGISTER ARE MASKED WITH (0000111111110000)
0138          NAC ,,,CONT & ALU ,,DBUS,QREG,AND,AND,DQ & CKALU & DRUS SDATA
      / &         DATINSRT D#4080
      ;
      ; A JUMP TO BCRI00X1 TAKES PLACE IF ZLSB WAS NOT SET.  A JUMP WITH POP STACK
      ; WILL THEN TAKE PLACE TO ILLEGOP.
0139          NAC ,FALSE,ZLSB,CJP & ALU ,,,NOP,OR,OR,ZQ
      / &         BADDRESS BCRI00X1
      ;
      ; A MODIFIED RETURN TAKES PLACE USING BITS 12 THROUGH 15 OF THE I/O COMMAND
      ; AS THE MODIFIER IF ZMSB WAS SET BY THE LAST INSTRUCTION.
013C          NAC B#1,TRUE,ZMSB,CRTN & ALU &  REG ,,,SR2
      ;
      ; JUMP TO ILLEGOP.  THE MICRO STACK IS POPPED SO AS TO MAINTAIN THE STACK
      ; CORRECTLY.
013D BCRI00X1: NAC ,FALSE,TZERO,CJPP & ALU &  BADDRESS ILLEGOP
      ;-------------------------------------------------------------------------
      ;
      ALIGN 16
0140  ;
0140  IOBRNCH1:   NAC ,,,JP & ALU &  BADDRESS OPUTEX0 ; CMD 0XXX
0141              NAC ,,,JP & ALU &  BADDRESS ILLEGOP ; CMD 1XXX
0142              NAC ,,,JP & ALU &  BADDRESS OPUTEX2 ; CMD 2XXX
0143              NAC ,,,JP & ALU &  BADDRESS ILLEGOP ; CMD 3XXX
0144              NAC ,,,JP & ALU &  BADDRESS OPUTEX4 ; CMD 4XXX
0145              NAC ,,,JP & ALU &  BADDRESS ILLEGOP ; CMD 5XXX
0146              NAC ,,,JP & ALU &  BADDRESS ILLEGOP ; CMD 6XXX
0147              NAC ,,,JP & ALU &  BADDRESS ILLEGOP ; CMD 7XXX
0148              NAC ,,,JP & ALU &  BADDRESS INPUTEX8 ;CMD 8XXX
0149              NAC ,,,JP & ALU &  BADDRESS ILLEGOP ;CMD 9XXX
014A              NAC ,,,JP & ALU &  BADDRESS INPUTEXA ;CMD AXXX
014B              NAC ,,,JP & ALU &  BADDRESS ILLEGOP ; CMD BXXX
014C              NAC ,,,JP & ALU &  BADDRESS INPUTEXC ; CMD CXXX
014D              NAC ,,,JP & ALU &  BADDRESS ILLEGOP ; CMD DXXX
014E              NAC ,,,JP & ALU &  BADDRESS ILLEGOP ; CMD EXXX
014F              NAC ,,,JP & ALU &  BADDRESS ILLEGOP ; CMD FXXX
      ;-------------------------------------------------------------------------
      ;-------------------------------------------------------------------------
      ;
      ; READ INTERRUPT MASK
      ; *******************
      ;
      ; (1) THE INSTRUCTION IN ER1 IS LOADED INTO THE INSTRUCTION REGISTER.
```



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

0150 RINTMSK:   NACF ,,,CONT & ALU & DBUS SREG & REG ,,,,R1
      ;
      ;(2) THE MACHINE INTERRUPT MASK IN ER15 IS LOADED INTO QREG.
0151          NAC ,,,CONT & ALU ,,ABUS,QREG,OR,OR,DZ & CKALU
      / &          REG ,,,,R15
      ;
      ;(3) THE CONTENTS OF THE Q REG ARE INVERTED AND PLACED IN GR1.
0152          NAC ,FALSE,TZERO,CRTN & ALU ,,,RAMF,FXNOR,FXNOR,ZQ & CKALU & REG ,,GR1
      ;-------------------------------------------------------------------------
      ;
      ; READ STATUS WORD
      ; ****************
      ;
      ;(1) THE INSTRUCTION WORD IN ER1 IS PLACED BACK IN THE INSTRCUTION REGISTER.
0153 RSTATWRD:  NACF ,,,CONT & ALU & DBUS SREG & REG ,,,,R1
      ;
      ;(2) THE STATUS WORD IS READ AND PLACED IN GR1. RETURN TAKES PLACE
0154          NAC ,FALSE,TZERO,CRTN & ALU ,,IBUS,RAMF,OR,OR,DZ & REG ,,GR1
      / &         CKALU & DBUS ISTATUS & STATSHFT B#0000001000000
      ;-------------------------------------------------------------------------
      ;
      ;
      ; READ AND CLEAR FAULT REGISTER
      ; *****************************
      ;
      ;(1) THE INSTRUCTION WORD IN ER1 IS PLACED IN THE INSTRUCTION REG
0155 FAULT:     NACF ,,,CONT & ALU & DBUS SREG & REG ,,,,R1
      ;
      ;(2) THE CONTENTS OF THE FAULT REGISTER ARE LOADED INTO GR1, THE FAULT REG (FR3)
      ;   IS CLEARED. RETURN TAKES PLACE.
0156          NAC ,FALSE,TZERO,CRTN & ALU ,,ABUS,RAMF,OR,OR,DZ & DBUS SDATA
      / &         REG ,,GR1,,R3 & CKALU & CKREG & DATINSRT D#0
      ;-------------------------------------------------------------------------
      ;-------------------------------------------------------------------------
      ;
      ; READ CPU STATE WORD (NOT STATUS!!)
      ; **********************************
      ;
0157 RSTATE:    NACF ,,,CONT & ALU & DBUS SREG & REG ,,,,R1
      ;
      ;(2) THE MACHINE STATE WORD (IN ER4) IS PLACED IN GR1 - RETURN TAKES PLACE.
0158          NAC ,FALSE,TZERO,CRTN & ALU ,,ABUS,RAMF,OR,OR,DZ & REG ,,GR1,,R4
```

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

      / &         CKALU
      ;
      ;-------------------------------------------------------------------------------
      ;
      ; READ ENGINEER'S SWITCHFIELD
      ; ***************************
      ;
      ;(1) THE INSTRUCTION IN ER1 IS LOADED INTO THE INSTRUCTION REGISTER.
0159 RMANDT:     NACF ,,,CONT & ALU & DBUS SREG & REG ,,,,R1
      ;
      ;(2) THE MANUAL DATA SWITCHFIELD ON THE ENGINEER'S CONTROL BOX IS LOADED
      ;    INTO GR1.
      ;    RETURN TAKES PLACE.
015A             NAC ,FALSE,TZERO,CRTN & ALU ,,DBUS,RAMF,OR,OR,DZ & DBUS SMANDT
      / &        CKALU & REG ,,GR1
      ;
      ;
      ;-------------------------------------------------------------------------------
      ; SET PENDING INTERRUPT REGISTER
      ; ******************************
      ;
      ;(1) THE CONTENTS OF THE PENDING INTERRUPT REGISTER IN BOTH MACHINE INTERRUPT
      ; UNITS ARE CLEARED.
015B STPDGINT:   NACF ,,,CONT & ALU & DBUS SREG & REG ,,,,R1 & MACINT CLRIN
      / &        ENAUICK & STATSHFT @@@@@@@11@@@
      ;
      ;(2) THE CONTENTS OF GR1 ARE LOADED INTO THE PENDING INTERRUPT REGISTER.
      ;  RETURN TAKES PLACE.
015C             NAC ,FALSE,TZERO,CRTN & ALU ,,,NOP,OR,OR,ZA & REG ,,,GR1 & DBUS SALU
      / &        ENAUICK & STATSHFT B#@@1@@@@@@@@
      ;
      ;
      ORG RINTMSK+16-1
      ;
      ; PROCESSOR AND AUXILLIARY REGISTER CONTROL ( COMMAND GROUP A@@X )
      ;
      ; 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 BRCHI@@X IN ORDER TO ELIMINATE THE CASE WHERE THE
015F  ; CENTRE 8 BITS OF THE COMMAND WORD IS NON ZERO.
015F INPUTEXA:   NAC ,FALSE,TZERO,CJS & ALU &  BADDRESS BRCHI@@X
      ;
      ;
0160             NAC ,,,JP & ALU &  BADDRESS RINTMSK  ; CMD A@@@
```



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

0161             NAC ,,,JP & ALU &  BADDRESS RSTATE  ; CMD A@@1
0162             NAC ,,,JP & ALU &  BADDRESS RMANDT  ; CMD A@@2
0163             NAC ,,,JP & ALU &  BADDRESS ILLEGOP  ; CMD A@@3
0164             NAC ,,,JP & ALU &  BADDRESS ILLEGOP  ; CMD A@@4
0165             NAC ,,,JP & ALU &  BADDRESS ILLEGOP  ; CMD A@@5
0166             NAC ,,,JP & ALU &  BADDRESS ILLEGOP  ; CMD A@@6
0167             NAC ,,,JP & ALU &  BADDRESS ILLEGOP  ; CMD A@@7
0168             NAC ,,,JP & ALU &  BADDRESS ILLEGOP  ; CMD A@@8
0169             NAC ,,,JP & ALU &  BADDRESS ILLEGOP  ; CMD A@@9
016A             NAC ,,,JP & ALU &  BADDRESS ILLEGOP  ; CMD A@@A
016B             NAC ,,,JP & ALU &  BADDRESS ILLEGOP  ; CMD A@@B
016C             NAC ,,,JP & ALU &  BADDRESS ILLEGOP  ; CMD A@@C
016D             NAC ,,,JP & ALU &  BADDRESS ILLEGOP  ; CMD A@@D
016E             NAC ,,,JP & ALU &  BADDRESS RSTATWRD ; CMD A@@F
016F             NAC ,,,JP & ALU &  BADDRESS RFAULT   ; CMD A@@F
      ;
      ;-------------------------------------------------------------------------------
      ;
      ; PIO OUTPUT
      ; **********
      ;
      ;(1) THE INSTRUCTION IS RELOADED INTO THE INSTRUCTION REGISTER.
0170 OPUTEX@:  NACF ,,,CONT & ALU & DBUS SREG & REG ,,,,R1
      ;
      ;(1A) A MASK WITH ALL BUT THE TOP FOUR BITS SET IS LOADED INTO THE Q REG.
0171             NAC ,,,CONT & ALU ,,DBUS,QREG,OR,OR,DZ & DBUS SDATA & CK&LU
      / &        DATINSRT D#4@95
      ;
      ;(1B) THE CONTENTS OF ER2 ARE ANDED WITH THE CONTENTS OF THE Q REG AND THE
      ; RESULT PLACED BACK IN ER2.
0172             NAC ,,,CONT & ALU ,,ABUS,NOP,AND,AND,DQ & CKREG & DBUS SALU
      / &        REG ,,,,R2
      ;
      ;
      ;(2) THE CONTENTS OF GR1 ARE OUTPUT ON THE MEMORY BUS USING THE CONTENTS
      ; OF ER2 AS THE ADDRESS.  BIT 8 OF THE AUICK 1 FIELD IS HELD HIGH TO SHOW
      ; THAT THE OPERATION IS AN I/O OPERATION AND NOT A MEMORY OPERATION.
0173             NAC ,FALSE,TZERO,CRTN & ALU ,,,NOP,OR,OR,ZA & REG ,,,GR1,R2 & DBUS SALU
      / &        CKPORT & WRITE & ENAUICK & STATSHFT B#@@@1@@@@@@@@
      ;
      ;-------------------------------------------------------------------------------
      ;
      ; PIO INPUT
      ; *********
      ;
      ;(1) THE INSTRUCTION IS RELOADED INTO THE INSTRUCTION REGISTER.
```

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

 0174 INPUTEI8: NACF ,,,CONT & ALU & DBUS SREG & REG ,,,,R1
      ;
      ;(1A) A MASK WITH ALL BUT THE TOP FOUR BITS SET IS LOADED INTO THE Q REG.
 0175           NAC ,,,CONT & ALU ,,DBUS,QREG,OR,OR,DZ & DBUS SDATA & CKALU
      / &       DATAWSRT D#4095
      ;
      ;(1B) THE CONTENTS OF ER2 ARE ANDED WITH THE CONTENTS OF THE QREG AND THE RESULT
      ; PLACED IN ER2.
 0176           NAC ,,,CONT & ALU ,,APUS,NOP,AND,ANL,DQ & CKREG & [illegible] SALU
      / &       REG ,,,,R2
      ;
      ;(2) GR1 IS LOADED FROM THE MEMORY BUS USING THE CONTENTS OF FR1 AS THE ADDRESS.
      ;  BIT 8 OF THE AUX OR FIELD 1 IS HELD HIGH TO SHOW PERIPHERAL DEVICES THAT
      ; THE OPERATION IS AN I/O OPERATION.
 0177           NAC ,,,CONT & ALU & REG ,,,,R2 & CKPORT & READ
      / &       ENAUECK & STATSHFT B#####2000000
      ;
      ;(3) THE DATA FROM THE I/O DEVICE IS LOADED INTO GR1
 0178           NAC ,FALSE,TZERO,CRTN & ALU ,,DBUS,RAMF,OR,OR,DZ & CKALU & DBUS SMEM
      / &       REG ,,GR1
      ;
      ;-------------------------------------------------------------------------------
      ;
      ; READ TIMER A
      ;
      ;(1) THE INSTRUCTION IS RESTORED TO THE INSTRUCTION REGISTER
 0179 INPTIMA:  NACF ,,,CONT & ALU & DBUS SREG & REG ,,,,R1
      ;
      ;(2) THE CONTENTS OF TIMER A ARE TRANSFERRED TO GR1
 017A           NAC ,FALSE,TZERO,CRTN & ALU ,,DBUS,RAMF,OR,OR,DZ & REG ,,GR1
      / &       CKALU & DBUS STIMA
      ;
      ;-------------------------------------------------------------------------------
      ;
      ; INPUT TIMER B
      ; *************
      ;
      ;(1) THE INSTRUCTION IS RESTORED TO THE INSTRUCTION REGISTER.
 017B INPTIMB:  NACF ,,,CONT & ALU & DBUS SREG & REG ,,,,R1
      ;
      ;(2) THE CONTENTS OF TIMER B ARE TRANSFERRED TO GR1
 017C           NAC ,FALSE,TZERO,CRTN & ALU ,,DBUS,RAMF,OR,OR,DZ & REG ,,GR1
      / &       CKALU & DBUS STIMB
      ;
      ;-------------------------------------------------------------------------------
      ;
```



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

      ;
      ; PROCESSOR AND AUXILLIARY REGISTER CONTROL  (COMMAND GROUP C0[illegible])
      ; ************************************************
      ;
      ORG OPUTEI0+15
      ; THIS SECTION DISTINGUISHES BETWEEN CASES WITH DIFFERENT VALUES OF THE LEAST
      ; SIGNIFICANT FOUR BITS OF THE I/O COMMAND.
      ;
 017E ;(1) JUMP SUB TO BRCHIWAR.
 017F INPUTEIC: NAC ,FALSE,TZERO,CJS & ALU &  BADDRESS BRCHIW0R
      ;
      ;
 0180           NAC ,,,JP & ALU &  BADDRESS ILLEGOP ; CMD C000
 0181           NAC ,,,JP & ALU &  BADDRESS ILLEGOP ; CMD C001
 0182           NAC ,,,JP & ALU &  BADDRESS ILLEGOP ; CMD C002
 0183           NAC ,,,JP & ALU &  BADDRESS ILLEGOP ; CMD C003
 0184           NAC ,,,JP & ALU &  BADDRESS ILLEGOP ; CMD C004
 0185           NAC ,,,JP & ALU &  BADDRESS ILLEGOP ; CMD C005
 0186           NAC ,,,JP & ALU &  BADDRESS ILLEGOP ; CMD C006
 0187           NAC ,,,JP & ALU &  BADDRESS ILLEGOP ; CMD C007
 0188           NAC ,,,JP & ALU &  BADDRESS ILLEGOP ; CMD C008
 0189           NAC ,,,JP & ALU &  BADDRESS ILLEGOP ; CMD C009
 018A           NAC ,,,JP & ALU &  BADDRESS INPTIMA ; CMD C00A
 018B           NAC ,,,JP & ALU &  BADDRESS ILLEGOP ; CMD C00B
 018C           NAC ,,,JP & ALU &  BADDRESS ILLEGOP ; CMD C00C
 018D           NAC ,,,JP & ALU &  BADDRESS ILLEGOP ; CMD C00D
 018E           NAC ,,,JP & ALU &  BADDRESS INPTIMB ; CMD C00E
 018F           NAC ,,,JP & ALU &  BADDRESS ILLEGOP ; CMD C00F
      ;
      ;-------------------------------------------------------------------------------
      ;
      ; SET INTERRUPT MASK
      ; ******************
      ;
      ; 1750 DEFINES INTERRUPTS TO BE PREVENTED FROM OCCURING WHEN THE MASK BIT IS
      ; ZERO.  THE 2914 DISALLOWS INTERRUPTS WHEN THE MASK BIT IS ONE.  THE CONTENTS
      ; OF RA MUST THEREFORE BE INVERTED BEFORE BEING LOADED INTO THE 2914 MASK
      ; REGISTERS.
      ;
      ;(1) THE CONTENTS OF FR1 (INSTRUCTION) ARE LOADED INTO THE INSTRUCTION
      ;  REGISTER.
 0190 SETIMSK: NACF ,,,CONT & ALU & DBUS SREG & REG ,,,,R1
      ;
      ;(4) THE CONTENTS OF GR1 ARE INVERTED AND LOADED INTO ER15
 0191           NAC ,,,CONT & ALU ,,,NOP,EXNOR,EXNOR,ZA & REG ,,GR1,R15 & DBUS SALU
```

```
      / &       CKREG
      ;-------------------------------
      ; MIMICINT
      ; ********
      ; THIS SUBROUTINE MIMICS THE CONTENTS OF THE REGISTER ER15 IN THE 2914
      ; MASK REGISTERS.   THE MIMICING TAKES ACCOUNT OF WHETTHER INTERRUPTS ARE
      ; DISABLED OR NOT.
      ;
      ;(2) A MASK @@@@@@@@@1@@@@@@ (BIT 9 SET) IS LOADED INTO QREG.
@192 MIMICINT: NAC ,,,CONT & ALU ,,DBUS,QREG,OR,OR,DZ & DBUS SDATA & DATINSRT D#64
      / &       CKALU
      ;
      ;(3) THE MACHINE STATE WORD IN ER4 IS ANDED WITH THE MASK IN QREG
@193          NAC ,,,CONT & ALU ,,ABUS,NOP,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 CHIPS BUT ALWAYS SETS BIT @ IN 2914 TO ZERO TO TAKE ACCOUNT OF THE FACT THAT
      ; POWER DOWN CANNOT BE MASKED.
@194          NAC ,FALSE,ZLSB,CJP & ALU & BADDRESS ENABLINT
      ;
      ;(5) SINCE INTERRUPTS ARE DISABLED, JUMP TAKES PLACE TO DISABINT WHICH LOADS
      ; THE SPECIAL DISABLE MASK INTO THE 2914 CHIPS.
@195          NAC ,,,JP & ALU & BADDRESS DISABINT
      ;
      ;-------------------------------------------------------------------------
      ;
      ; CLEAR INTERRUPT REQUESTS
      ; ************************
      ;
      ;(1) ALL MACHINE INTERRUPTS ARE CLEARED.
@196 CLRINREQ: NAC ,FALSE,TZERO,CRTN & ALU & ENAUXCK & MACINT CLRIN
      / &       STATSHFT B#@@@@@@@11@@@
      ;
      ;-------------------------------------------------------------------------
      ;
      ; 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 CHIPS.
      ; 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 @111111111111111 IS PLACED IN QREG.
@197 ENAPLINT:  NAC ,,,CONT & ALU ,,DBUS,QREG,OR,OR,DZ & DBUS SDATA & CKALU
```



```
      / &       DATINSRT D#32767
      ;
      ;(2) THE CONTENTS OF ER15 ARE ANDED WITH THE QREG AND THE RESULT PLACED IN
      ; THE MASK REGISTERS OF THE 2914 CHIPS.
@198          NAC ,FALSE,TZERO,CRTN & ALU ,,ABUS,NOP,AND,AND,DQ & REG ,,,,R15
      / &       DBUS SALU & ENAUXCK & STATSHFT B#@@@@@@11@@@ & 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'S SO THAT ALL BUT
      ; THE TWO HIGHEST PRIORITY ARE MASKED, LEAVING 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.
@199 DISABINT: NAC ,,,CONT & ALU ,,DBUS,QREG,OR,OR,DZ & DBUS SDATA & DATINSRT D#16383
      / &       CKALU
      ;
      ;(3) THE DATA IN THE Q REG IS ORED WITH THE INTERRUPT MASK IN ER15 AND LOADED
      ; INTO THE 2914 MASK REGISTERS.
      ; RETURN TAKES PLACE.
@19A          NAC ,FALSE,TZERO,CRTN & ALU ,,ABUS,NOP,OR,OR,DQ & DBUS SALU & ENAUXCK
      / &       MACINT LDM & STATSHFT B#@@@@@@11@@@ & REG ,,,,R15
      ;
      ;-------------------------------------------------------------------------
      ;
      ; RESET NORMAL POWER UP
      ; *********************
      ;
      ; THIS HAS NOT YET BEEN IMPLEMENTED
      ;
@19B RESNMPU:  NAC ,,,JP & ALU & BADDRESS ILLEGOP
      ;
      ;-------------------------------------------------------------------------
      ;
      ; WRITE STATUS WORD
      ; *****************
      ;
      ;(1) THE INSTRUCTION IN ER1 IS LOADED INTO THE INSTRUCTION REGISTER.
```

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

 019C WTSTSWD:  NACF ...CONT & ALU & DBUS SREG & REG ....R1
                ;
                ;(2) THE WORD IN GR1 IS WRITTEN INTO THE LEAST SIGNIFICANT 12 BITS OF THE
                ; MACHINE STATUS WORD.
 019D           NAC ...CONT & ALU ...NOP,OR,OR,ZA & REG ...GR1 & ENAUICK & DBUS SALU
                / &         AUICK B#01AA
                ;
                ;(3) THE MOST SIGNIFICANT FOUR BITS OF THE STATUS WORD ARE LOADED.
                ;    RETURN TAKES PLACE.
 019E           NAC ,FALSE,TZERO,CRTN & ALU ,,NOP,,NOP,OR,OR,ZA & REG ...GR1
                / &         DBUS SALU & STATSHFT B#AAAAAAAAAAAA
                ;
                ;-------------------------------------------------------------------------
                ;
                ; PROCESSOR AND AUXILLIARY REGISTER CONTROL ( COMMANDS OR OP CODES)
                ; ****************************************
                ;
                ; THIS SECTION DISTINGUISHES BETWEEN CASES WITH DIFFERENT 4 BITS OF THE LEAST
                ;  SIGNIFICANT FOUR BITS OF THE I/O COMMAND
                ;
                ORG SETINMSK+15
                ;
                ;(1)     JUMP SUB TO BRCHE## TAKES PLACE TO ELIMINATE THE CASE WITH MIDDLE
                ; 8 BITS OF I/O COMMAND NON ZERO AND BRANCH TO ACCOMODATE THE 16 4 BITS OF
                ; TH I/O COMMAND.
 019F CPUTEE2:  NAC ,FALSE,TZERO,CJS & ALU & BADDRESS BRCHE##
                ;
 01A0           NAC ...JP & ALU & BADDRESS SETINMSK
 01A1           NAC ...JP & ALU & BADDRESS CLRINPRQ
 01A2           NAC ...JP & ALU & BADDRESS ENAPRINT
 01A3           NAC ...JP & ALU & BADDRESS DISAPINT  ; CMD 2003
 01A4           NAC ...JP & ALU & BADDRESS CLAPSINT  ; CMD 2004
 01A5           NAC ...JP & ALU & BADDRESS SETPDINT  ; CMD 2005
 01A6           NAC ...JP & ALU & BADDRESS OPTPDISP  ; CMD 2006
 01A7           NAC ...JP & ALU & BADDRESS ILLEGOP   ; CMD 2007
 01A8           NAC ...JP & ALU & BADDRESS ILLEGOP   ; CMD 2008
 01A9           NAC ...JP & ALU & BADDRESS ILLEGOP   ; CMD 2009
 01AA           NAC ...JP & ALU & BADDRESS ILLEGOP   ; CMD 200A
 01AB           NAC ...JP & ALU & BADDRESS ILLEGOP   ; CMD 200B
 01AC           NAC ...JP & ALU & BADDRESS ILLEGOP   ; CMD 200C
 01AD           NAC ...JP & ALU & BADDRESS ILLEGOP   ; CMD 200D
 01AE           NAC ...JP & ALU & BADDRESS ILLEGOP   ; CMD 200E
 01AF           NAC ...JP & ALU & BADDRESS SPRTINT   ; CMD 200F
                ;
                ;-------------------------------------------------------------------------
                ;
```



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

                ; START TIMER A
                ; ***************
                ;
                ; BIT 4 OF THE STATE WORD INDICATES WHETHER TIMER A IS RUNNING OR NOT.
                ; APPROPRIATE BIT PATTERNS IN THE STATE WORD ARE MIMICED BY THE CONTENTS
                ; OF DEDICATED REGISTERS ARBITRARILY THE SYSTEM.  THESE REGISTERS ARE ALL LOADED
                ; WHEN CK STATE DISPLAY IS ACTIVE.
                ;
                ;(1) A MASK SETTING BIT 4 OF THE STATE WORD HIGH IS PLACED IN QREG.
 01B0 STRTIMA:  NA ...CONT & ALU ...NOP,ZERO,OR,OR,DZ & SCALU & DBUS SDATA
                / &         DATINSET B#40
                ;
                ;(2) THE STATE WORD IS ANDED WITH THE CONTENTS OF THE QREG AND THE
                ; CONTENTS REMAIN IN R4 AND ENTER THE STATE DISPLAY LATCH.
                ; RETURN TAKES PLACE.
 01B1           NAC ,FALSE,TZERO,CRTN & ALU ,,ABUS,NOP,OR,OR,DQ & REG ....R4
                / &         DBUS SALU & ENAUICK & AUICK B#0010
                ;
                ;-------------------------------------------------------------------------
                ;
                ; HALT TIMER A
                ; ************
                ;
                ; 1) THE QREG IS LOADED WITH A MASK WITH BIT 4 SET AND ALL ELSE CLEAR.
 01B2 HTIMA:    NAC ...CONT ,ALU ...NOP,ISREG,FINOR,FINOP,DZ & DBUS SDATA & SCALU
                / &         DATINSET B#40
                ;
                ;(2) THE CONTENTS OF THE STATE WORD ARE ANDED WITH THE CONTENTS OF THE QREG
                ; AND THE RESULT PLACED IN R4 AND THE STATE LATCHES.
 01B3           NAC ,FALSE,TZERO,CRTN & ALU ,,ABUS,NOP,AND,AND,DQ & DBUS SALU
                / &         REG ....R4 & REGS & ENAUICK & AUICK B#0010
                ;
                ;-------------------------------------------------------------------------
                ;
                ; OTPUT TIMER A
                ; *************
                ;
                ;(1)THE INSTRUCTION REGISTER ARE LOADED WITH THE INSTRUCTION IN ER1.
                ;   JUMP SUB TAKES PLACE TO STRTIMA.
 01B4 OPTIMA:   NACF ,FALSE,TZERO,CJS & ALU & DBUS SREG & REG ....R1
                / &         BADDRESS STRTIMA
                ;
                ;(2) THE CONTENTS OF GR1 ARE LOADED INTO TIMER A
                ;  RETURN TAKES PLACE.
 01B5           NAC ,FALSE,TZERO,CRTN & ALU ...NOP,OR,OR,ZA & REG ...GR1 & DBUS SALU
                / &         ENAUICK & STATSHFT B#0000010000
```

```
      ;
      ;-------------------------------------------------------------------------------
      ;
      ; 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.
      ;
      ;(1) THE QREG IS LOADED WITH A ONE IN BIT POS 5 AND ZERO ELSEWHERE.
@1B6 STRTIMB:   NAC ,,,CONT & ALU ,,DBUS,QREG,OR,OR,DZ & CKALU & DBUS SDATA
      / &       DATINSRT D#1024
      ;
      ;(2) THE STATE WORD IS ORED WITH THE CONTENTS OF THE QREGISTER AND LOADED
      ; INTO ER4 AND THE STATE LATCHES.
      ;  RETURN TAKES PLACE.
@1B7          NAC ,FALSE,TZERO,CRTN & ALU ,,ABUS,NOP,OR,OR,DQ & CKREG & DBUS SALU
      / &       ENAUICK & REG ....R4 & AUICK B#0010
      ;
      ;-------------------------------------------------------------------------------
      ;
      ; HALT TIMER B
      ; ************
      ;
      ;(1) THE QREGISTER IS LOADED WITH A ZERO IN BIT 5 AND A ONE ELSEWHERE.
@1B8 HLTIMB:    NAC ,,,CONT & ALU ,,DBUS,QREG,EXNOR,EXNOR,DZ & CKALU & DBUS SDATA
      / &       DATINSRT D#1024
      ;
      ;(2) THE STATE WORD FROM ER4 IS ANDED WITH THE CONTENTS OF THE QREGISTER
      ; AND LOADED INTO ER4 AND THE STATE LATCHES.
@1B9          NAC ,FALSE,TZERO,CRTN & ALU ,,ABUS,NOP,AND,AND,DQ & CKREG & DBUS SALU
      / &       REG ,,,R4 & ENAUICK & AUICK B#0010
      ;
      ;-------------------------------------------------------------------------------
      ;
      ; OUTPUT TIMER B
      ; **************
      ;
      ;(1) THE INSTRUCTION REGISTER IS LOADED WITH THE INSTRUCTION IN ER1.
      ; JUMP SUB TAKES PLACE TO STRTIMB.
@1BA OPTIMB:    NACF ,FALSE,TZERO,CJS & ALU & DBUS SREG & REG ,,,R1
      / &       BADDRESS STRTIMB
      ;
      ;(2) THE CONTENTS OF GR1 ARE LOADED INTO THE TIMER B.
      ;  RETURN TAKES PLACE.
@1BB          NAC ,FALSE,TZERO,CRTN & ALU ,,,NOP,OR,OR,ZA & REG ,,GR1 & DBUS SALU
```



```
      / &       ENAUICK & STATSHFT B#0000100000000
      ;
      ;-------------------------------------------------------------------------------
      ;
      ; CLEAR FAULT REGISTER
      ; ********************
      ;
      ;(1) THE FAULT REGISTER IS CLEARED AND RETURN TAKES PLACE
@1BC CLRFTRG: NAC ,FALSE,TZERO,CRTN & ALU & DBUS SDATA & DATINSRT D#0 & REG ....R5
      / &       CKRFG
      ;
      ;-------------------------------------------------------------------------------
      ;
      ; OUTPUT TO DATA DISPLAY FIELD
      ; ****************************
      ;
      ;(1) THE INSTRUCTION IS RELOADED TO THE INSTRUCTION REGISTER
@1BD OPDTDISP:  NACF & ALU & DBUS SREG & REG ....R1
      ;
      ;(2) THE DATA IN GR1 IS LOADED INTO THE DATA DISPLAY LATCHES.
      ; RETURN TAKES PLACE.
@1BE          NAC ,FALSE,TZERO,CRTN & ALU ,,,NOP,OR,OR,ZA & REG ,,GR1 & DBUS SALU
      / &       ENAUICK & AUICK B#0001
      ;
      ;
      ; PROCESSOR AND AUXILLIARY REGISTER CONTROL  ( COMMAND GROUP 4XXX )
      ;
      ORG STRTIMA+15
      ;
@1BF ;(1) JUMP TAKES PLACE TO BRCHI@@X.
@1BF OPUTEX4:   NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS BRCHI@@X
      ;
      ;
@1C0          NAC ,,,JP & ALU & BADDRESS ILLEGOP ; CMD 4000
@1C1          NAC ,,,JP & ALU & BADDRESS ILLEGOP ; CMD 4001
@1C2          NAC ,,,JP & ALU & BADDRESS ILLEGOP ; CMD 4002
@1C3          NAC ,,,JP & ALU & BADDRESS ILLEGOP ; CMD 4003
@1C4          NAC ,,,JP & ALU & BADDRESS ILLEGOP ; CMD 4004
@1C5          NAC ,,,JP & ALU & BADDRESS ILLEGOP ; CMD 4005
@1C6          NAC ,,,JP & ALU & BADDRESS ILLEGOP ; CMD 4006
@1C7          NAC ,,,JP & ALU & BADDRESS ILLEGOP ; CMD 4007
@1C8          NAC ,,,JP & ALU & BADDRESS STRTIMA ; CMD 4008
@1C9          NAC ,,,JP & ALU & BADDRESS HLTIMA ; CMD 4009
@1CA          NAC ,,,JP & ALU & BADDRESS OPTIMA ; CMD 400A
@1CB          NAC ,,,JP & ALU & BADDRESS ILLEGOP ; CMD 400B
```

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

 01CC            NAC ,,JP & ALU & BADDRESS STRTIMB ; CMD 40CC
 01CD            NAC ,,JP & ALU & BADDRESS HLTIMB ; CMD 40CD
 01CE            NAC ,,JP & ALU & BADDRESS OPTIMB ; CMD 40CE
 01CF            NAC ,,JP & ALU & BADDRESS ILLEGOP ; CMD 40CF
      ;
      ;---------- -------- ---------- -- ---------- ---- ---- ---------- --
      ;
      ; RESET PENDING INTERRUPT
      ; ***********************
      ;
      ;(1) THE INSTRUCTION IN ER1 IS LOADED INTO THE INSTRUCTION REGISTER.
      ; JUMP SUB TAKES PLACE IF  FIELD BIT
 01D0 OPPOINT:  NACF ,PALSF,IIRR ,JSB & ALU & DBUS DATA & REG ....R1
      / &       BADDRESS HARDBLBIT
      ;
      ; 2.  REGISTER ER0 IS W ITAINS A MASK WITH A BIT SET T  ONE IN THE
      ; POSITION IN THE MASK WHERE THE INTERRUPT IS TO BE CLEARED.
      ; THE PENDING INTERRUPT IS  CLEARED AT THE BIT POSITION 1 AND THE MASK
      ; IS LOADED INTO THE  REGISTER.
 01D1            NAC ,,C,NT & ALU ,,ABUS,,R,CR,DZ & REG ....R0
      / &       IBUS DATA & ENABLE C & STAT BIT B#FFFFFFFFFFFFFFFF & MA INT CLEAR
      ;
      ;(3) THE CONTENTS OF THE  REGISTER ARE MASKED WITH #FFFFFFFFFFFFFFFF IF JSB
      ; IF THE BIT IN RB  REGISTER WAS THE MACHINE ERROR INTERRUPT.
 01D2            NAC ,,CONT & ALU ,,ABUS,AND,AND,DZ & IBUS DATA
      / &       DATINSRT B#FF
      ;
      ;(4) RETURN TAKES PLACE IF JMSR WAS SET.
 01D3            NAC ,CRTN,JMSR,CRTN & ALU
      ;
      ;(5) THE FAULT REGISTER IS CLEARED.  RETURN TAKES PLACE.
 01D4            NAC ,PALSF,IZERO,CRTN & ALU ,,NOP,AND,AND,DZ & REG ....R3 & IBUS DATA
      / &       CKREG
      ;
      ;---------- -------- ---------- -------------------------- ----------
      ;---------- -------- ---------- -------------------------- ----------
      ;
      ; NO - OPERATION AND BREAKPOINT
      ; *****************************
      ;
 01D5 OPCODEFF:  NAC ,,CONT & ALU ,,ABUS,NOP,OR,OR,DZ & REG ....R.
      ;
      ;(2) RETURN TAKES PLACE IF ZLSB WAS SET BY THE LAST INSTRUCTION, THE CONTENTS
      ; OF ER1 ARE INVERTED AND PASSED THROUGH THE ALU AGAIN.
 01D6            NAC ,TRUE,ZLSB,CRTN & ALU ,,ABUS,NOP,EXNOR,EXNOR,DZ & REG ....R1
      ;
```



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

      ;(3) IF ZLSB WAS NOT SET BY THE PREVIOUS INSTRUCTION, JUMP TAKES PLACE TO
      ; ILLEGOP1.
 01D7            NAC ,PALSF,ZLSB,CJP & ALU & BADDRESS ILLEGOP1
      ;
      ;(4)  JUMP AND PSF TAKES PLACE TO HALT.
 01D8            NAC ,PALSF,IZERO,CJMP & ALU & BADDRESS HALT
      ;
      ;---------- -------- ---------- -- ---------- ---- ---- ---------- --
      ;
      ; ILLEGAL OPCODE ROUTINE ( NON 1750 INSTRUCTION )
      ; ***********************************************
      ;
      ; THE Q REGISTER IS LOADED SO THAT THERE IS ONE IN BIT POSITION 5 AND
      ; ZERO ELSEWHERE.
 01D9 ILLEGOP1:  NAC ,,CONT & ALU ,,IBUS,LPRS,,R,CR,DZ & IBUS DATA & CKALU
      &         DATINSRT B#04
      ;
      ;(2) JUMP TO ILLEGOP2 TAKES PLACE.
 01DA            NAC ,,JP & ALU & BADDRESS ILLEGOP2
      ;
      ;---------- -------- ---------- -- ---------- ---- ---- ---------- --
      ;
      ;---------- -------- ---------- -- ---------- ---- ---- ---------- --
      ;
      ; THIS IS THE END OF THE EMULATION SEQUENCES.
      ;
      ;
      END
```

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

      F: EQU H#FFFF
      ;-------------------------------------------------------------------------
      ; ADDRESSES OF SUBROUTINES AND OTHER ENTRY POINTS INTO 1ST I
      ; ************************************************************
FFFF ;
00A0 OFDI1: EQU H#A0
00A6 OFDI2: EQU H#A6
00AA OFDI3: EQU H#AA
00B0 OFBI1: EQU H#B0
00B4 OFBI1: EQU H#B4
00B8 OFII1: EQU H#B8
00BC OFII3: EQU H#BC
00C8 OFIM:  EQU H#C8
00C5 OFISP: EQU H#C5
00C9 OFISW: EQU H#C9
00CD OFSENPUR: EQU H#CD
00CF OSDI1: EQU H#CF
00D5 OSBI:  EQU H#D5
00D9 OSBI1: EQU H#D9
00DE OSII1: EQU H#DE
00E0 OSCENPUR: EQU H#E0
00E2 ASSIGN: EQU H#E2
00EE ASMULT: EQU H#EE
00F6 ASDLSTG2: EQU H#F6
00FC OVERFLOW: EQU H#FC
0100 FETDIADD: EQU H#100
     MIMICINT: EQU H#192
      ;-------------------------------------------------------------------------
0192 ;
```



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

      ORG H#400
      ;
      ;
      ;-------------------------------------------------------------------------
      ;
      ; SINGLE PRECISION INTEGER ADD
      ; ****************************
      ;
      ;(1) THIS IS THE ENTRY POINT FOR REGISTER TO REGISTER ADDRESS MODE.
      ;    THE CONTENTS OF THE SPECIFIED REGISTERS ARE ADDED.
      ;    THE MACHINE AND MICROSTATUS REGISTERS ARE LOADED DIRECTLY FROM
      ;    IZ,IC,IN,IP
      ;    CARRY IN TO THE ALU IS SET TO ZERO
0402 ;    CARRY TO THE APPROPRIATE STATUS BIT IS CARRY OUT OF THE ALU
0400 OPCODEA1:: NAC ,,,CONT
     / &        ALU B#1,B#0,,RAMF,ADD,ADD,AB
     / &        REG ,,GR1,GR2
     / &        STATSHFT B#0000000010000
     / &        CARRYSEL CCAR
     / &        DBUS SNONE
     / &        CEALU
      ;
      ;(2) A RETURN OCCURS CONDITIONAL ON OVERFLOW NOT BEING SET.
0401            NAC ,FALSE,OVR,CRTN
     / &        ALU
     / &        DBUS SNONE
      ;
      ;(3) UNCONDITIONAL JUMP TAKES PLACE TO OVERFLOW ROUTINE.
0402            NAC ,,,JP
     / &        ALU
     / &        DBUS SNONE
     / &        BADDRESS OVERFLOW
      ;
      ;
      ; (4) THIS IS THE ENTRY POINT FOR BASE RELATIVE ADDRESSING
      ; A JUMP IS MADE TO THE SUBROUTINE OFB1
0403 OPCODEI0:: NAC ,FALSE,TZERO,CJS
     / &        BADDRESS OFB1
     / &        ALU & DBUS SNONE
      ;
      ;(5) THE DERIVED OPERAND IS ADDED TO THE CONTENTS OF IR2
0404 SPIA1:     NAC ,,,CONT
     / &        ALU B#1,B#0,DBUS,RAMF,ADD,ADD,DA
     / &        REG B#10,B#10,RZERO,RZERO,R5
     / &        DBUS SREG
     / &        CARRYSEL CCAR
```

EM ES 403


```
AMD05/29 AMDASM MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750A - EMSL

      / &       STATSHFT M=0000000010000
      / &       CKALU
      :
      ;(6) CONDITIONAL RETURN TAKES PLACE ( ADDITIONAL ON  VERFLOW NOT BEING
      ; SET.
0405            NAC ,FALSE,,VR,CRTN
      / &       DBUS SNONE & ALU
      :
      ;(7) UNCONDITIONAL JUMP TAKES PLACE T  THE OVERFLOW ROUTINE
0406            NAC ,,,JP
      / &       BADDRESS OVERFLOW
      / &       DBUS SNONE & ALU
      :
      :
      :(8) THIS IS THE BASE RELATIVE INDIRECT ENTRY POINT.
      ;     JUMP TAKES PLACE TO THE SUBROUTINE BRI.
0407 IOPCD6#41 NAC ,FALSE,,ZERO,,CJS
      / &       BADDRESS ,BRI
      / &       ALU & DBUS SNONE
      :
      ;(9) JUMP TAKES PLACE TO SPIA1
0408            NAC ,,,JP
      / &       BADDRESS SPIA1
      / &       ALU & DBUS SNONE
      :
      :
      :(10) THIS IS THE ENTRY POINT FOR INTERMEDIATE COUNT POSITIVE ADDRESSING.
      ;     A JUMP SUB TAKES PLACE TO THE SUBROUTINE OPISP
0409 OPCODEA2:: NAC ,FALSE,ZERO,CJS
      / &       BADDRESS OPISP
      / &       ALU & DBUS SNONE
      :
      ;(11) THE CONTENTS OF THE DERIVED OPERAND ARE ADDED TO THE CONTENTS OF THE
      ;  SPECIFIED REGISTER
040A SPIA2:     NAC ,,,CONT
      / &       ALU B#1,B##0,DBUS,P   ',ADD,ADD,DA
      / &       REG ,,GR1,GR1,R5
      / &       STATSHFT M=0000#00010000
      / &       DBUS SR&Q
      / &       CARRYSEL CCAR
      / &       CKALU
      :
      ;(12) A RETURN OCCURS CONDITIONAL ON OVERFLOW NOT HAVING  ( SET.
040B            NAC ,FALSE,OVR,CRTN
```



```
AMD05/29 AMDASM MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750A - EMSL

      / &       IP S SNONE & ALU
      :
      ;(13) AN UNCONDITIONAL NAC JUMP T  THE  VERFLOW ROUTINE OCCURS.
040C            NAC ,,,JP
      / &       BADDRESS  VERFLOW
      / &       ALU & DBUS SNONE
      :
      :
      :(14) THIS I  THE ENTRY P INT F R IMMEDIATE AND DIRECT-INDEXED ADDRESSING.
      ;     A JUMP SUB TO MAKE  THE JUMP TAKE PLACE
040D OPCODEA0:: NAC ,FALSE,ZERO,CJS
      / &       BADDRESS ,IMM
      / &       ALU & DBUS SNONE
      :
      ;     A JUMP IS MADE T  SPIA1
040E            NAC ,,,JP
      / &       BADDRESS SPIA1
      / &       ALU & DBUS SNONE
      :
      ;(16) THIS IS THE ENTRY P INT FOR IMMEDIATE ADDRESSING.
      ;     A JUMP SUB TO MAKE  THE JUMP TAKE PLACE.
040F IOPCD6A1:: NAC ,FALSE,ZERO,CJS
      / &       BADDRESS ,IMM
      / &       ALU & DBUS SNONE
      :
      ;(15) A JUMP IS MADE TO SPIA1
0410            NAC ,,,JP
      / &       BADDRESS SPIA1
      / &       ALU & DBUS SNONE
      ;-------------------------------------------------------------------------
      :
      ; SINGLE PRECISION INTEGER SUBTRACT
      ; *********************************
      :
      :
      :
      :(1) THIS IS THE ENTRY P INT FOR REGISTER TO REGISTER ADDRESS MODE.
      ;     THE CONTENTS OF THE SPECIFIED REGISTERS ARE SUBTRACTED.
      ;     THE MACHINE AND MICROSTATUS REGISTERS ARE LOADED DIRECTLY FROM
      ;     IZ,IC,IN,IF
      ;     CARRY IN TO THE ALU IS SET TO ONE
      ;     CARRY TO THE APPROPRIATE STATUS BIT IS CARRY OUT OF THE ALU
0411 OPCODEP1:: NAC ,,,CONT
      / &       ALU B#1,B##0,,RAM0,SUBA,,UPDA,AR
```

```
AMD29116 AMDASM MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE CYCLE OPCODES FOR 1750A (EMSL

        / &     REG ,,OR1,OR1
          &     STAT,OFF B#01000000,0000
          &     CARRYSEL CLAR
          &     CK&ALU
          ;
          ;(1) A RETURN OCCURS, INDICATING THAT NO VIRTUAL OVERFLOW HAS OCCURRED.
0410              NAC ,FALSE,,RTN
          / &     DBUS SNONE & ALU
          ;
          ;(2) AN UNCONDITIONAL JUMP TO THE OVERFLOW ROUTINE OCCURS.
0411              NAC ,,JP
          / &     BADDRESS OVERFLOW
          / &     DBUS SNONE & ALU
          ;
          ;(3) THIS IS THE ENTRY POINT FOR DIRECT AND DIRECT-INDEXED ADDRESSING.
          ;     A JUMP SUB IS MADE TO THE SUBROUTINE OPEX1
0412 OPCODEB0:: NAC ,FALSE,TZYB0,CJS
          / &     BADDRESS OPDX1
          / &     ALU & DBUS SNONE
          ;
          ;   A JUMP IS MADE TO SPIS2.
```



```
AMD29116 AMDASM MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE CYCLE OPCODES FOR 1750A (EMSL

          ;   JUMP TAKEN PLACE IF OVERFLOW BIT IS SET.
0413 L...  NAC ,FALSE,,CJS
          / &     BADDRESS ...
          &     ALU & DBUS ...
          ;
          ;   JUMP TAKEN PLACE IF ...
0414 ...  NAC ,,,...
          / &     BADDRESS ...
          / &     ALU & DBUS ...
          ;
          ;
          ;   THIS IS THE ENTRY ... FOR IMMEDIATE SHORT POSITIVE ADDRESSING.
          ;     A JUMP SUB TAKES ... TO THE SUBROUTINE SPISP
0415 OPCODE8F:: NAC ,FALSE,...
          / &     BADDRESS ...
          / &     ALU & DBUS SNONE
          ;
          ;   THE CONTENT OF THE ... AND SUBTRACTED FROM ... CONTENTS OF THE
          ;   SPECIFIED REGISTER
0416 OPCODE1F:: NAC ,,,C,NZ
          / &     ALU B#1,B#A,B...,B#A,,OP...,CBR,DA
          / &     REG ,,OR1,...,...
          / &     STATSEL B#...
          / &     DBUS SNONE
          / &     CARRYSEL CLAR
          / &     CK&ALU
          ;
          ;(2) A RETURN OCCURS, INDICATING THAT NO VIRTUAL OVERFLOW HAS OCCURRED.
0417              NAC ,FALSE,,RTN
          / &     DBUS SNONE & ALU
          ;
          ;(3) AN UNCONDITIONAL JUMP TO THE OVERFLOW ROUTINE OCCURS.
0418              NAC ,,,JP
          / &     BADDRESS OVERFLOW
          / &     ALU & DBUS SNONE
          ;
          ;
          ;(14) THIS IS THE ENTRY POINT FOR DIRECT AND DIRECT-INDEXED ADDRESSING.
          ;     A JUMP SUB IS MADE TO THE SUBROUTINE OPEX1
0419 OPCODEB0:: NAC ,FALSE,TZYB0,CJS
          / &     BADDRESS OPDX1
          / &     ALU & DBUS SNONE
          ;
          ;   A JUMP IS MADE TO SPIS2.
```

```
AMDOS/29 AMDASM MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR STANDARD 16K28 DCB AT 2.0 17:0
```

```
0417            NAC ...,P
      >         BADDRESS ,DFISP
      &         ALU & DBUS SNONE

      ;(14  THIS IS THE ENTRY POINT FOR IMMEDIATE & DIR
      ;     A JUMP SUB IS MADE TO THE APPROPRIATE OPCM
0420 XOPCD4A2:: NAC ,FALSE,TZERO,JSB
      &         BADDRESS ,PIM
      &         ALU & DBUS SNONE

      ;(15  A JUMP IS MADE TO SPIS0
0421            NAC ...,P
      &         BADDRESS ,SPIS0
      &         ALU & DBUS SNONE

      ;-----------------------------------------------------------------
      ; SINGLE PRECISION INTEGER MULTIPLY
      ; *********************************


      ;(1) ENTRY POINT FOR REGISTER TO REGISTER ADDRESSING MODE
      ;     ***********
      ; THE DATA IN 2R2 IS TRANSFERRED TO BRD
      ;
0422 OPCODEC1:: NAC ...,P
      / &       ALU ...,NOP,2R,2R,2A
      / &       RFC ...,2R1,BD
      / &       CK4EJ
      / &       DBUS BACU
      / &       BADDRESS SPIM2
      ;
      ;(2) ENTRY POINT FOR ISP ADDRESSING
      ;     ***********
      ;     JUMP SUB TO ISP OPERAND FETCH ROUTINE
0423 OPCODEC2:: NAC ,FALSE,TZERO,JSB
      / &       BADDRESS ,PISP
      / &       ALU & DBUS SNONE
      ;
      ;(3) JUMP TO START OF MULTIPLY SEQUENCE
0424            NAC ...,P
      / &       BADDRESS SPIM2
      / &       ALU & DBUS SNONE
      ;
      ;
```



```
AMDOS/29 AMDASM MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR STANDARD 16K28 DCB SUBSET FOR 1750. EMSL
```

```
      ;
      ; (4) ENTRY POINT FOR INDIRECT ADDRESSING
      ; ***********
      ;
0425 OPCODEC3:: NAC ,P
      / &       BADDRESS ,PISP
      / &       ALU & DBUS SNONE
      ;
      ; (5)
0426            NAC ...,P
      / &       BADDRESS ,PIM2
      / &       ALU & DBUS SNONE
      ;
      ; ENTRY POINT FOR INDEXED AND INDIRECT INDEXED ADDRESSING
      ; ***********
      ;
      ; (6) JUMP TAKES PLACE TO THE APPROPRIATE OPERAND FETCH ROUTINE.
0427 OPCODEC0:: NAC ,FALSE,TZERO,...
      / &       BADDRESS ,PDEX
      / &       ALU & DBUS SNONE
      ;
      ; (7) JUMP TO THE START POINT OF MULTIPLY SEQUENCE TAKES PLACE.
0428            NAC ...,P
      / &       BADDRESS ,PIM2
      / &       ALU & DBUS SNONE
      ;
      ; ENTRY POINT FOR IMMEDIATE ADDRESSING
      ; ***********
      ;
      ;(8) JUMP SUB TAKES PLACE TO THE IMMEDIATE OPERAND FETCH ROUTINE.
0429 XOPCD4A4:  NAC ,FALSE,TZERO,...
      / &       BADDRESS ,PIM
      / &       ALU & DBUS SNONE
      ;
      ; START OF MULTIPLY ROUTINE
      ;
042A SPIM0:     NAC ,FALSE,TZERO,DSB
      / &       BADDRESS ,PDIM
      / &       ALU & DBUS SNONE
      ;
      ;
042B            NAC ,FALSE,TZERO,...
```

```
AMD08/29 AMDASM MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EMSL

      / &       BADDRESS ASMULT
      / &       ALU & DBUS SNONE
      ;
      ;
0420            NAC ,,,CONT
      / &       ALU ,,,NOP,OR,OR,ZA
      / &       REG ,,SR1,SR1
      / &       DBUS SNONE
      ;
      ;
      ;
      ;(24) THE MULTIPLY IS NOW COMPLETE.  TEST FOR OVERFLOW NOW TAKES PLACE.
      ;     JUMP TO OVERFLOW TAKES PLACE IF THE MS BITS IN SR1 IS NOT ZERO.
0422 SPIM4:     NAC ,FALSE,ZMSB,CJP
      / &       ADDRESS OVERFLOW
      / &       ALU ,,,NOP,OR,OR,ZA
      / &       REG ,,SR1,SR1
      / &       DBUS SNONE
      ;
      ;
      ;(25) JUMP TO OVERFLOW TAKES PLACE IF THE LS BITS OF SR1 IS NOT ZERO.
0424            NAC ,FALSE,ZLSB,CJP
      / &       ADDRESS OVERFLOW
      / &       ALU & DBUS SNONE
      ;
      ;(26) THE LS WORD OF THE PRODUCT IN THE QREG IS MASKED TO LEAVE ONLY THE MS
      ;   BIT.
0427            NAC ,,,CONT
      / &       ALU ,,DBUS,PAMP,AND,AND,DQ
      / &       DBUS SDATA
      / &       DATASHFT D#32768
      ;
      ;(27) JUMP TO OVERFLOW TAKES PLACE IF THE MS BITS IS NOT ZERO. ER5 IS PASSED
      ;  THROUGH THE ALU SO THAT THE TEST CONDITION MSB-8 WILL BE LOADED ACCORDING
      ;  TO THE VALUE OF THE MS BIT OF ER4. (THE REST OF ER5 IS ZERO).
0430            NAC ,FALSE,ZMSB,CJP
      / &       BADDRESS OVERFLOW
      / &       ALU ,,DBUS,NOP,OR,OR,D2
      / &       REG ,,,R8
      / &       DBUS SREG
      ;
      ;(28) JUMP TAKES PLACE TO SPIM5 IF THE MS BITS OF ER8 WAS ZERO.
0431            NAC ,TRUE,ZMSB,CJP
      / &       ALU & DBUS SNONE
      / &       BADDRESS SPIM5
```



```
AMD05/29 AMDASM MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EMSL

      ;
      ;(29) THE LS WORD OF THE QREG IS NEGATED AND LOADED INTO SR1.  THIS IS THE
      ; ACTION WHEN IT HAS BEEN DETERMINED THAT THE PRODUCT MUST BE NEGATIVE.
0432            NAC ,,,CONT
      / &       ALU SR1,PAMP,,PAMP,ZERO,,ZERO,DQ
      / &       REG ,,SR1,SR1
      &         CKALU
      / &       MACSTEN PASSAU
      / &       CARRYSEL ZERO
      &         DBUS SNONE
      &         STATSHFT PASSUP/ZERO/ZERO
      ;
      ;(30) RETURN TAKES PLACE
0433            NAC ,FALSE,ZERO,,RET
      &         ALU & DBUS SNONE
      ;
      ;(31) IN THE CASE WHEN THE SIGN OF THE PRODUCT HAS BEEN DETERMINED TO
      ;  BE POSITIVE, THE LS WORD OF THE PRODUCT AND ZERO IS LOADED INTO SR1. THE STATUS
      ;  REGISTER LOADED AT THIS INSTANT MAY OR MAY NOT BE CORRECT.
0434 SPIM5:     NAC ,FALSE,ZERO,,JFSN
      / &       ALU 8*1,PAMP,,PAMP,ZERO,,ZERO
      &         REG ,,SR1,SR1
      / &       CARRYSEL ZERO
      &         CKALU
      &         DBUS SNONE
      &         STATSHFT PASSUP/ZERO/ZERO
      ;
      ;
      ;
      ; SINGLE PRECISION INTEGER DIVIDE
      ; ******************************
      ;
      ;
      ; ENTRY POINT FOR REGISTER TO REGISTER ADDRESSING MODE
      ; ***********
      ;
      ; THE DATA IN SR2 IS TRANSFERRED TO ER5
0435 OPCODED1:: NAC ,,,JP
      / &       ALU ,,,NOP,OR,OR,ZA
      / &       REG ,,,GR2,R5
      / &       CKREG & DBUS SALU
      / &       BADDRESS SPIDP
      ;
      ; ENTRY POINT FOR ISP ADDRESSING MODE
      ; ***********
```

```
                                                                    PAGE  ..

                                                                    ;SD /MDASM MICRO AS..
                                                                    ;104 SEQUENCES FOR ..                    ..-..-1981  ..:..:..  ..EMUL

                                                                    ; START OF DIVIDE ..

                                                                    ;(10) A TEST IS MA..
                                                                    ;PID@:    NAC ,,..                                          ;YES
                                                                    ;         ALU ,,AP..
                                                                    ;&        REG ,,..
                                                                    ;&        CKALU & ..

                                                                    ;(11)JUMP TO SPIL..                                  ;DIVISOR IS NOT ..
                                                                    ;         NAC ,,AD..
                                                                    ;&        BADDR..
                                                                    ;&        ALU ,,AP..
                                                                    ;&        REG ,,..
                                                                    ;&        IPES SW..

                                                                    ;(12) JUMP TO OVER..                                 ;YES
                                                                    ;         NAC ,TP..
                                                                    ;&        BADDRES..
                                                                    ;&        ALU & DE..

                                                                    ;(13) A JUMP IS MA..
                                                                    ;PID1:    NAC ,PAL..                                 ;FIRST QUADRA T DIV
                                                                    ;&        PADDRESS A..
                                                                    ;&        ALU & TR..

                                                                    ;(14) THE DIVIDEND..                                 ;CLEARED
                                                                    ;         NAC ,,CL..
                                                                    ;&        ALU ,,RAMA..
                                                                    ;&        REG RWA1,,..
                                                                    ;&        CKALU & CK..
                                                                    ;&        DATINSRT ..

                                                                    ;(14A) THE DIVISOR IS ..
                                                                    ;         NAC ,,CONT..
                                                                    ;&        ALU ,,APL ,,RAMB..
                                                                    ;&        REG ,,CRL..
                                                                    ;&        CKALU & DRS..

                                                                    ;(15) THE CONSTANT ..
                                                                    ;     THE MICROPROGRAM ..                    ..-..-1981  ..:..:..  ..INTO THE STACK. THE COUNTER IS
                                                                    ;     NOT LOADED
                                                                          NAC ,TRUE,T ..
                                                                    ;&    ALU ,,EBUS..
                                                                    ;&    CKALU & SS..

AMDASM MICRO ASSEMBLER  V..-..  ..-..-1981  ..:..:..  PAGE  ..
IMPLEMENTATION SEQUENCES FOR ..                    ..-..-1981  ..:..:..  ..EMUL

                JUMP ..
0414            FIDE..          NA..
                &               PA..
                &               A..

0419            ;JMP ..         ..
                ;               ..
                &               ..
                &               ..
                ;               ..
                ;               ..
                ;               ..

041B            ;               ..
                &               ..
                &               ..
                &               ..

041D            ;               ..
                &               ..
                &               ..

                ;               ..
                ;               ..
                ;               ..
                ;               ..
                ;               ..
                ENTRY POINT FOR ..                                  ..ISD ..ITI)  ..
                ;****  ******

                ;JMP ..         ..
041A            ;               ..          NA..
                &               SS..
                &               A..

                ;JMP ..         ..
0419            ;               NA..
                &               BA..
                &               AL..

                ENTRY POINT FOR IMMEDIATE ADDR..  ..S MODE
                ;****  ******

                JUMP TO P..  PIM
0430 XOPCD4A0   ;               NA..  ..RA..S ..SH..
                &               BA..  P..M
                &               A..  ..BUS  ..N AE
```

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

      / &       DATINSRT D#1
      ;
      ;(15) QREG AND GR1 ARE UFSHIFTED, 0 GOES INTO THE MS BIT OF BOTH.
0444            NAC ,,,CONT
      / &       ALU B#0,B#1,,RAMQU,OR,OR,ZA
      / &       REG ,,GR1,GR1
      / &       CKALU
      / &       STATSHFT B#0001000010000
      ;
      ;(16) JUMP BACK TO (15) TAKES PLACE IF N IN THE MICROSTATUS REGISTER IS NOT
      ;  SET.
0445            NAC ,TRUE,MUX,LOOP
      / &       ALU & DBUS SNONE
      / &       STATSHFT B#0000000011110
      ;
      ;(16A) THE DIVISOR AND THE QUOTIENT MASK ARE DOWNSHIFTED TO COMPENSATE
      ;  FOR THE FACT THAT ONE EXTRA SHIFT HAS BEEN DONE.
0446            NAC ,,,CONT
      / &       ALU ,,,RAMQD,OR,OR,ZA
      / &       REG ,,GR1,GR1
      / &       CKALU & DBUS SNONE
      / &       STATSHFT B#0001000000000
      ;
      ;(16B) THE DIVISOR AND QUOTIENT MASK ARE DOWNSHIFTED AGAIN SO THAT
      ; THE FIRST NON ZERO BIT IS ALLIGNED DO BIT POS. 14.
0447            NAC ,,,CONT
      / &       ALU ,,,RAMQD,OR,OR,ZA
      / &       REG ,,GR1,GR1
      / &       CKALU & DBUS SNONE
      / &       STATSHFT B#0000110000000
      ;
      ;(17) THE DIVISOR IS SUBTRACTED FROM THE REMAINDER.
0448 SPID2:     NAC ,,,CONT
      / &       ALU ,,,RAMF,SUBR,SUBR,AB
      / &       REG B#01,,GR1,GR1
      / &       CKALU & DBUS SNONE
      / &       STATSHFT B#0100000000000
      ;
      ;(18) DBE SIGN BIT OF THE REMAINDER IS MASKED OFF.
0449 SPID3:     NAC ,,,CONT
      / &       ALU ,,DBUS,NOP,AND,AND,DA
      / &       REG B#01,B#01,GR1,GR1
      / &       DBUS SDATA & DATINSRT D#32768
      ;
      ;(19) JUP TAKES PLACE IF REMAINDER IS POSITIVE
044A            NAC ,TRUE,ZMSB,CJP
```



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

      / &       BADDRESS SPID4
      / &       ALU & DBUS SNONE
      ;
      ;(20) DIVISOR IS ADDED TO REMAINDER
044B            NAC ,,,CONT
      / &       ALU ,,,RAMF,ADD,ADD,AB
      / &       REG B#01,,GR1,GR1
      / &       CKALU & DBUS SNONE
      / &       STATSHFT B#0000000000000
      ;
      ;(21) THE DIVISOR AND THE QUOTIENT MASK ARE SHIFTED DOWN.
044C            NAC ,,,CONT
      / &       ALU ,,,RAMQD,OR,OR,ZA
      / &       REG ,,GR1,GR1
      / &       CKALU & DBUS SNONE
      / &       CARRYSEL CQLSB
      / &       STATSHFT B#0000000000000
      ;
      ;(21)JUMP TO SPID2 TAKES PLACE IF CARRY WAS NOT SET. IE BIT IN QUOTIENT MASK
      ; HAS NOT BEEN SHIFTED OUT OF GR1.
044D            NAC ,FALSE,CARR,CJP
      / &       BADDRESS SPID2
      / &       ALU ,,ABUS,NOP,OR,OR,DZ
      / &       REG ,,,R8
      / &       DBUS SNONE
      ;
      ;(22) JUMP TO SPID5 TAKES PLACE IF THE MS BIT WAS SET INDICATING A NEGATIVE
      ; RESULT.
044E            NAC ,FALSE,ZMSB,CJP & ALU & DBUS SNONE & BADDRESS SPID5
      ;
      ;(22) THE QUOTIENT IN R9 IS LOADED INTO GR1. RETURN TAKES PLACE IF THE
      ; QUOTIENT IS TO BE POSITIVE.
044F            NAC ,FALSE,TZERO,CRTN
      / &       ALU B#1,R#0,ABUS,RAMF,OR,OR,DZ
      / &       REG ,,GR1,GR1,R9
      / &       CKALU & DBUS SNONE
      / &       STATSHFT B#0000000010000
      / &       CARRYSEL CZERO
      ;
      ;(23) A ONE IS PLACED IN THE APPROPRIATE POSITION IN THE QUOTIENT USING THE
      ; QUOTIENT MASK IN QREG.
0450 SPID4:     NAC ,,,CONT
      / &       ALU ,,ABUS,NOP,OR,OR,DQ
      / &       REG ,,,R9
      / &       CKREG & DBUS SALU
      ;
```

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

      ;(24) THE DIVISOR AND THE QUOTIENT MASK ARE SHIFTED DOWN.
 0451         NAC ,,,CONT
      / &     ALU ,,,RAMQD,OR,OR,ZA
      / &     REG ,,GR1,GR1
      / &     CKALU & DBUS SNONE
      / &     CARRYSEL CQLSB
      / &     STATSHFT B"0000000000000
      ;
      ;(25) JUMP TO SPID2 TAKES PLACE IF THE QUOTIENT
      ;  MASK HAS NOT BEEN SHIFTED OUT OF THE GR1.
 0452         NAC ,FALSE,CARR,CJP
      / &     BADDRESS SPID2
      / &     ALU ,,ABUS,NOP,OR,OR,DZ
      / &     REG ,,,R8
      / &     DBUS SNONE
      ;
      ;(25A) JUMP TO SPID5 TAKES PLACE IF THE MS BIT WAS SET INDICATING A NEGATIVE
      ; RESULT.
 0453         NAC ,FALSE,ZMSB,CJP
      / &     ALU & DBUS SNONE & BADDRESS SPID5
      ;
      ;
      ;(26) THE QUOTIENT IS PLACED IN GR1, THE MACHINE STATUS REGISTER IS
      ; LOADED AND &TURN TAKES PLACE.
 0454         NAC ,FALSE,TZERO,CRTN
      / &     ALU B"1,B"0,ABUS,RAMF,OR,OR,DZ
      / &     REG ,,GR1,GR1,R9
      / &     CARRYSEL CZERO
      / &     CKALU & DBUS SNONE
      / &     STATSHFT B"0000000010000
      ;
      ;(26A) THE REMAINDER IN GR1 + 1 IS NEGATED.
 0455 SPID5:  NAC ,,,CONT
      / &     ALU ,,,RAMF,SUBS,SUBS,ZA
      / &     REG B"01,B"01,GR1,GR1
      / &     MACSTEN B"0000
      / &     CARRYSEL CZERO
      / &     CKALU & DBUS SNONE
      / &     STATSHFT B"0100000000000
      ;
      ;
      ;(26B) THE QUOTIENT IS NEGATED AND RETURN TAKES PLACE.
 0456         NAC ,FALSE,TZERO,CRTN
      / &     ALU B"1,B"0,ABUS,RAMF,SUBR,SUBR,DZ
      / &       CKALU & DBUS SNONE
```



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

      / &     REG ,,GR1,GR1,R9
      / &     STATSHFT B"0100000010000
      / &     CARRYSEL 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 OPCODEF1:: NAC ,FALSE,TZERO,CRTN
      / &     ALU ,B"0,,NOP,SUBS,SUBS,AB
      / &     REG ,,GR2,GR1
      / &     DBUS SNONE
      / &     STATSHFT B"0100000010000
      / &     CARRYSEL CZERO
      ;
      ; ENTRY POINT FOR BASE RELATIVE ADDRESSING
      ; ***********
      ;
      ;
 0458 OPCODE32:: NAC ,FALSE,TZERO,CJS
      / &     ALU & DBUS SNONE
      / &     BADDRESS OFB1
      ;
 0459         NAC ,FALSE,TZERO,CRTN
      / &     ALU ,B"0,ABUS,NOP,SUBR,SUBR,DA
      / &     REG ,B"10,,RZERO,R5
      / &     DBUS SNONE
      / &     CARRYSEL CZERO & STATSHFT B"0100000010000
      ;
      ; ENTRY POINT FOR BASE RELATIVE INDEXED ADDRESSING
      ; ***********
      ;
      ; JUMP SUB TO OPERAND FETCH ROUTINE
 045A IOPCD40C:  NAC ,FALSE,TZERO,CJS
      / &     ALU & DBUS SNONE & BADDRESS OFBX1
```

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

      ;
      ; DERIVED OP IN RR5 IS SUBTRACTED FROM CONTENTS OF IR2
 0458           MAC ,FALSE,TZERO,CRTN
      / &       ALU ,B#0,ABUS,NOP,SUBR,SUBR,DA
      / &       REG ,B#10,,RZERO,R5
      / &       DBUS SNONE & CARRYSEL CZERO
      / &       STATSHFT B#0100000010000
      ;
      ; ENTRY POINT FOR ISP ADDRESSING
      ; ++++++++++++
 045C OPCODEF2:: MAC ,FALSE,TZERO,CJS
      / &       ALU & DBUS SNONE & BADDRESS OFISP
      ;
      ;
 045D           MAC ,FALSE,TZERO,CRTN
      / &       ALU ,B#0,ABUS,NOP,SUBR,SUBR,DA
      / &       REG ,,,GR1,R5
      / &       DBUS SNONE & CARRYSEL CZERO & STATSHFT B#0130000010000
      ;
      ; ENTRY POINT FOR ISN ADDRESSING
      ; ++++++++++++
 045E OPCODEF3:: MAC ,FALSE,TZERO,CJS
      / &       ALU & DBUS SNONE & BADDRESS OFISN
      ;
      ;
 045F           MAC ,FALSE,TZERO,CRTN
      / &       ALU ,B#0,ABUS,NOP,SUBR,SUBR,DA
      / &       REG ,,,GR1,R5 & DBUS SNONE & CARRYSEL CZERO
      / &       STATSHFT B#0100000010000
      ;
      ;
      ; ENTRY POINT FOR DIRECT OR DIRECT INDEXED ADDRESSING
      ; ++++++++++++
      ;
 0460 OPCODEF0:: MAC ,FALSE,TZERO,CJS
      / &       ALU & DBUS SNONE & BADDRESS OFDI1
      ;
      ;
 0461           MAC ,FALSE,TZERO,CRTN
      / &       ALU ,B#0,ABUS,NOP,SUBR,SUBR,DA
      / &       REG ,,,GR1,R5 & DBUS SNONE & CARRYSEL CZERO
      / &       STATSHFT B#0100000010000
      ;
      ;
      ; ENTRY POINT FOR IMMEDIATE ADDRESSING
      ; ++++++++++++
```



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

 0462 IOPCD4AA:  MAC ,FALSE,TZERO,CJS
      / &       ALU & DBUS SNONE & BADDRESS OFIM
      ;
      ;
 0463           MAC ,FALSE,TZERO,CRTN
      / &       ALU ,B#0,ABUS,NOP,SUBR,SUBR,DA
      / &       REG ,,,GR1,R5 & DBUS SNONE & CARRYSEL CZERO
      / &       STATSHFT B#0100000010000
      ;
      ;-----------------------------------------------------------------
      ;
      ; COMPARE BETWEEN LIMITS
      ; **********************
      ;
      ; ENTRY POINT FOR DIRECT OR DIRECT INDEXED ADDRESSING
      ; ++++++++++++
      ;
 0464 OPCODEF4:: MAC ,FALSE,TZERO,CJS
      / &       DBUS SNONE & ALU & BADDRESS OFDI2
      ;
      ; THE LOWER LIMIT IS LOADED INTO QREG. THE MACHINE STATUS REGISTER IS CLEARED.
 0465           MAC ,,,CONT & ALU ,B#0,ABUS,QREG,ON,OR,DZ & REG ,,,,R5 & DBUS SNONE
      / &       CKALU & STATSHFT B#0000000000011 & MACSTEN B#0000
      ;
      ; THE LOWER LIMIT IN Q IS SUBTRACTED FROM THE UPPER LIMIT. ALL OF THE MICROSTATUS
      ; REGISTER BITS ARE LOADED.  ONLY THE CARRY BIT OF THE MACHINE
      ; STATUS REGISTER IS LOADED.
 0466           MAC ,,,CONT & ALU B#0,B#0,ABUS,NOP,SUBS,SUBS,DQ & REG ,,,,R6
      / &       DBUS SNONE & STATSHFT 0100000010000 & MACSTEN B#1101
      / &       CARRYSEL CONE
      ;
      ; RETURN TAKES PLACE IF MICRO STATUS BIT N WAS SET.
 0467           MAC ,TRUE,MUX,CRTN & ALU & DBUS SNONE & STATSHFT B#0000000011110
      ;
      ;THE LOWER LIMIT IS SUBTRACTED FROM THE CONTENTS OF GR1
      ; THE MACHINE STATUS IS SET ACCORDING TO THE RESULT. CARRY IS SET TO ZERO.
 0468           MAC ,,,CONT
      / &       ALU ,B#0,ABUS,NOP,SUBR,SUBR,DA
      / &       REG ,,,GR1,R5 & DBUS SNONE & CARRYSEL CZERO
      / &       STATSHFT B#0100000101110
      ;
      ; RETURN TAKES PLACE IF THE RESULT WAS NEGATIVE INDICATING THAT THE CONTENTS
      ; OF GR1 WERE LESS THAN THE LOWER LIMIT.
 0469           MAC ,TRUE,MUX,CRTN
      / &       ALU & DBUS SNONE & STATSHFT B#0000000101110
```

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

      ;
      ; THE UPPER LIMIT IS SUBTRACTED FROM THE CONTENTS OF GR1.  THE MACHINE STATUS
      ; IS SET ON THE RESULT OF THE SUBTRACTION.  CARRY IS SET EQUAL TO ZERO.
0468          NAC ,,,CONT
      / &     ALU ,B#0,ABUS,NOP,SUBR,SUBR,DA
      / &     REG ,,,GR1,R6 & DBUS SNONE & CARRYSEL CZERO
      / &     STATSHFT B#0100000010000
      ;
      ; RETURN TAKES PLACE IF THE RESULT OF THE SUBTRACTION WAS POSITIVE INDICATING
      ; THAT THE CONTENTS OF GR1 WERE GREATER THAN THE UPPER LIMIT.
0469          NAC ,TRUE,MUX,CRTN
      / &     ALU & DBUS SNONE & STATSHFT B#0000000100110
      ;
      ; THE CONTENTS OF GR1 ARE BETWEEN THE LIMITS SO THE MACHINE STATUS IS SET
      ; FOR A ZERO RESULT.
046C          NAC ,FALSE,TZERO,CRTN
      / &     ALU ,B#0,,NOP,AND,AND,DZ
      / &     DBUS SNONE & CARRYSEL CZERO
      / &     STATSHFT B#0000000010000
      ;
      ;
      ;-----------------------------------------------------------------------
      ;
      ;
      ;
      ;-----------------------------------------------------------------------
      ;
      ; INCREMENT MEMORY BY POSITIVE INTEGER
      ; ************************************
      ;
      ; ENTRY POINT
      ; ***********
      ;
      ;(1) JUMP SUB TAKES PLACE TO OPDX1
046D OPCODEA3:: NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS OPDX1
      ;
      ; THE INSTRUCTION IN ER1 IS SWOPPED WITH THE CONTENTS OF IR0.
046E          NAC ,,,CONT & ALU ,,ABUS,RAMA,OR,OR,DZ & REG ,,RZERO,RZERO,R1
      / &     CKALU & CKREG & DBUS SALU
      ;
      ;(3) THE INSTRUCTION IN IR0 IS DOWNSHIFTED.
046F          NAC ,,,CONT & ALU ,,,RAMD,OR,OR,ZA & REG ,,RZERO,RZERO & CKALU
      / &     DBUS SNONE & STATSHFT B#0000000000000
      ;
      ;(4) THE INSTRUCTION IN IR0 IS DOWNSHIFTED.
```



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

0470          NAC ,,,CONT & ALU ,,,RAMD,OR,OR,ZA & REG ,,RZERO,RZERO & CKALU
      / &     DBUS SNONE & STATSHFT B#0000000000000
      ;
      ;(5) THE INSTRUCTION IN IR0 IS MASKED BY 0000000000111111 AND DOWNSHIFTED.
0471          NAC ,,,CONT & ALU ,,DBUS,RAMD,AND,AND,DA & REG ,,RZERO,RZERO
      / &     DBUS SDATA & DATINSRT D#63 & CKALU
      ;
      ;(6) 2 IS ADDED TO THE CONTENTS OF IR0 AND IT IS DOWNSHIFTED.
0472          NAC ,,,CONT & ALU ,,DBUS,RAMD,ADD,ADD,DA & REG ,,RZERO,RZERO & CKALU
      / &     DBUS SDATA & DATINSRT D#2
      ;
      ;(7) THE POSITIVE INTEGER NOW OBTAINED IS ADDED TO THE CONTENTS OF ER5.
0473          NAC ,,,CONT & ALU ,B#0,ABUS,NOP,ADD,ADD,DA & DBUS SALU & CKREG
      / &     REG ,,,RZERO,R5 & STATSHFT B#0000000010000 & CARRYSEL CCAR
      / &     MACSTEN B#0000
      ;
      ;(3) JUMP TO OSGENPUR TAKES PLACE IF OVERFLOW WAS NOT SET BY THE PREVIOUS
      ; INSTRUCTION.
      ; THE CONTENTS OF ER1 ARE PUT BACK IN IR0
0474          NAC ,FALSE,OVR,CJP & ALU ,,ABUS,RAMF,OR,OR,DZ & REG ,,RZERO,,R1
      / &     DBUS SNONE & CKALU & BADDRESS OSGENPUR
      ;
      ;(4) JUMP SUB TO OSGENPUR TAKES PLACE.
0475          NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS OSGENPUR
      ;
      ;(5) JUMP TO OVERFLOW TAKES PLACE.
0476          NAC ,,,JP & ALU & DBUS SNONE & BADDRESS OVERFLOW
      ;
      ;-----------------------------------------------------------------------
      ;
      ;
      ; DECREMENT MEMORY BY POSITIVE INTEGER
      ; ************************************
      ;
      ; DECREMENT MEMORY BY POSITIVE INTEGER
      ; ************************************
      ;
      ; ENTRY POINT
      ; ***********
      ;
      ;(1) JUMP SUB TAKES PLACE TO OPDX1
0477 OPCODEB3:: NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS OPDX1
      ;
      ; THE INSTRUCTION IN ER1 IS SWOPPED WITH THE CONTENTS OF IR0.
0478          NAC ,,,CONT & ALU ,,ABUS,RAMA,OR,OR,DZ & REG ,,RZERO,RZERO,R1
      / &     CKALU & CKREG & DBUS SALU
```

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

        ;
        ;(3) THE INSTRUCTION IN IR0 IS DOWNSHIFTED.
0479            NAC ,,,CONT & ALU ,,,RAMD,OR,OR,ZA & REG ,,RZERO,RZERO & CKALU
        / &     DBUS SNONE & STATSHFT B#0000000000000
        ;
        ;(4) THE INSTRUCTION IN IR0 IS DOWNSHIFTED.
047A            NAC ,,,CONT & ALU ,,,RAMD,OR,OR,ZA & REG ,,RZERO,RZERO & CKALU
        / &     DBUS SNONE & STATSHFT B#0000000000000
        ;
        ;(5) THE INSTRUCTION IN IR0 IS MASKED BY 0000000000111111 AND DOWNSHIFTED
047B            NAC ,,,CONT & ALU ,,DBUS,RAMD,AND,AND,DA & REG ,,RZERO,RZERO
        / &     DBUS SDATA & DATINSRT D#63 & CKALU
        ;
        ;(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            NAC ,,,CONT & ALU ,,DBUS,RAMD,ADD,ADD,DA & REG ,,RZERO,RZERO
        / &     DBUS SDATA & DATINSRT D#2 & CKALU
        ;
        ;(7) THE POSITIVE INTEGER NOW OBTAINED IS SUBTRACTED FROM THE CONTENTS OF ER5.
047D            NAC ,,,CONT & ALU ,B#2,ABUS,NOP,SUBS,SUBS,DA & DBUS SALU & CKALU
        / &     REG ,,,RZERO,R5 & STATSHFT B#0100020010000 & CARRYSEL CCAR
        / &     MACSTEN B#0000
        ;
        ;(8) JUMP TO OSGENPUR TAKES PLACE IF OVERFLOW WAS NOT SET BY THE PREVIOUS
        ; INSTRUCTION.
        ; THE CONTENTS OF ER1 ARE PUT BACK IN IR0
047E            NAC ,FALSE,OVR,CJP & ALU ,,ABUS,RAMF,OR,OR,DZ & REG ,,RZERO,,R1
        / &     DBUS SNONE & CKALU & PADDRESS OSGENPUR
        ;
        ;(4) JUMP SUB TO OSGENPUR.
047F            NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & PADDRESS OSGENPUR
        ;
        ;(5) JUMP TO OVERFLOW TAKES PLACE.
0480            NAC ,,,JP & ALU & DBUS SNONE & BADDRESS OVERFLOW
        ;
        ;-------------------------------------------------------------------------
        ;
        ; SINGLE PRECISION ABSOLUTE VALUE OF REGISTER
        ; *******************************************
        ;
        ;(1) THE CONTENTS OF GR2 (RB) ARE PLACED IN GR1 (RA), THE MACHINE STATUS IS
        ; LOADED.
0481 OPCODEA4:: NAC ,,,CONT & ALU ,B#0,,RAMF,OR,OR,ZA & REG ,,GR1,GR2
        / &     CKALU & DBUS SNONE & STATSHFT B#0000000010000 & CARRYSEL CZERO
        / &     MACSTEN B#0000
        ;
```



```
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            NAC ,FALSE,MUX,CRTN & ALU & DBUS SNONE & STATSHFT B#0000000101110
        ;
        ;(3) THE CONTENTS OF GR2 ((RB) ARE NEGATED AND PLACED IN GR1 (RA).
0483            NAC ,FALSE,TZERO,CRTN & ALU ,B#0,,RAMF,SUBS,SUBS,ZA & CKALU
        / &     DBUS SNONE & STATSHFT B#0100000010000 & CARRYSEL CZERO
        / &     REG ,,GR1,GR2
        ;
        ;-------------------------------------------------------------------------
        ;
        ; SINGLE PRECISION NEGATE REGISTER
        ; ********************************
        ;
        ; ENTRY POINT
        ; ***********
        ;
        ;(1) THE CONTENTS OF GR2 (RB) ARE NEGATED AND PLACED IN GR1 (RA),
        ;  THE MACHINE STATUS IS SET ACCORDINGLY AND RETURN TAKES PLACE.
0484 OPCODEB4:: NAC ,,,CONT & ALU ,B#0,,RAMF,SUBS,SUBS,ZA & CKALU
        / &     REG ,,GR1,GR2 & DBUS SNONE & CARRYSEL CZERO
        / &     STATSHFT B#0100000010000
        ;
        ;(2) RETURN OCCURS IF OVERFLOW HAS NO BEEN SET.
0485            NAC ,FALSE,OVR,CRTN & ALU & DBUS SNONE
        ;
        ;(3) JUMP TO OVERFLOW
0486            NAC ,,,JP & ALU & DBUS SNONE & BADDRESS OVERFLOW
        ;-------------------------------------------------------------------------
        ;
        ;
        ;-------------------------------------------------------------------------
        ;
        ; INCLUSIVE LOGICAL OR
        ; ********************
        ;
        ; ENTRY POINT FOR REGISTER ADDRESSING
        ; ***********
        ;
        ;(1) THE CONTENTS OF GR2 ARE ORED WITH THE CONTENTS OF GR1 AND THE RESULT PLACED
        ; IN GR1.  THE MACHINE STATUS REGISTER IS LOADED ACCORDING TO THE RESULT.
0487 OPCODEE1:: NAC ,FALSE,TZERO,CRTN & ALU ,B#0,,RAMF,OR,OR,AB & REG ,,GR1,GR2
        / &     CKALU & DBUS SNONE & STATSHFT B#0000000010000
        / &     CARRYSEL CZERO
        ;
        ; ENTRY POINT FOR BASE RELATIVE ADDRESSING
        ; ***********
```

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

      ;
      ;(1) JUMP SUB TO OFB1
0488 OPCODE30:: NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS OFB1
      ;
      ;(2) THE DERIVED OPERAND IS ORED WITH THE CONTENTS OF IR2 AND THE RESULT
      ; PLACED IN IR2.
0489            NAC ,FALSE,TZERO,CRTN & ALU ,B#0,ABUS,RAMF,OR,OR,DA
     / &        REG B#10,B#10,RZERO,RZERO,R5 & DBUS SNONE & STATSHFT B#0000000010000
     / &        CKALU & CARRYSEL CZERO
      ;
      ; ENTRY POINT FOR BASE RELATIVE INDEXED ADDRESSING
      ; ************
      ;
      ;(1) JUMP SUB TO OFBI1
048A XOPCD40F:  NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS OFBI1
      ;
      ;(2) THE DERIVED OPERAND IS ORED WITH THE CONTENTS OF IR2 AND THE RESULT
      ;  PLACED IN IR2.
048B            NAC ,FALSE,TZERO,CRTN & ALU ,B#0,ABUS,RAMF,OR,OR,DA
     / &        REG B#10,B#10,RZERO,RZERO,R5 & DBUS SNONE & STATSHFT B#0000000010000
     / &        CKALU & CARRYSEL CZERO
      ;
      ; ENTRY POINT FOR DIRECT OR DIRECT INDEXED ADDRESSING
      ; ************
      ;
      ;(1) JUMP SUB TO OFDI1
048C OPCODE40:: NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS OFDI1
      ;
      ;(2) THE DERIVED OPERAND IS ORED WITH THE CONTENTS OF GR1 AND THE RESULT PLACED
      ; IN GR1.
048D            NAC ,FALSE,TZERO,CRTN & ALU ,B#0,ABUS,RAMF,OR,OR,DA
     / &        REG ,,GR1,GR1,R5 & DBUS SNONE & STATSHFT B#0000000010000 & CKALU
     / &        CARRYSEL CZERO
      ;
      ; ENTRY POINT FOR IMMEDIATE ADDRESSING
      ; ************
      ;
      ;(1) JUMP SUB TO OFIM
048E XOPCD40:  NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS OFIM
      ;
      ;(2) THE DERIVED OPERAND IS ORED WITH THE CONTENTS OF GR1 AND THE RESULT LEFT
      ;  IN GR1.
048F            NAC ,FALSE,TZERO,CRTN & ALU ,B#0,ABUS,RAMF,OR,OR,DA
     / &        REG ,,GR1,GR1,R5 & DBUS SNONE & STATSHFT B#0000000010000 & CKALU
     / &        CARRYSEL CZERO
      ;
```



```
AMDOS/29 AMDASM 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 (RB) ARE ANDED WITH THE CONTENTS OF GR1 (RA).
      ; THE RESULT IS STORED IN GR1 (RA).
0490 OPCODE33:: NAC ,FALSE,TZERO,CRTN & ALU ,B#0,,RAMF,AND,AND,AB & REG ,,GR1,GR2
     / &        CKALU & DBUS SNONE & STATSHFT B#0000000010000
     / &        CARRYSEL CZERO
      ;
      ; ENTRY POINT FOR BASE RELATIVE ADDRESSING
      ; ************
      ;
      ;(1) JUMP SUB TO OFB1
0491 OPCODE31:: NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS OFB1
      ;
      ;(2) THE DERIVED OPERAND IS ANDED WITH THE CONTENTS OF IR2 AND THE RESULT
      ; LEFT IN IR2.
0492            NAC ,FALSE,TZERO,CRTN & ALU ,B#0,ABUS,RAMF,AND,AND,DA
     / &        REG B#10,B#10,RZERO,RZERO,R5 & CKALU & DBUS SNONE
     / &        STATSHFT B#0000000010000
     / &        CARRYSEL CZERO
      ;
      ; ENTRY POINT FOR BASE RELATIVE INDEXED ADDRESSING.
      ; ************
      ;
      ;(1) JUMP SUB TO OFBI1
0493 XOPCD40E:  NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS OFBI1
      ;
      ;(2) THE DERIVED OPERAND IS ANDED WITH THE CONTENTS OF IR2 AND THE RESULT LEFT
      ;  IN IR2.
0494            NAC ,FALSE,TZERO,CRTN & ALU ,B#0,ABUS,RAMF,AND,AND,DA
     / &        REG B#10,B#10,RZERO,RZERO,R5 & CKALU & DBUS SNONE
     / &        STATSHFT B#0000000010000
     / &        CARRYSEL CZERO
      ;
      ; ENTRY POINT FOR DIRECT AND DIRECT INDEXED ADDRESSING
      ; ************
      ;
      ;(1) JUMP SUB TO OFDI1
0495 OPCODE32:: NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS OFDI1
```

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

      ;
      ;(2) THE DERIVED OPERAND IS ANDED WITH THE CONTENTS OF GR1 AND THE RESULT
      ; PLACED IN GR1.
0496            NAC ,FALSE,TZERO,CRTN & ALU ,B#0,ABUS,RAMF,AND,AND,DA
      / &       REG ,,GR1,GR1,R5 & CKALU & DBUS SNONE & STATSHFT B#00000010000
      / &       CARRYSEL CZERO
      ;
      ; ENTRY POINT FOR IMMEDIATE ADDRESSING
      ; ************
      ;
0497 XOPCD4A7:  NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS OPIM
      ;
      ;(2) THE DERIVED OPERAND IS ANDED WITH THE CONTENTS OF GR1 (RA) AND THE RESULT
      ; LEFT IN GR1 (RA).
0498            NAC ,FALSE,TZERO,CRTN & ALU ,B#0,ABUS,RAMF,AND,AND,DA
      / &       REG ,,GR1,GR1,R5 & CKALU & DBUS SNONE & STATSHFT B#00000010000
      / &       CARRYSEL CZERO
      ;
      ;-------------------------------------------------------------------------
      ;
      ; EXCLUSIVE LOGICAL OR
      ; ********************
      ;
      ; ENTRY POINT FOR REGISTER ADDRESSING
      ; ************
      ;
      ;(1) THE CONTENTS OF GR2 (RB) IS EXCLUSIVE ORED WITH THE CONTENTS OF GR1 (RB).
      ;  THE RESULT IS LEFT IN GR1 (RA).
0499 OPCODEE5:: NAC ,FALSE,TZERO,CRTN & ALU ,B#0,,RAMF,EXOR,EXOR,AB
      / &       REG ,,GR1,GR2 & CKALU & DBUS SNONE & STATSHFT B#00000010000
      / &       CARRYSEL CZERO
      ;
      ; ENTRY POINT FOR DIRECT OR DIRECT INDEXED ADDRESSING
      ; ************
      ;
      ;(1) JUMP SUB TO OPDX1
049A OPCODEE4:: NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS OPDX1
      ;
      ;(2) THE DERIVED OPERAND IS EXCLUSIVE ORED WITH THE CONTENTS OF GR1 (RA) AND
      ; THE RESULT LEFT IN GR1 (RA).
049B            NAC ,FALSE,TZERO,CRTN & ALU ,B#0,ABUS,RAMF,EXOR,EXOR,DA
      / &       REG ,,GR1,GR1,R5 & CKALU & DBUS SNONE & STATSHFT B#00000010000
      / &       CARRYSEL CZERO
      ;
      ; ENTRY POINT FOR IMMEDIATE ADDRESSING
      ; ************
```



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

      ;
      ;(1) JUMP SUB TO OPIM
049C XOPCD4A9:  NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS OPIM
      ;
      ;(2) THE DERIVED OPERAND IS EXCRED WITH THE CONTENTS OF GR1 (RA) AND THE RESULT
      ;  LEFT IN GR1 (RA).
049D            NAC ,FALSE,TZERO,CRTN & ALU ,B#0,ABUS,RAMF,EXOR,EXOR,DA
      / &       REG ,,GR1,GR1,R5 & CKALU & DBUS SNONE & STATSHFT B#00000010000
      / &       CARRYSEL CZERO
      ;
      ;-------------------------------------------------------------------------
      ;
      ; LOGICAL NAND
      ; ************
      ;
      ; ENTRY POINT FOR REGISTER ADDRESSING
      ; ************
      ;
      ;(1) THE CONTTENTS OF GR2 (RB) ARE ANDED WITH THE CONTENTS OF GR1 (RA) AND
      ; THE RESULT LEFT IN GR1 (RA).
049E OPCODEE7:: NAC ,,,CONT & ALU ,,,RAMF,AND,AND,AB & REG ,,GR1,GR2 & CKALU
      / &       DBUS SNONE
      ;
      ;(2) THE CONTTENTS OF GR1 (RA) ARE INVERTED (BY TAKINGG THE EXCLUSIVE NOR WITH
      ;  ZERO) AND PLACED IN GR1 (RA).
04AF            NAC ,FALSE,TZERO,CRTN & ALU ,B#0,,RAMF,EXNOR,EXNOR,ZA
      / &       REG ,,GR1,GR1 & CKALU & DBUS SNONE & STATSHFT B#00000010000
      / &       CARRYSEL CZERO
      ;
      ; ENTRY POINT FOR DIRECT OR DIRECT INDEXED ADDRESSING
      ; ************
      ;
      ;(1) JUMP SUB TO OPDX1
04A0 OPCODEE6:: NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS OPDX1
      ;
      ;(2) THE DERIVED OPERAND IS ANDED WITH THE CONTENTS OF GR1 (RA) AND THE RESULT
      ; LEFT IN GR1 (RA).
04A1            NAC ,,,CONT & ALU ,,ABUS,RAMF,AND,AND,DA & REG ,,GR1,GR1,R5 & CKALU
      / &       DBUS SNONE
      ;
      ;(3) THE NOT OF THE CONTENTS OF GR1 (RA) IS TAKEN AND THE RESULT PLACED IN
      ; GR1 (RA).
04A2            NAC ,FALSE,TZERO,CRTN & ALU ,B#0,,RAMF,EXNOR,EXNOR,ZA
      / &       REG ,,GR1,GR1 & CKALU & DBUS SNONE & STATSHFT B#00000010000
      ;
      ; ENTRY POINT FOR IMMEDIATE ADDRESSING
```

```
AMD29/29 AMDASM MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750A. (EMSL

        ; **********
        ;
        ;(1) JUMP FOR TO OPIM
04A7 EOPCODE59: NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS OPIM
        ;
        ;(2) THE DESTINATION OPERAND IS ANDED WITH THE CONTENTS OF RA AND THE RESULT
        ; PLACED IN THE RA.
04A8            NA  ,...CONT
        / &     ALU ,,RAMD,AND,RAMA,ZA & REG ,,RZERO,RZERO & DBUS SNONE
        ;
        ;(3) THE CONTENTS OF IR0 ARE DOWNSHIFTED.
        ; THE RESULT
04A9            NAC ,FALSE,TZERO,CJPL & ALU ,ENT,,,DBUS,ZA & REG ,,RZERO,R0 & DBUS SALU
        / &     SR1 ,SR1,SR1 & STATSHFT B"0000000000000
        / &     CKALU & CKREG
        ;
        ;---------- ---------- ---------- ---------- ---------- ----------
        ;
        ;----------
        ;
        ;SHFTNM -- SUBROUTINE THAT DERIVES THE REQUIRED NUMBER OF SHIFT PLACES
        ; FROM THE SHI FIELD IN THE INSTRUCTION WORD.  THIS ROUTINE IS USED BY THE
        ; INSTRUCTIONS SLL,SRL,SRA,SLC,LSLL,DSRL,DSRA,CLEC.
        ;
04AA SHFTNM:    NAC ,,,CONT & ALU ,,DBUS,QREG,OR,OR,DZ & DBUS SDATA & DATAINSRT B"48
        / &     CKALU
        ;
        ;(2) THE INSTRUCTION FROM ER1 IS MASKED AND LOADED INTO IR0. THE CONTENTS
        ; OF IR2 ARE TEMPORARILY PLACED IN ER1.
04AB            NAC ,,,CONT
        / &     ALU ,,ABUS,RAMA,AND,AND,DQ
        / &     REG ,,RZERO,RZERO,R1
        / &     DBUS SALU & CKALU & CKREG
        ;
        ;(3) THE CONTENTS OF IR0 ARE DOWNSHIFTED.
04AC            NAC ,,,CONT
        / &     ALU ,,RAMD,OR,OR,ZA
        / &     REG ,,RZERO,RZERO
        / &     DBUS SNONE & CKALU & STATSHFT B"0000000000000
        ;
        ;(4) THE CONTENTS OF IR0 ARE DOWNSHIFTED.
04AD            NAC ,,,CONT
        / &     ALU ,,RAMD,OR,OR,ZA
        / &     REG ,,RZERO),RZERO
        / &     DBUS SNONE & CKALU & STATSHFT B"0000000000000
        ;
```



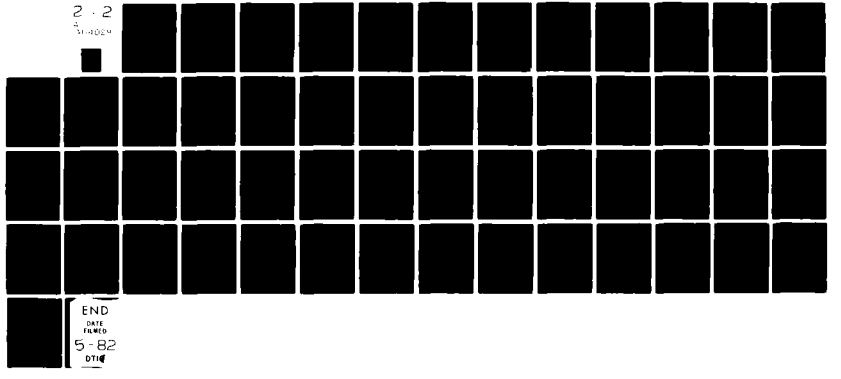
```
AMD29/29 AMDASM MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750A. (EMSL

        ;(5) THE CONTENTS OF ER0 ARE DOWNSHIFTED.
04AA            NAC ,,,CONT
        / &     ALU ,,RAMD,OR,OR,ZA
        / &     REG ,,RZERO,RZERO
        / &     DBUS SNONE & CKALU & STATSHFT B"0000000000000
        ;
        ; (6) THE CONTENTS OF ER0 ARE DOWNSHIFTED.
        ;     THE RESULT
04AB            NAC ,,,CONT
        / &     ALU ,,RAMD,OR,OR,ZA
        / &     REG ,,RZERO,RZERO
        / &     DBUS SNONE & CKALU & STATSHFT B"0000000000010
        ;
        ;(7) THE CONTENTS OF IR0 ARE DECREMENTED BY 1.
04AC            NAC ,,,CONT & ALU ,,ABUS,RAMD,OR,ZA & REG ,,RZERO,R0 & DBUS SALU
        / &     CKREG
        ;
        ;(8) THE CONTENTS OF IR0 ARE SWAPPED WITH THE CONTENTS OF ER0.
04AD            NAC ,FALSE,TZERO,CJPL
        / &     ALU ,,ABUS,RAMA,S,DB,**
        / &     REG ,,RZERO,RZERO,R0
        / &     CKALU & CKREG & DBUS SALU
        ;
        ;---------- ---------- ---------- ----------
        ;
        ;----------
        ;
        ; SHIFT LEFT LOGICAL
        ; ******************
        ;
        ; ENTRY POINT
        ; ***********
        ;
04AE OPCODE60:: NAC ,FALSE,TZERO,CJS
        / &     ALU & DBUS SNONE & BADDRESS SHFTNM
        ;
        ;(2) THE CONTENTS OF GR2 ARE UPSHIFTED LOGICAL.
04AF SLL1:      NAC ,,,CONT
        / &     ALU ,,RAMD,OR,OR,ZA
        / &     CKALU & REG ,,GR2,GR2
        / &     STATSHFT B"0001000000
        ;
        ;(3) 1 IS SUBTRACTED FROM THE CONTENTS OF ER0
04B0            NAC ,,,CONT
        / &     ALU ,,ANDT,NOP,SUBS,SUBS,DZ
```

```
AMDOS/29 AMDASM MICRO ASSEMBLER, V1.2
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750A INSTRUCTIONS

        / &         CKALU & REG ,,,,R+ & DBUS SALU
        / &         STATSHFT B"0000000,0000" & "CKREG"
        ;
        ;(4) JUMP TO SRL1 TAKES PLACE IF ZLSB IS NOT SET.
0491            NAC ,FALSE,ZLSB,CJP
        ;(5) THE CONTENTS OF ERO ARE TWO'S-SHIFTED LOGICAL
        / &         ALU & DBUS SNONE & BADDRESS SRL1
        ;
        ; &  RETURN TAKES PLACE AND THE MACHINE STATUS REGISTER IS LOADED.
0492            NAC ,FALSE,TZERO,CRTN
        / &         ALU ,B"0",,NOP,CR,OR,ZA
        / &         CKALU & REG ,,SR2,SR2
        / &         STATSHFT B"0000000010000
        / &         CARRYSEL "ZERO"
        ;
        ;-------- --- -------- ------ ---- ----------------------------------
        ;
        ; SHIFT RIGHT LOGICAL
        ; ********************
        ;
        ; ENTRY POINT
        ; ***********
        ;
0493 OPCODE01:  NAC ,FALSE,TZERO,CJS
        / &         ALU & DBUS SNONE & BADDRESS SHFTNM
        ;
        ;
0494 SRL1:      NAC ,,,CONT
        / &         ALU ,,,RAMQ,OR,/R,ZA
        / &         CKALU & REG ,,SR2,SR2
        / &         STATSHFT B"0000000000000
        ;
        ; (3) 1 IS SUBTRACTED FROM THE CONTENTS OF ERO
0495            NAC ,,,CONT
        / &         ALU ,,ABUS,NOP,SUBS,SUBS,DZ
        / &         CKALU & REG ,,,,RH & DBUS SALU
        / &         STATSHFT B"0000000010000 & CKREG
        ;
        ;(4) JUMP TO SRL1 TAKES PLACE IF ZLSB IS NOT SET.
0496            NAC ,FALSE,ZLSB,CJP
        / &         ALU & DBUS SNONE & BADDRESS SRL1
        ;
        ;(5) RETURN TAKES PLACE. MACHINE STATUS REGISTER IS LOADED.
0497            NAC ,FALSE,TZERO,CRTN
        / &         ALU ,B"0",,NOP,CR,OR,ZA
        / &         REG ,,,OR2 & DBUS SNONE & CARRYSEL OZERO
```



```
AMDOS/29 AMDASM MICRO ASSEMBLER, V1.2
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750A INSTRUCTIONS

        / &         STATSHFT B"0000000010000
        ;
        ;---------- --- -------- ------ ---- ----------------------------------
        ;
        ; SHIFT RIGHT CIRCULAR
        ; ********************
        ;
04B4 SRC1:      NAC ,,,C NT
        / &         ALU ,,,RAMQ,,,R,ZA
        / &         CKALU & REG ,,,RC,OR2
        / &         STATSHFT B"0010100000000
        ;
        ;(3) 1 IS SUBTRACTED FROM THE CONTENTS OF ERO
04B5            NAC ,,,C NT
        / &         ALU ,,AB ,,A ,,ZERO,SUBS,DZ
        / &         CKALU & REG ,,,,R+ & DBUS SALU
        / &         STATSHFT B"0000000A01000 & CKER
        ;
        ;(4) JUMP TO SRC1 TAKES PLACE IF ZLSB ( A ) SET.
04B6            NAC ,FALSE,ZLSB,CJP
        / &         ALU & DBUS SN NE & BADDRESS SRC1
        ;
        ;(5) RETURN TAKES PLACE AND THE MACHINE STATUS REGISTER IS LOADED.
04B7            NAC ,FALSE,TZERO,CRTN
        / &         ALU ,B"0",,NOP,CR,OR,ZA
        / &         CKALU & REG ,,,OR2,OR2
        / &         STATSHFT B"0000000010000
        / &         CARRYSEL "ZERO"
        ;
        ;----------- --- -------- ------ ---- ----------------------------------
        ;
        ; SHIFT LEFT ARITHMETIC
        ; *********************
        ;
04B8 SLA1:      NAC ,,,CONT & ALU ,,,RAMQ, R,OR,ZA & CKALU & REG ,,OR2,OR2,RH
        / &         STATSHFT B"0000010000000
        / &         DBUS SALU & CKREG
        ;
        ;  THE SHIFTED CONTENTS OF ERO ARE EXCLUSIVE ORED WITH THE ORIGINAL CONTENTS
        ; (NOW IN ERQ).  THE CARRY IS LOADED WITH THE RESULT.
04B9            NAC ,,,CONT & ALU ,,ABUS,OPIS,FIOP,FIOP,OA & REG ,,,OR2,RB & CKALU
        / &         DBUS SNONE
        ;
        ;  THE CONTENTS OF THE ,REG ARE MASKED BY 10000000000000.
04BE            NAC ,,,CONT & ALU ,,IBUS,NOP,AND,AND,DZ & DBUS SDATA
        / &         DATINSRT DW5276H
```

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 SHRIMPTON
UNCLASSIFIED RAE-FS-403 DRIC-BR-80502 NL



```
AMDOS/29 AMDASM MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EMSL
      ;-------------------------------------------------------------------
      ;
      ; DOUBLE SHIFT LEFT LOGICAL
      ; *************************
      ;
      ; ENTRY POINT
      ; ***********
      ;
04CE OPCODE65:: NAC ,FALSE,TZERO,CJS
      / &       ALU & DBUS SNONE & BADDRESS SHFTNM
      ;
      ;(2) THE CONTENTS OF REGISTER RB+1 (GR2+1) ARE LOADED INTO THE QREG
04CF DSLL2:     NAC ,,,CONT
      / &       ALU ,,,QREG,OR,OR,ZA
      / &       REG ,B#01,,GR2 & DBUS SNONE & CKALU
      ;
      ;(3) THE CONTENTS OF RB (GR2) AND QREG ARE CONCATENATED AND UPSHIFTED
      ; LOGICAL.
04D0 DSLL1:     NAC ,,,CONT
      / &       ALU ,,,RAMQU,OR,OR,ZA
      / &       REG ,,GR2,GR2
      / &       DBUS SNONE & CKALU & STATSHFT B#0001100000000
      ;
      ;(4) 1 IS SUBTRACTED FROM THE CONTENTS OF ER9 (THE CARRY INPUT IS SET
      ; TO 0 BY STATSHFT CAUSING THIS TO HAPPEN)
04D1            NAC ,,,CONT
      / &       ALU ,,ABUS,NOP,SUBS,SUBS,DZ
      / &       REG ,,,,R9 & DBUS SALU
      / &       CKREG & STATSHFT B#0000000000000
      ;
      ;(5) JUMP TO DSLL1 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.
04D2            NAC ,FALSE,ZLSB,CJP
      / &       ALU & DBUS SNONE & BADDRESS DSLL1
      ;
      ;  (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.
04D3            NAC ,FALSE,TZERO,CJS
      / &       ALU & DBUS SNONE & BADDRESS ASDLSTG2
      ;
      ; (9) RETURN TAKES PLACE.
04D4            NAC ,FALSE,TZERO,CRTN & ALU & DBUS SNONE
      ;
      ;
```



```
AMDOS/29 AMDASM MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EMSL
      ;-------------------------------------------------------------------
      ; DOUBLE SHIFT RIGHT LOGICAL
      ; **************************
      ;
      ; ENTRY POINT
      ; ***********
      ;
04D5 OPCODE66:: NAC ,FALSE,TZERO,CJS
      / &       ALU & DBUS SNONE & BADDRESS SHFTNM
      ;
      ;(2) THE CONTENTS OF REGISTER RB+1 (GR2+1) ARE LOADED INTO THE QREG
04D6 DSRL2:     NAC ,,,CONT
      / &       ALU ,,,QREG,OR,OR,ZA
      / &       REG ,B#01,,GR2 & DBUS SNONE & CKALU
      ;
      ;(3) THE CONTENTS OF RB (GR2) AND QREG ARE CONCATENATED AND DOWNSHIFTED
      ; LOGICAL.
04D7 DSRL1:     NAC ,,,CONT
      / &       ALU ,,,RAMQD,OR,OR,ZA
      / &       REG ,,GR2,GR2
      / &       DBUS SNONE & CKALU & STATSHFT B#0001100000000
      ;
      ;(4) 1 IS SUBTRACTED FROM THE CONTENTS OF ER9 (THE CARRY INPUT IS SET
      ; TO 0 BY STATSHFT CAUSING THIS TO HAPPEN)
04D8            NAC ,,,CONT
      / &       ALU ,,ABUS,NOP,SUBS,SUBS,DZ
      / &       REG ,,,,R9 & DBUS SALU
      / &       CKREG & STATSHFT B#0000000000000
      ;
      ;(5) JUMP TO DSRL1 TAKES PLACE IF ZLSB IS NOT SET
04D9            NAC ,FALSE,ZLSB,CJP
      / &       ALU & DBUS SNONE & BADDRESS DSRL1
      ;
      ;  (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.
04DA            NAC ,FALSE,TZERO,CJS
      / &       ALU & DBUS SNONE & BADDRESS ASDLSTG2
      ;
      ; (9) RETURN TAKES PLACE.
04DB            NAC ,FALSE,TZERO,CRTN & ALU & DBUS SNONE
      ;
      ;-------------------------------------------------------------------
      ;
      ; DOUBLE SHIFT RIGHT ARITHMETIC
      ; *****************************
```

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

      ;
      ;
      ; ENTRY POINT
      ; ***********
      ;
04DC OPCODE87:: NAC ,FALSE,TZERO,CJS
     / &        ALU & DBUS SNONE & BADDRESS SHFTNM
      ;
      ;(2) THE CONTENTS OF REGISTER RB+1 (GR2+1) ARE LOADED INTO THE QREG
04DD DSRA2:     NAC ...CONT
     / &        ALU ...QREG,OR,OR,ZA
     / &        REG ,B#01,,GR2 & DBUS SNONE & CKALU
      ;
      ;(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.
04DE            NAC ...CONT
     / &        ALU ...RAMU,OR,OR,ZA
     / &        REG ,,GR2,GR2 & DBUS SNONE
     / &        STATSHFT B#0000000000000
      ;
      ;(3) THE CONTENTS OF RB (GR2) AND QREG ARE CONCATENATED AND DOWNSHIFTED
      ; WITH THE CARRY GOING INTO THE MS BIT..
04DF DSRA1:     NAC ...CONT
     / &        ALU ...RAMQD,OR,OR,ZA
     / &        REG ,,GR2,GR2
     / &        DBUS SNONE & CKALU & STATSHFT B#0001000000000
      ;
      ;(4) 1 IS SUBTRACTED FROM THE CONTENTS OF ER9 (THE CARRY INPUT IS SET
      ; TO 0 BY STATSHFT CAUSING THIS TO HAPPEN)
04E0            NAC ...CONT
     / &        ALU ,,ABUS,NOP,SUBS,SUBS,DZ
     / &        REG ....R9 & DBUS SALU
     / &        CKREG & STATSHFT B#0000000000000
      ;
      ;(5) JUMP TO DSRA1 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.
04E1            NAC ,FALSE,ZLSB,CJP
     / &        ALU & DBUS SNONE & BADDRESS DSRA1
      ;
      ;  (6) 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.
04E2            NAC ,FALSE,TZERO,CJS
     / &        ALU & DBUS SNONE & BADDRESS ASDLSTG2
```



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

      ;
      ; (9) RETURN TAKES PLACE.
04E3            NAC ,FALSE,TZERO,CRTN & ALU & DBUS SNONE
      ;
      ;-----------------------------------------------------------------------
      ;
      ; DOUBLE SHIFT LEFT ARITHMETIC
      ; ****************************
      ;
      ;(1) THE CONTENTS OF REGISTER RB+1 (GR2+1) ARE LOADED INTO THE QREG
04E4 DSLA1:     NAC ...CONT
     / &        ALU ...QREG,OR,OR,ZA
     / &        REG ,B#01,,GR2 & DBUS SNONE & CKALU
      ;
      ;(2) THE CONTENTS OF RB (GR2) AND QREG ARE CONCATENATED AND UPSHIFTED
      ; LOGICAL. THE ORIGINAL CONTENTS ARE PLACED IN ER8.
04E5 DSLA2:     NAC ...CONT
     / &        ALU ...RAMQU,OR,OR,ZA
     / &        REG ,,GR2,GR2,R8
     / &        DBUS SALU & CKALU & STATSHFT B#0001100000000
     / &        CKREG
      ;
      ; THE CONTENTS OF ER8 ARE EXORED WITH GR2.  THE CONTENTS OF GR2 ARE PLACED IN
      ; ER8.  THE RESULT OF THE FUNCTION IS PLACED IN GR2.
04E6            NAC ,,CONT & ALU ,,ABUS,RAMA,EXOR,EXOR,DA & REG ,,GR2,GR2,R8
     / &        DBUS SALU & CKREG & CKALU
      ;
      ; THE CONTENTS OF GR2 ARE MASKED BY 1000000000000000. NO REGISTERS ARE LOADED.
04E7            NAC ,,CONT & ALU ,,DPUS,NOP,AND,AND,DA & REG ,,GR2 & DBUS SDATA
     / &        DATINSRT D#32768
      ;
      ;(3) JUMP SUBROUTINE TO OVERFLOW TAKES PLACE IF THE ZMSB AUXILIARY STATUS
      ;  BIT WAS NOT SET. THE CONTENTS OF IR2 ARE RESTORED.
04E8            NAC ,FALSE,ZMSB,CJS & ALU ,,ABUS,RAMP,OR,OR,DZ & REG ,,GR2,,R8
     / &        DBUS SNONE & BADDRESS OVERFLOW & CKALU
      ;
      ;(4) 1 IS SUBTRACTED FROM THE CONTENTS OF ER9 (THE CARRY INPUT IS SET
      ; TO 0 BY STATSHFT CAUSING THIS TO HAPPEN)
04E9            NAC ...CONT
     / &        ALU ,,ABUS,NOP,SUBS,SUBS,DZ
     / &        REG ....R9 & DBUS SALU
     / &        CKREG & STATSHFT B#0000000000000
      ;
      ;(5) JUMP TO DSLA2 TAKES PLACE IF ZLSB IS NOT SET
      ;  THE MACHINE STATUS REGISTER IS LOADED ACCORDING TO THE CONTENTS OF
```

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

      ;  GR2 (THE MOST SIGNIFICANT HALF OF THE DOUBLE LENGTH
      ;  WORD.
04EA            NAC ,FALSE,ZLSB,CJP
     / &        ALU & DBUS SNONE & BADDRESS DSLA2
      ;
      ;  (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.
04EB            NAC ,FALSE,TZERO,CJS
     / &        ALU & DBUS SNONE & BADDRESS ASDLSTG2
      ;
      ; (9) RETURN TAKES PLACE.
04EC            NAC ,FALSE,TZERO,CRTN & ALU & DBUS SNONE
      ;-------------------------------------------------------------------------
      ;
      ;
      ; DOUBLE SHIFT LEFT CYCLIC
      ; ************************
      ;
      ;
      ; ENTRY POINT
      ; ***********
      ;
04ED OPCODE68:: NAC ,FALSE,TZERO,CJS
     / &        ALU & DBUS SNONE & BADDRESS SHFTNM
      ;
      ;(2) THE CONTENTS OF REGISTER RB+1 (GR2+1) ARE LOADED INTO THE QREG
04EE DSLC2:     NAC ...CONT
     / &        ALU ...QREG,OR,OR,ZA
     / &        REG ,B#01,,GR2 & DBUS SNONE & CKALU
      ;
      ;(3) THE CONTENTS OF RB (GR2) AND QREG ARE CONCATENATED AND UPSHIFTED
      ; CYCLIC.
04EF DSLC1:     NAC ...CONT
     / &        ALU ...RAMQU,OR,OR,ZA
     / &        REG ,,GR2,GR2
     / &        DBUS SNONE & CKALU & STATSHFT B#0011110000000
      ;
      ;(4) 1 IS SUBTRACTED FROM THE CONTENTS OF ER9 (THE CARRY INPUT IS SET
      ; TO 0 BY STATSHFT CAUSING THIS TO HAPPEN)
04F0            NAC ...CONT
     / &        ALU ,,ABUS,NOP,SUBS,SUBS,DZ
     / &        REG ,...,R9 & DBUS SALU
     / &        CEREG & STATSHFT B#0000000000000
      ;
      ;(5) JUMP TO DSLC1 TAKES PLACE IF ZLSB IS NOT SET
      ;  THE MACHINE STATUS REGISTER IS LOADED ACCORDING TO THE CONTENTS OF
```



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

      ;  GR2 (THE MOST SIGNIFICANT HALF OF THE DOUBLE LENGTH
      ;  WORD.
04F1            NAC ,FALSE,ZLSB,CJP
     / &        ALU & DBUS SNONE & BADDRESS DSLC1
      ;
      ;  (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.
04F2            NAC ,FALSE,TZERO,CJS
     / &        ALU & DBUS SNONE & BADDRESS ASDLSTG2
      ;
      ; (9) RETURN TAKES PLACE.
04F3            NAC ,FALSE,TZERO,CRTN & ALU & DBUS SNONE
      ;-------------------------------------------------------------------------
      ;
      ; DOUBLE SHIFT RIGHT CYCLIC
      ; *************************
      ;
      ;
      ;(2) THE CONTENTS OF REGISTER RB+1 (GR2+1) ARE LOADED INTO THE QREG
04F4 DSRC2:     NAC ...CONT
     / &        ALU ...QREG,OR,OR,ZA
     / &        REG ,B#01,,GR2 & DBUS SNONE & CKALU
      ;
      ;(3) THE CONTENTS OF RB (GR2) AND QREG ARE CONCATENATED AND DOWNSHIFTED
      ; CYCLIC.
04F5 DSRC1:     NAC ...CONT
     / &        ALU ...RAMQD,OR,OR,ZA
     / &        REG ,,GR2,GR2
     / &        DBUS SNONE & CKALU & STATSHFT B#0011110000000
      ;
      ;(4) 1 IS SUBTRACTED FROM THE CONTENTS OF ER9 (THE CARRY INPUT IS SET
      ; TO 0 BY STATSHFT CAUSING THIS TO HAPPEN)
04F6            NAC ...CONT
     / &        ALU ,,ABUS,NOP,SUBS,SUBS,DZ
     / &        REG ,...,R9 & DBUS SALU
     / &        CEREG & STATSHFT B#0000000000000
      ;
      ;(5) JUMP TO DSRC1 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.
04F7            NAC ,FALSE,ZLSB,CJP
     / &        ALU & DBUS SNONE & BADDRESS DSRC1
      ;
      ;  (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.
```

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

04F8                MAC ,FALSE,TZERO,CJS
     / &            ALU & DBUS SNONE & BADDRESS ASDLST02
     ;
     ; (9) RETURN TAKES PLACE.
04F9                MAC ,FALSE,TZERO,CRTN & ALU & DBUS SNONE
     ;
     ;-------------------------------------------------------------------------------
     ;
     ; SHFTND16 -- 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.
04FA SHFTND16:  MAC ,,,CONT & ALU B#0,,,NOP,OR,OR,ZA & REG ,,,GR2,R9 & CKREG
     / &        DBUS SALU & STATSHFT B#0000000010000
     ;
     ;(2) RETURN TAKES PLACE IF MICRO Z IS SET INDICATING A ZERO SHIFT.  0 IS PLACED
     ; IN ER8
04FB                MAC ,TRUE,MUX,CRTN & ALU ,,,NOP,AND,AND,DZ & DBUS SALU & REG ,,,,R8
     / &        CKREG & STATSHFT B#0000000010100
     ;
     ;(2A) 1 IS SUBTRACTED FROM GR2 AND THE RESULT PLACED IN THE QREG.
04FC                MAC ,,,CONT & ALU ,,,QREG,SUBR,SUBR,ZA & RE2 ,,,GR2
     / &        STATSHFT B#0000000000000 & CKALU & DBUS SNONE
     ;
     ;(3) QREG CONTENTS ARE MASKED WITH 1111111111110000, MICROSTATUS IS LOADED
     ;  ACCORDING TO RESULT. NB THE MASK PATTERN NEEDED IS COINCIDENALLY THE SAME
     ;  AS A STATUS AND SHIFT PATTERN THAT WILL LOAD THE MICROSTATUS REGISTER.
04FD                MAC ,,,CONT & ALU B#0,,,DBUS,NOP,AND,AND,DQ & DBUS SDATA
     / &        DATAINSRT D#65520
     ;
     ;
     ;(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.
04FE                MAC ,TRUE,MUX,CRTN & ALU ,,ABUS,NOP,ADD,ADD,DZ & REG ,,,,R8
     / &        DBUS SALU & STATSHFT B#0100000010100 & CKREG
     ;
     ;(5) GR2 IS NEGATED AND PLACED IN ER9.
04FF                MAC ,,,CONT & ALU ,,,NOP,SUBS,SUBS,ZA & REG ,,,GR2,R9 & CKREG
```



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

     / &        DBUS SALU & STATSHFT B#0100000000000
     ;
     ;(6) 1 IS SUBTRACTED FROM THE CONTENTS OF ER9 AND THE RESULT
     ; PLACED IN QREG.
0500                MAC ,,,CONT & ALU ,,ABUS,QREG,SUBS,SUBS,DZ & REG ,,,,R9 & CKALU
     / &        STATSHFT B#0000000000000
     ;
     ;(7) CONTENTS OF QREG ARE MASKED WITH 1111111111110000 AND THE MICROSTATUS
     ; REGISTER SET ACCORDING TO THE RESULT.
0501                MAC ,,,CONT & ALU B#0,,DBUS,NOP,AND,AND,DQ & DBUS SDATA
     / &        DATAINSRT D#65520
     ;
     ;(8) RETURN TAKES PLACE IF MICROSTATUS ZERO IS SET. 1 IS ADDED TO THE CONTENTS
     ; OF ER8.
0502                MAC ,TRUE,MUX,CRTN & ALU ,,ABUS,NOP,ADD,ADD,DZ & DBUS SALU & CKREG
     / &        REG ,,,,R8 & STATSHFT B#0100000010100
     ;
     ;(9) RETURN TAKES PLACE, 1 IS ADDED TO ER8.
0503                MAC ,FALSE,TZERO,CRTN & ALU ,,ABUS,NOP,ADD,ADD,DZ & DBUS SALU & CKREG
     / &        REG ,,,,R8 & STATSHFT B#0100000000000
     ;
     ;-------------------------------------------------------------------------------
     ; SHFTND32 -- 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.
0504 SHFTND32:     MAC ,,,CONT & ALU B#0,,,NOP,OR,OR,ZA & RE2 ,,,GR2,R9 & CKREG
     / &        DBUS SALU & STATSHFT B#0000000010000
     ;
     ;(2) RETURN TAKES PLACE IF MICRO Z IS SET INDICATING A ZERO SHIFT.  0 IS PLACED
     ; IN ER8
0505                MAC ,TRUE,MUX,CRTN & ALU ,,,NOP,AND,AND,DZ & DBUS SALU & REG ,,,,R8
     / &        CKREG & STATSHFT B#0000000010100
     ;
     ;(2A) 1 IS SUBTRACTED FROM GR2 AND THE RESULT PLACED IN THE QREG.
0506                MAC ,,,CONT & ALU ,,,QRE2,SUBR,SUBR,ZA & RE2 ,,,GR2
     / &        STATSHFT B#0000000000000 & CKALU & DBUS SNONE
     ;
     ;(3) QREG CONTENTS ARE MASKED WITH 1111111111100000, MICROSTATUS IS LOADED
```

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

      ;  ACCORDING TO RESULT. NB THE MASK PATTERN NEEDED IS COINCIDENALLY THE SAME
      ;  AS A STATUS AND SHIFT PATTERN THAT WILL LOAD THE MICROSTATUS REGISTER.
0507            NAC ,,,CONT & ALU B##,,DBUS,NOP,AND,AND,DQ & DBUS SDATA
      / &       DATINSRT D#65504
      ;
      ;
      ;(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.
0508            NAC ,TRUE,MUX,CRTN & ALU ,,ABUS,NOP,ADD,ADD,DZ & REG ,,,,R8
      / &       DBUS SALU & STATSHFT B#0100000010100 & CKREG
      ;
      ;(5) GR2 IS NEGATED AND PLACED IN ER9.
0509            NAC ,,,CONT & ALU ,,,NOP,SUBS,SUBS,ZA & REG ,,,GR2,R9 & CKREG
      / &       DBUS SALU & STATSHFT B#0100000000000
      ;
      ;(6) 1 IS SUBTRACTED FROM THE CONTENTS OF ER9 AND THE RESULT
      ; PLACED IN QREG.
050A            NAC ,,,CONT & ALU ,,ABUS,QREG,SUBS,SUBS,DZ & REG ,,,,R9 & CKALU
      / &       STATSHFT B#0000000000000
      ;
      ;(7) CONTENTS OF QREG ARE MASKED WITH 1111111111100000 AND THE MICROSTATUS
      ; REGISTER SET ACCORDING TO THE RESULT.
050B            NAC ,,,CONT & ALU B##,,DBUS,NOP,AND,AND,DQ & DBUS SDATA
      / &       DATINSRT D#65504
      ;
      ;(8) RETURN TAKES PLACE IF MICROSTATUS ZERO IS SET. 1 IS ADDED TO THE CONTENTS
      ; OF ER8.
050C            NAC ,TRUE,MUX,CRTN & ALU ,,ABUS,NOP,ADD,ADD,DZ & DBUS SALU & CKREG
      / &       REG ,,,,R8 & STATSHFT B#0100000010100
      ;
      ;(9) RETURN TAKES PLACE, 1 IS ADDED TO ER8.
050D            NAC ,FALSE,TZERO,CRTN & ALU ,,ABUS,NOP,ADD,ADD,DZ & DBUS SALU & CKREG
      / &       REG ,,,,R8 & STATSHFT B#0100000000000
      ;
      ;-------------------------------------------------------------------------
      ;PLGR1GR2 --- THIS SUBROUTINE PLACES THE CONTENTS OF THE SR1 FIELD IN THE
      ; INSTRUCTION IN THE GR2 FIELD POSITION IN THE INSTRUCTION REGISTER.
      ;
      ;(1)THE CONTENTS OF ER1 ARE SWOPPED WITH THE CONTENTS OF IR8
050E PLGR1GR2:  NAC ,,,CONT & ALU ,,ABUS,RAMA,OR,OR,DZ & REG ,,RZERO,RZERO,R1
      / &       CKALU & CKREG & DBUS SALU
      ;
      ;(2) THE CONTENTS OF IR8 ARE DOWN SHIFTED.
050F            NAC ,,,CONT & ALU ,,,RAMD,OR,OR,ZA & REG ,,RZERO,RZERO & CKALU
      / &       DBUS SNONE & STATSHFT B#0010100000000
```



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

      ;
      ;(3) THE CONTENTS OF IR8 ARE DOWNSHIFTED.
0510            NAC ,,,CONT & ALU ,,,RAMD,OR,OR,ZA & REG ,,RZERO,RZERO & CKALU
      / &       DBUS SNONE & STATSHFT B#0010100000000
      ;
      ;(4) THE CONTENTS OF IR8 ARE DOWNSHIFTED.
0511            NAC ,,,CONT & ALU ,,,RAMD,OR,OR,ZA & REG ,,RZERO,RZERO & CKALU
      / &       DBUS SNONE & STATSHFT B#0010100000000
      ;
      ;(5) THE CONTENTS OF IR8 ARE DOWNSHIFTED.
0512            NAC ,,,CONT & ALU ,,,RAMD,OR,OR,ZA & REG ,,RZERO,RZERO & CKALU
      / &       DBUS SNONE & STATSHFT B#0010100000000
      ;
      ;(5) THE CONTENTS OF IR8 ARE SWOPPED WITH THE CONTENTS OF ER1. THE INSTRUCTION
      ; REGISTER IS LOADED.
0513            NACF ,FALSE,TZERO,CRTN & ALU ,,ABUS,RAMA,OR,OR,DZ & DBUS SALU
      / &       REG ,,RZERO,RZERO,R1 & CKALU & CKREG
      ;
      ;-------------------------------------------------------------------------
      ;
      ; SHIFT LOGICAL COUNT IN REGISTER
      ; *******************************
      ;
      ; ENTRY POINT
      ; ***********
      ;
0514 OPCODE6A:: NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS SHFTND16
      ;
      ;(2) JUMP TAKES PLACE TO PLGR1GR2 SO THAT THE SR1 FIELD IN THE INSTRUCTION
      ; REGISTER IS TRANSFERRED TO THE GR2 FIELD SO THAT THE INSTRUCTIONS SLL,SRL,SRA
      ; SLC,DSLL,DSRL,DSRA,DSLC CAN BE USED.   THIS RESULTS IN A LARGE SAVING IN
      ; CODE ALTHOUGH THERE IS A TIME PENALTY INCURRED BY USING THIS SUBROUTINE.
0515            NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS PLGR1GR2
      ;
      ;(3) THE CONTENTS OF ER8 ARE PASSED THROUGH THE ALU.
0516            NAC ,,,CONT & ALU ,,APUS,NOP,OR,OR,DZ & DBUS SNONE & REG ,,,,R8
      ;
      ;(4) RETURN TAKES PLACE IF LS BYTE WAS ZERO IMPLYING NO SHIFT.
0517            NAC ,TRUE,ZLSB,CRTN & ALU ,,ABUS,NOP,SUBS,SUBS,DZ & REG ,,,,R8
      / &       CKREG & STATSHFT B#0000000000000 & DBUS SALU
      ;
      ;(5) JUMP TO SLL1 TAKES PLACE IF LS BYTE OF ER8 WAS ZERO AFTER SUBTRACTION IN
      ;   (3) INDICATING POSITIVE (IE. LEFT) SHIFT.
0518            NAC ,TRUE,ZLSB,CJP & ALU & DBUS SNONE & BADDRESS SLL1
      ;
      ;(6) 1 IS SUBTRACTED FROM ER8
```

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

 0519            NAC ,,,CONT & ALU ,,ABUS,NOP,SUBS,SUBS,DZ & REG ,,,,R8 & DBUS SALU
     / &         CKREG & STATSHFT B@@@@@@@@@@@@@
     ;
     ;(7) JUMP TO SRL1 TAKES PLACE IF LS BYTE OF ER8 WAS ZERO AFTER THE SUBTRACTION
     ;  MEANING A NEGATIVE SHIFT (IE RIGHT) IS REQUIRED.
 051A            NAC ,TRUE,ZLSB,CJP & ALU & DBUS SNONE & BADDRESS SRL1
     ;
     ;(8) IN THE EVENT THAT ER8 IS STILL NONZERO, JUMP TO OVERFLOW TAKES PLACE.
 051B            NAC ,,,JP & ALU & DBUS SNONE & BADDRESS OVERFLOW
     ;
     ;-----------------------------------------------------------------------
     ;
     ; SHIFT ARITHMETIC COUNT IN REGISTER
     ; @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
     ;
     ; ENTRY POINT
     ; ++++++++++++
     ;
 051C OPCODE6B:: NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS SHFTND16
     ;
     ;(2) JUMP TAKES PLACE TO PLGR1GR2 SO THAT THE GR1 FIELD IN THE INSTRUCTION
     ; REGISTER IS TRANSFERRED TO THE GR2 FIELD SO THAT THE INSTRUCTIONS SLL,SRL,SRA
     ; SLC,DSLL,DSRL,DSRA,DSLC CAN BE USED.   THIS RESULTS IN A LARGE SAVING IN
     ; CODE ALTHOUGH THERE IS A TIME PENALTY INCURRED BY USING THIS SUBROUTINE.
 051D            NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS PLGR1GR2
     ;
     ;(3) THE CONTENTS OF ER8 ARE PASSED THROUGH THE ALU.
 051E            NAC ,,,CONT & ALU ,,ABUS,NOP,OR,OR,DZ & DBUS SNONE & REG ,,,,R8
     ;
     ;(4) RETURN TAKES PLACE IF LS BYTE WAS ZERO IMPLYING NO SHIFT.
 051F            NAC ,TRUE,ZLSB,CRTN & ALU ,,ABUS,NOP,SUBS,SUBS,DZ & REG ,,,,R8
     / &         CKREG & STATSHFT B@@@@@@@@@@@@@ & DBUS SALU
     ;
     ;(5) JUMP TO SLA1 TAKES PLACE IF LS BYTE OF ER8 WAS ZERO AFTER SUBTRACTION IN
     ;   (3) INDICATING POSITIVE (IE. LEFT) SHIFT.
 0520            NAC ,TRUE,ZLSB,CJP & ALU & DBUS SNONE & BADDRESS SLA1
     ;
     ;(6) 1 IS SUBTRACTED FROM ER8
 0521            NAC ,,,CONT & ALU ,,ABUS,NOP,SUBS,SUBS,DZ & REG ,,,,R8 & DBUS SALU
     / &         CKREG & STATSHFT B@@@@@@@@@@@@@
     ;
     ;(7) JUMP TO SRA0 TAKES PLACE IF LS BYTE OF ER8 WAS ZERO AFTER THE SUBTRACTION
     ;  MEANING A NEGATIVE SHIFT (IE RIGHT) IS REQUIRED.
 0522            NAC ,TRUE,ZLSB,CJP & ALU & DBUS SNONE & BADDRESS SRA0
     ;
     ;(8) IN THE EVENT THAT ER8 IS STILL NONZERO, JUMP TO OVERFLOW TAKES PLACE.
```



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

 0523            NAC ,,,JP & ALU & DBUS SNONE & BADDRESS OVERFLOW
     ;
     ;-----------------------------------------------------------------------
     ;
     ; SHIFT CYCLIC COUNT IN REGISTER
     ; @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
     ;
     ; ENTRY POINT
     ; ++++++++++++
     ;
 0524 OPCODE6C:: NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS SHFTND16
     ;
     ;(2) JUMP TAKES PLACE TO PLGR1GR2 SO THAT THE GR1 FIELD IN THE INSTRUCTION
     ; REGISTER IS TRANSFERRED TO THE GR2 FIELD SO THAT THE INSTRUCTIONS SLL,SRL,SRA
     ; SLC,DSLL,DSRL,DSRA,DSLC CAN BE USED.   THIS RESULTS IN A LARGE SAVING IN
     ; CODE ALTHOUGH THERE IS A TIME PENALTY INCURRED BY USING THIS SUBROUTINE.
 0525            NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS PLGR1GR2
     ;
     ;(3) THE CONTENTS OF ER8 ARE PASSED THROUGH THE ALU.
 0526            NAC ,,,CONT & ALU ,,ABUS,NOP,OR,OR,DZ & DBUS SNONE & REG ,,,,R8
     ;
     ;(4) RETURN TAKES PLACE IF LS BYTE WAS ZERO IMPLYING NO SHIFT.
 0527            NAC ,TRUE,ZLSB,CRTN & ALU ,,ABUS,NOP,SUBS,SUBS,DZ & REG ,,,,R8
     / &         CKREG & STATSHFT B@@@@@@@@@@@@@ & DBUS SALU
     ;
     ;(5) JUMP TO SLC1 TAKES PLACE IF LS BYTE OF ER8 WAS ZERO AFTER SUBTRACTION IN
     ;   (3) INDICATING POSITIVE (IE. LEFT) SHIFT.
 0528            NAC ,TRUE,ZLSB,CJP & ALU & DBUS SNONE & BADDRESS SLC1
     ;
     ;(6) 1 IS SUBTRACTED FROM ER8
 0529            NAC ,,,CONT & ALU ,,ABUS,NOP,SUBS,SUBS,DZ & REG ,,,,R8 & DBUS SALU
     / &         CKREG & STATSHFT B@@@@@@@@@@@@@
     ;
     ;(7) JUMP TO SRC1 TAKES PLACE IF LS BYTE OF ER8 WAS ZERO AFTER THE SUBTRACTION
     ;  MEANING A NEGATIVE SHIFT (IE RIGHT) IS REQUIRED.
 052A            NAC ,TRUE,ZLSB,CJP & ALU & DBUS SNONE & BADDRESS SRC1
     ;
     ;(8) IN THE EVENT THAT ER8 IS STILL NONZERO, JUMP TO OVERFLOW TAKES PLACE.
 052B            NAC ,,,JP & ALU & DBUS SNONE & BADDRESS OVERFLOW
     ;
     ;-----------------------------------------------------------------------
     ;
     ; DOUBLE SHIFT LOGICAL COUNT IN REGISTER
     ; @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
     ;
     ; ENTRY POINT
```

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

      ; ++++++++++++
      ;
 052C OPCODE6D:: NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS SHFTND32
      ;
      ;(2) JUMP TAKES PLACE TO PLGR1GR2 SO THAT THE GR1 FIELD IN THE INSTRUCTION
      ; REGISTER IS TRANSFERRED TO THE GR2 FIELD SO THAT THE INSTRUCTIONS SLL,SRL,SRA
      ; SLC,DSLL,DSRL,DSRA,DSLC CAN BE USED.   THIS RESULTS IN A LARGE SAVING IN
      ; CODE ALTHOUGH THERE IS A TIME PENALTY INCURRED BY USING THIS SUBROUTINE.
 052D           NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS PLGR1GR2
      ;
      ;(3) THE CONTENTS OF ER8 ARE PASSED THROUGH THE ALU.
 052E           NAC ,,,CONT & ALU ,,ABUS,NOP,OR,OR,DZ & DBUS SNONE & REG ....R8
      ;
      ;(4) RETURN TAKES PLACE IF LS BYTE WAS ZERO IMPLYING NO SHIFT.
 052F           NAC ,TRUE,ZLSB,CRTN & ALU ,,ABUS,NOP,SUBS,SUBS,DZ & REG ....,R8
     / &        CKREG & STATSHFT B#@@@@@@@@@@@@ & DBUS SALU
      ;
      ;(5) JUMP TO DSLL2 TAKES PLACE IF LS BYTE OF ER8 WAS ZERO AFTER SUBTRACTION IN
      ;   (3) INDICATING POSITIVE (IE. LEFT) SHIFT.
 0530           NAC ,TRUE,ZLSB,CJP & ALU & DBUS SNONE & BADDRESS DSLL2
      ;
      ;(6) 1 IS SUBTRACTED FROM ER8
 0531           NAC ,,,CONT & ALU ,,ABUS,NOP,SUBS,SUBS,DZ & REG ....R8 & DBUS SALU
     / &        CKREG & STATSHFT B#@@@@@@@@@@@@
      ;
      ;(7) JUMP TO DSRL2 TAKES PLACE IF LS BYTE OF ER8 WAS ZERO AFTER THE SUBTRACTION
      ;  MEANING A NEGATIVE SHIFT (IE RIGHT) IS REQUIRED.
 0532           NAC ,TRUE,ZLSB,CJP & ALU & DBUS SNONE & BADDRESS DSRL2
      ;
      ;(8) IN THE EVENT THAT ER8 IS STILL NONZERO, JUMP TO OVERFLOW TAKES PLACE.
 0533           NAC ,,,JP & ALU & DBUS SNONE & BADDRESS OVERFLOW
      ;
      ;-------------------------------------------------------------------------
      ;
      ; DOUBLE SHIFT ARITHMETIC COUNT IN REGISTER
      ; ****************************************
      ;
      ; ENTRY POINT
      ; ++++++++++++
      ;
 0534 OPCODE6E:: NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS SHFTND32
      ;
      ;(2) JUMP TAKES PLACE TO PLGR1GR2 SO THAT THE GR1 FIELD IN THE INSTRUCTION
      ; REGISTER IS TRANSFERRED TO THE GR2 FIELD SO THAT THE INSTRUCTIONS SLL,SRL,SRA
      ; SLC,DSLL,DSRL,DSRA,DSLC CAN BE USED.   THIS RESULTS IN A LARGE SAVING IN
      ; CODE ALTHOUGH THERE IS A TIME PENALTY INCURRED BY USING THIS SUBROUTINE.
```



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

 0535           NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS PLGR1GR2
      ;
      ;(3) THE CONTENTS OF ER8 ARE PASSED THROUGH THE ALU.
 0536           NAC ,,,CONT & ALU ,,ABUS,NOP,OR,OR,DZ & DBUS SNONE & REG ....R8
      ;
      ;(4) RETURN TAKES PLACE IF LS BYTE WAS ZERO IMPLYING NO SHIFT.
 0537           NAC ,TRUE,ZLSB,CRTN & ALU ,,ABUS,NOP,SUBS,SUBS,DZ & REG ....,R8
     / &        CKREG & STATSHFT B#@@@@@@@@@@@@ & DBUS SALU
      ;
      ;(5) JUMP TO DSLA1 TAKES PLACE IF LS BYTE OF ER8 WAS ZERO AFTER SUBTRACTION IN
      ;   (3) INDICATING POSITIVE (IE. LEFT) SHIFT.
 0538           NAC ,TRUE,ZLSB,CJP & ALU & DBUS SNONE & BADDRESS DSLA1
      ;
      ;(6) 1 IS SUBTRACTED FROM ER8
 0539           NAC ,,,CONT & ALU ,,ABUS,NOP,SUBS,SUBS,DZ & REG ....R8 & DBUS SALU
     / &        CKREG & STATSHFT B#@@@@@@@@@@@@
      ;
      ;(7) JUMP TO DSRA2 TAKES PLACE IF LS BYTE OF ER8 WAS ZERO AFTER THE SUBTRACTION
      ;  MEANING A NEGATIVE SHIFT (IE RIGHT) IS REQUIRED.
 053A           NAC ,TRUE,ZLSB,CJP & ALU & DBUS SNONE & BADDRESS DSRA2
      ;
      ;(8) IN THE EVENT THAT ER8 IS STILL NONZERO, JUMP TO OVERFLOW TAKES PLACE.
 053B           NAC ,,,JP & ALU & DBUS SNONE & BADDRESS OVERFLOW
      ;
      ;-------------------------------------------------------------------------
      ;
      ; DOUBLE SHIFT CYCLIC COUNT IN REGISTER
      ; ************************************
      ;
      ; ENTRY POINT
      ; ++++++++++++
      ;
 053C OPCODE6F:: NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS SHFTND32
      ;
      ;(2) JUMP TAKES PLACE TO PLGR1GR2 SO THAT THE GR1 FIELD IN THE INSTRUCTION
      ; REGISTER IS TRANSFERRED TO THE GR2 FIELD SO THAT THE INSTRUCTIONS SLL,SRL,SRA
      ; SLC,DSLL,DSRL,DSRA,DSLC CAN BE USED.   THIS RESULTS IN A LARGE SAVING IN
      ; CODE ALTHOUGH THERE IS A TIME PENALTY INCURRED BY USING THIS SUBROUTINE.
 053D           NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS PLGR1GR2
      ;
      ;(3) THE CONTENTS OF ER8 ARE PASSED THROUGH THE ALU.
 053E           NAC ,,,CONT & ALU ,,ABUS,NOP,OR,OR,DZ & DBUS SNONE & REG ....,R8
      ;
      ;(4) RETURN TAKES PLACE IF LS BYTE WAS ZERO IMPLYING NO SHIFT.
 053F           NAC ,TRUE,ZLSB,CRTN & ALU ,,ABUS,NOP,SUBS,SUBS,DZ & REG ....,R8
     / &        CKREG & STATSHFT B#@@@@@@@@@@@@ & DBUS SALU
```

```
AMD05/29 AMDASM MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EMSL

     ;
     ;(5) JUMP TO DSLC2 TAKES PLACE IF LS BYTE OF ER8 WAS ZERO AFTER SUBTRACTION IN
     ;    (3) INDICATING POSITIVE (IE. LEFT) SHIFT.
0540            NAC ,TRUE,ZLSB,CJP & ALU & DBUS SNONE & BADDRESS DSLC2
     ;
     ;(6) 1 IS SUBTRACTED FROM ER8
0541            NAC ,,,CONT & ALU ,,ABUS,NOP,SUBS,SUBS,DZ & REG ,,,,R8 & DBUS SALU
     / &        CKREG & STATSHFT B"0000000000000
     ;
     ;(7) JUMP TO DSRC2 TAKES PLACE IF LS BYTE OF ER8 WAS ZERO AFTER THE SUBTRACTION
     ;  MEANING A NEGATIVE SHIFT (IE RIGHT) IS REQUIRED.
0542            NAC ,TRUE,ZLSB,CJP & ALU & DBUS SNONE & BADDRESS DSRC2
     ;
     ;(8) IN THE EVENT THAT ER8 IS STILL NONZERO, JUMP TO OVERFLOW TAKES PLACE.
0543            NAC ,,,JP & ALU & DBUS SNONE & BADDRESS OVERFLOW
     ;
     ;-------------------------------------------------------------------------
     ;
     ;
     ;-------------------------------------------------------------------------------
     ;
     ; SINGLE PRECISION LOAD
     ; ********************
     ;
     ; ENTRY POINT FOR REGISTER ADDRESSING
     ; ***********
     ;
     ;(1) THE CONTENTS OF GR2 (RB) ARE PLACED IN GR1 (RA)
0544 OPCODE81:: NAC ,FALSE,TZERO,CRTN & ALU ,B"0,,RAMF,OR,OR,ZA
     / &        REG ,,GR1,GR2 & CKALU & DBUS SNONE & CARRYSEL CZERO
     / &        STATSHFT B"0000000010000
     ;
     ; ENTRY POINT FOR ISP ADDRESSING
     ; ***********
     ;
     ;(1) JUMP SUB TAKES PLACE TO OPISP
0545 OPCODE82:: NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS OPISP
     ;
     ;(2) THE DERIVED OPERAND IS PLACED IN GR1, MACHINE STATUS IS LOADED AND
     ; RETURN TAKES PLACE.
0546            NAC ,FALSE,TZERO,CRTN & ALU ,B"0,ABUS,RAMF,OR,OR,DZ & REG ,,GR1,,R5
     / &        CKALU & DBUS SNONE & STATSHFT B"0000000010000
     / &        CARRYSEL CZERO
     ;
     ;
     ; ENTRY POINT FOR ISN ADDRESSING
```



```
AMD05/29 AMDASM MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EMSL

     ; ***********
     ;
     ;(1) JUMP SUB TO OPISN
0547 OPCODE83:: NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS OPISN
     ;
     ;(2) THE DERIVED OPERAND IN ER5 IS PLACED IN GR1, MACHINE STATUS IS LOADED
     ; AND RETURN TAKES PLACE.
0548            NAC ,FALSE,TZERO,CRTN & ALU ,B"0,ABUS,RAMF,OR,OR,DZ & REG ,,GR1,,R5
     / &        CKALU & CARRYSEL CZERO & DBUS SNONE & STATSHFT B"0000000010000
     ;
     ;
     ; ENTRY POINT FOR BASE RELATIVE ADDRESSING
     ; ***********
     ;
     ;(1) JUMP SUB TO OFB1
0549 OPCODE80:: NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS OFB1
     ;
     ;(2) THE DERIVED OPERAND IN ER5 IS PLACED IN IR2, MACHINE STATUS IS LOADED
     ; AND RETURN TAKES PLACE.
054A            NAC ,FALSE,TZERO,CRTN & ALU ,B"0,ABUS,RAMF,OR,OR,DZ
     / &        REG B"10,,RZERO,,R5 & CKALU & DBUS SNONE & CARRYSEL CZERO
     / &        STATSHFT B"0000000010000
     ;
     ;
     ; ENTRY POINT FOR BASE RELATIVE INDEXED ADDRESSING
     ; ***********
     ;
     ;(1) JUMP SUB TAKES PLACE TO OFBX1
054B XOPCD480:  NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS OFBX1
     ;
     ;(2) THE DERIVED OPERAND IS PLACED IN IR2, THE MACHINE STATUS REGISTER IS
     ; LOADED ACCORDING TO THE ALU OUTPUT, RETURN TAKES PLACE.
054C            NAC ,FALSE,TZERO,CRTN & ALU ,B"0,ABUS,RAMF,OR,OR,DZ
     / &        REG B"10,,RZERO,,R5 & CKALU & DBUS SNONE & CARRYSEL CZERO
     / &        STATSHFT B"0000000010000
     ;
     ;
     ; ENTRY POINT FOR DIRECT OR DIRECT INDEXED ADDRESSING
     ; ***********
     ;
     ;(1) JUMP SUB TO OFDX1
054D OPCODE80:: NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS OFDX1
     ;
     ;(2) THE DERIVED OPERAND IN ER5 IS LOADED INTO GR1 (RA), THE MACHINE STATUS
     ; REGISTER IS LOADED ACCORDING TO ITS VALUE. RETURN TAKES PLACE.
054E            NAC ,FALSE,TZERO,CRTN & ALU ,B"0,ABUS,RAMF,OR,OR,DZ
```

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

      / &         REG ,,GR1,,R5 & CKALU & DBUS SNONE & CARRYSEL CZERO
      / &         STATSHFT B#0000000010000
      ;
      ; ENTRY POINT FOR INDIRECT AND INDIRECT INDEXED ADDRESSING
      ; ************
      ;
      ;(1) JUMP SUB TO OFIX1
0547  OPCODE84:: NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS OFIX1
      ;
      ;(2) THE DERIVED OPERAND IN ER5 IS PLACED IN GR1, THE MACHINE STATUS
      ; REGISTER IS LOADED AND RETURN TAKES PLACE.
0550              NAC ,FALSE,TZERO,CRTN & ALU ,B#0,ABUS,RAMF,OR,OR,DZ
      / &         REG ,,GR1,,R5 & CKALU & DBUS SNONE & CARRYSEL CZERO
      / &         STATSHFT B#0000000010000
      ;
      ; ENTRY POINT FOR IMMEDIATE ADDRESSING
      ; ************
      ;(1) JUMP SUB TO FETDIADD
0551  OPCODE85:: NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS FETDIADD
      ;
      ;(2) THE DERIVED OPERAND IN AR2 IS LOADED INTO GR1, THE MACHINE STATUS REGISTER
      ; IS LOADED ACCORDING TO ITS VALUE. RETURN TAKES PLACE.
0552              NAC ,FALSE,TZERO,CRTN & ALU ,B#0,ABUS,RAMF,OR,OR,DZ
      / &         REG ,,GR1,,R2 & CKALU & DBUS SNONE & CARRYSEL CZERO
      / &         STATSHFT B#0000000010000
      ;
      ;-------------------------------------------------------------------------
      ;
      ; LOAD FROM UPPER BYTE
      ; ********************
      ;
      ; ENTRY POINT FOR DIRECT OR DIRECT INDEXED ADDRESSING
      ; ************
      ;
      ;(1) JUMP SUB TO OFDX1
0553  OPCODE8B:: NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS OFDX1
      ;
      ;(2) THE DERIVED OPERAND IN ER5 IS SWOPPED WITH THE CONTENTS OF GR1.
0554  LUB1:       NAC ,FALSE,TZERO,PUSH & ALU ,,ABUS,RAMA,OR,OR,DZ & REG ,,GR1,GR1,R5
      / &         DBUS SALU & CKALU & CKREG & DATINSRT D#8
      ;
      ;(3) THE CONTENTS OF GR1 ARE SHIFTED 8 PLACES DOWN.
```



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

0555              NAC ,FALSE,COUNT,RPCT & ALU ,,,RAMD,OR,OR,ZA
      / &         CKALU & DBUS SNONE & REG ,,GR1,GR1 & STATSHFT B#0000 0000000
      ;
      ;(3A) A MASK (1111111100000000) IS PLACED IN QREG
0556              NAC ,,,CONT & ALU ,,DBUS,QREG,OR,OR,DZ
      / &         DBUS SDATA & DATINSRT D#65280 & CKALU
      ;
      ;(4) THE CONTENTS OF ER5 (ORIGINALLY THE CONTENTS OF GR1) ARE MASKED BY
      ; THE CONTENTS OF Q REG (1111111100000000) AND PLACED IN QREG.
0557              NAC ,,,CONT & ALU ,,ABUS,QREG,AND,AND,DQ & REG ,,,,R5 & CKALU
      / &         DBUS SNONE
      ;
      ;(5) THE CONTENTS OF QREG (MS HALF OF ORIGINAL RA) ARE ORED WITH THE
      ; CONTENTS OF GR1, THE RESULT IS PLACED IN GR1 AND THE STATUS REGISTER
      ; LOADED.  RETURN TAKES PLACE.
0558              NAC ,FALSE,TZERO,CRTN & ALU ,B#0,,RAMF,OR,OR,AQ
      / &         REG ,,GR1,GR1 & CKALU & DBUS SNONE & CARRYSEL CZERO
      / &         STATSHFT B#0000000010000
      ;
      ; ENTRY POINT FOR INDIRECT AND INDIRECT INDEXED ADDRESSING
      ; ************
      ;
      ;(1) JUMP SUB TO OFIX1
0559  OPCODE8D:: NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS OFIX1
      ;
      ;(2) JUMP TO START OF SEQUENCE FOR DIRECT ADDRESSING.
055A              NAC ,,,JP & ALU & DBUS SNONE & BADDRESS LUB1
      ;
      ;-------------------------------------------------------------------------
      ;
      ; LOAD FROM LOWER BYTE
      ; ********************
      ;
      ; ENTRY POINT FOR DIRECT OR DIRECT INDEXED ADDRESSING
      ; ************
      ;
      ;(1) JUMP SUB TO OFDX1
055F  OPCODE8C:: NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS OFDX1
      ;
      ;(2) THE DERIVED OPERAND IS PLACED IN THE QREG
055C  LLB1:       NAC ,,,CONT & ALU ,,ABUS,QREG,OR,OR,DZ & REG ,,,,R5 & CKALU
      / &         DBUS SNONE
```

```
AMD05/29 AMDASM MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EMSL

     ;
     ;(3) THE CONTENTS OF THE Q REG ARE MASKED BY (0000000011111111) AND THE
     ; RESULT LEFT IN QREG.
0559           NAC ,,,CONT & ALU ,,DBUS,QREG,AND,AND,DQ & CKALU
     / &       DBUS SDATA & DATINSRT D#255
     ;
     ;(4) THE CONTENTS OF GR1 ARE MASKED BY (1111111100000000) AND THE RESULT
     ;LEFT IN GR1.
055E           NAC ,,,CONT & ALU ,,DBUS,RAMF,AND,AND,DA & CKALU
     / &       DBUS SDATA & DATINSRT D#65280
     ;
     ;(5) THE CONTENTS OF THE QREG ARE ORED WITH THE CONTENTS OF GR1 AND THE
     ; RESULT LEFT IN GR1.
     ; MACHINE STATUS IS LOADED AND RETURN TAKES PLACE.
055F           NAC ,FALSE,TZERO,CRTN & ALU ,B#0,,RAMF,OR,OR,AQ
     / &       REG ,,GR1,GR1 & CKALU & STATSHFT B#0000000010000
     / &       CARRYSEL CZERO
     ;
     ;
     ;
     ; ENTRY POINT FOR INDIRECT OR INDIRECT INDEXED ADDRESSING
     ; ++++++++++++
     ;
     ;(1) JUMP SUB TO OFIX1
0560 OPCODE9E:: NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS OFIX1
     ;
     ;(2) JUMP TO CODE FOR DIRECT ADDRESSING
0561           NAC ,,,JP & ALU & BADDRESS LLB1 & DBUS SNONE
     ;
     ;-------------------------------------------------------------------------
     ;
     ; LOAD STATUS
     ; ++++++++++++
     ;
     ; ENTRY POINT FOR INDIRECT AND INDIRECT INDEXED ADDRESSING
     ; ++++++++++++
     ;
     ;(1) JUMP SUB TO OFIX3
0562 OPCODE7C:: NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS OFIX3
     ;
     ;(2) JUMP TO LDSTA1
0563           NAC ,,,JP & ALU & DBUS SNONE & BADDRESS LDSTA1
     ;
     ; ENTRY POINT FOR DIRECT AND DIRECT INDEXED ADDRESSING
     ; ++++++++++++
     ;
```



```
AMD05/29 AMDASM MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EMSL

     ;(1) JUMP SUB TO OFDX3
0564 OPCODE7D:: NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS OFDX3
     ;
     ;(2) THE INTERRUPT MASK (DERIVED OPERAND WORD 1) IS INVERTED AND LOADED INTO QREG.
0565 LDSTA1: NAC ,,,CONT & ALU ,,ABUS,QREG,EXNOR,EXNOR,DZ & REG ,,,,R5 & DBUS SNONE
     / &       CKALU
     ;( ) THE CONTENTS OF THE QREG ARE LOADED INTO ER15, JUMP SUB TAKES PLACE TO
     ; MIMICINT.
0566           NAC ,FALSE,TZERO,CJS & ALU ,,,NOP,OR,OR,ZQ & DBUS SALU & REG ,,,,R15
     / &       CKREG & BADDRESS MIMICINT
     ;
     ;
     ;(2) THE SECOND WORD OF THE DERIVED OPERAND IS LOADED INTO THE STATUS REGISTER
0567           NAC ,,,CONT & ALU & DBUS SREG & REG ,,,,R6 & EWAUICK & AUICK D#0100
     ;
     ;(2A) THE LEAST SIGNIFICNT FOUR BITS OF THE STATUS REGISTER ARE LOADED.
0568           NAC ,,,CONT & ALU ,D#0, & REG ,,,,R6 & DBUS SREG & STATSHFT B#0000000000000
     ;
     ;
     ;(3) THE THIRD WORD OF THE DERIVED OPERAND IS LOADED FIRST INTO THE Q REG.
0569           NAC ,,,CONT & ALU ,,ABUS,QREG,OR,OR,DZ & REG ,,,,R7 & CKALU
     / &       DBUS SNONE
     ;
     ;(4) THE THIRD WORD IN THE QREG IS NOW LOADED INTO THE PROGRAM COUNTER (ER0).
     ; RETURN TAKES PLACE.
056A           NAC ,FALSE,TZERO,CRTN & ALU ,,,NOP,OR,OR,ZQ & REG ,,,,R0 & CKREG
     / &       DBUS SALU
     ;
     ;-------------------------------------------------------------------------
     ;
     ; LOAD 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 OF A DIRECT OR DIRECT INDEXED OPERAND.
     ; THE ADDRESS OF THE OPERAND IS PLACED IN ER2.
056B OPCODE89:: NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS FETDIADD
     ;
     ;(2)A MASK (0000000011110000) IA PLACED IN QREG.
056C           NAC ,,,CONT & ALU ,,DBUS,QREG,OR,OR,DZ & CKALU
     / &       DBUS SDATA & DATINSRT D#240
     ;
     ;(3) THE INSTRUCTION IN ER1 IS MASKED AND LEFT IN THE QREG.
```

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

 056D            NAC ,,,CONT & ALU ,,ABUS,QREG,AND,AND,DQ & REG ,,,,R1 & CKALU
      / &        DBUS SALU
                 ;
                 ;
                 ;(4) LOADING OF THE INSTRUCTION REGISTER IS CARRIED OUT..
                 ;  THE MASKED INSTRUCTION IS ALSO PLACED IN ER1
 056E LM1:       NACF ,,,CONT & ALU ,,,NOP,OR,OR,ZQ & DBUS SALU
      / &        CKRFG & REG ,,,,R1
                 ;(5) JUMP SUB TAKES PLACE TO OPGENPUR.
 056F            NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS OPGENPUR
                 ;
                 ;(5A) THE CONTENTS OF ER5 ARE PLACED IN GR2.
 0570            NAC ,,,CONT & ALU ,,ABUS,RAMF,OR,OR,DZ & CKALU & REG ,,GR2,,R5
      / &        DBUS SNONE
                 ;(  ) THE CONTENTS OF ER1 ARE PLACED IN QREG
 0571            NAC ,,,CONT & ALU ,,ABUS,QREG,OR,OR,DZ & CKALU & REG ,,,,R1
      / &        DBUS SNONE
                 ;
                 ;
                 ;(5B) THE CONTENTS OF THE Q REGISTER ARE MASKED BY (0000000011110000) BUT NOT
                 ; LOADED.
 0572            NAC ,,,CONT & ALU ,,DBUS,NOP,AND,AND,DQ & DBUS SDATA & DATINSRT D#240
                 ;
                 ;(6) 1 IS ADDED TO THE CONTENTS OF ER2. RETURN TAKES PLACE IF ZLSB WAS SET BY
                 ;   THE PREVIOUS INSTRUCTION.
 0573            NAC ,TRUE,ZLSB,CRTN & ALU ,,ABUS,NOP,ADD,ADD,DZ & REG ,,,,R2 & CKREG
      / &        DBUS SALU & STATSHFT B#0100000000000
                 ;
                 ;(7) 15 IS SUBTRACTED FROM THE CONTENTS OF QREG.
                 ;  THIS MEANS THAT THE N FIELD IS DECREMENTED AND THE GR2 FIELD INCREMENTED.
 0574            NAC ,,,CONT & ALU ,,DBUS,QREG,SUBR,SUBR,DQ & CKALU
      / &        DBUS SDATA & DATINSRT D#14
                 ;
                 ;(8) JUMP TAKES PLACE TO LM1.
 0575            NAC ,,,JP & ALU & DBUS SNONE & BADDRESS LM1
                 ;
                 ;-------------------------------------------------------------------------------
                 ;
                 ;
                 ; EXCHANGE BYTES IN REGISTER
                 ; **************************
                 ;
                 ; ENTRY POINT
                 ; ***********
```



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

                 ;
 0576 OPCODEEC:: NAC ,FALSF,TZERO,PUSH & ALU & DBUS SNONE
      / &        DATINSRT D#6
                 ;
                 ;(2) THE CONTENTS OF GR1 ARE SHIFTED 8 PLACES DOWN CYCLIC
 0577            NAC ,FALSE,COUNT,RFCT & ALU ,,,RAMD,OR,OR,ZA & CKALU & DBUS SNONE
      / &        REG ,,GR1,GR1 & STATSHFT B#0010100000000
                 ;
                 ;(3) THE MACHINE STATUS REGISTER IS LOADED AND RETURN TAKES PLACE
 0578            NAC ,FALSE,TZERO,CRTN & ALU ,B#0,,NOP,OR,OR,ZA & REG ,,GR1,GR1
      / &        STATSHFT B#0000000010000 & CARRYSEL CZERO
                 ;-------------------------------------------------------------------------------
                 ;
                 ; EXCHANGE WORDS IN REGISTER
                 ; **************************
                 ;
                 ; ENTRY POINT
                 ; ***********
                 ;
                 ;(1) THE CONTENTS OF RB (GR2) ARE PLACED TEMPORARILY IN Q REG
 0579 OPCODEED:: NAC ,,,CONT & ALU ,,,QREG,OR,OR,ZA & REG ,,,GR2 & CKALU & DBUS SNONE
                 ;
                 ;(2) THE CONTENTS OF RA (GR1 ARE PLACED IN RB (GR2).
 057A            NAC ,,,CONT & ALU ,,,RAMF,OR,OR,ZA & REG ,,GR2,GR1 & CKALU & DBUS SNONE
                 ;
                 ;(3) THE CONTENTS OF THE QREG ARE PLACED IN RA (GR1) THE STATUS REGISTER
                 ; IS LOADED AND RETURN TAKES PLACE.
 057B            NAC ,FALSE,TZERO,CRTN & ALU ,B#0,,RAMF,OR,OR,ZQ & REG ,,GR1 & CKALU
      / &        DBUS SNONE & STATSHFT B#0000000010000 & CARRYSEL CZERO
                 ;-------------------------------------------------------------------------------
                 ;
                 ;
                 ;
                 ; SINGLE PRECISION STORE
                 ; **********************
                 ;
                 ; ENTRY POINT FOR BASE RELATIVE ADDRESSING
 057C OPCODE82:: NAC ,,,CONT & ALU ,,,NOP,OR,OR,ZA & REG ,B#10,,RZERO,R5 & CKREG
      / &        DBUS SALU
                 ;
                 ;(2) JUMP TO APPROPRIATE OPERAND SEND ROUTINE
 057D            NAC ,,,JP & ALU & DBUS SNONE & BADDRESS OSB1
                 ;
```

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

      ; ENTRY POINT FOR BASE RELATIVE INDEXED ADDRESSING
      ; ++++++++++++
057E EOPC0402:  NAC ,,,CONT & ALU ,,,NOP,OR,OR,ZA & REG ,B#10,,RZERO,R5 & CKREG
      / &         DBUS SALU
      ;
      ;(2) JUMP TO APPROPRIATE OPERAND SEND ROUTINE
057F              NAC ,,,JP & ALU & DBUS SNONE & BADDRESS OSBX1
      ;
      ;
      ; ENTRY POINT FOR DIRECT AND DIRECT INDEXED ADDRESSING
      ; ++++++++++++
      ;
0580 OPCODE90:: NAC ,,,CONT & ALU ,,,NOP,OR,OR,ZA & REG ,,,GR1,R5 & CKREG
      / &         DBUS SALU
      ;
      ;(2) JUMP TO OPERAND SEND ROUTINE.
0581              NAC ,,,JP & ALU & DBUS SNONE & BADDRESS OSDX1
      ;
      ;
      ; ENTRY POINT FOR INDIRECT AND INDIRECT INDEXED ADDRESSING
0582 OPCODE94:: NAC ,,,CONT & ALU ,,,NOP,OR,OR,ZA & REG ,,,GR1,R5 & CKREG
      / &         DBUS SALU
      ;
      ;(2) JUMP TO APPROPRIATE OPERAND SEND ROUTINE
0583              NAC ,,,JP & ALU & DBUS SNONE & BADDRESS OSIX1
      ;
      ;
      ;-------------------------------------------------------------------------
      ;
      ; STORE A NON NEGATIVE CONSTANT
      ; ****************************
      ;
      ; ENTRY POINT FOR DIRECT AND DIRECT INDEXED ADDRESSING
      ; ++++++++++++
      ;
      ;(1) JUMP SUB TO STC1 -- STC1 TAKES THE CONSTANT FROM THE FIELD WITHIN THE
      ; INSTRUCTION AND PLACES IT IN ER5
0584 OPCODE91:: NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS STC1
      ;
      ;(2) JUMP TO THE APPROPRIATE OPERAND SEND ROUTINE (OSDX1)
0585              NAC ,,,JP & DBUS SNONE & ALU & BADDRESS OSDX1
      ;
      ;
      ; ENTRY POINT FOR INDIRECT AND INDIRECT INDEXED ADDRESSING
      ; ++++++++++++
      ;
```



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

      ;(1) JUMP SUB TO STC1
0586 OPCODE92:: NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS STC1
      ;
      ;(2) JUMP TO APPROPRIATE OPERAND SEND ROUTINE. (OSIX1)
0587              NAC ,,,JP & ALU & DBUS SNONE & BADDRESS OSIX1
      ;
      ;
      ;
      ;(1) THE CONTENTS OF IR0 ARE PLACED IN ER5
0588 STC1:        NAC ,,,CONT & ALU ,,,NOP,OR,OR,ZA & REG ,,,RZERO,R5 & CKREG
      / &         DBUS SALU
      ;
      ;(2) THE INSTRUCTION IN ER1 IS SHIFTED DOWN AND PLACED IN IR0
0589              NAC ,,,CONT & ALU ,,ABUS,RAMD,OR,OR,DZ & REG ,,RZERO,,R1 & CKALU
      / &         DBUS SNONE & STATSHFT B#0000000000000
      ;
      ;(3) THE CONTENTS OF IR0 ARE DOWNSHIFTED
058A              NAC ,,,CONT & ALU ,,,RAMD,OR,OR,ZA & CKALU & REG ,,RZERO,RZERO
      / &         DBUS SNONE & STATSHFT B#2000000000000
      ;
      ;(4) THE CONTENTS OF IR0 ARE DOWNSHIFTED
058B              NAC ,,,CONT & ALU ,,,RAMD,OR,OR,ZA & CKALU & REG ,,RZERO,RZERO
      / &         DBUS SNONE & STATSHFT B#2000000000000
      ;
      ;(5) THE CONTENTS OF IR0 ARE MASKED BY 0000000000011111 AND DOWNSHIFTED.
058C              NAC ,,,CONT & ALU ,,DBUS,RAMD,AND,AND,DA & CKALU & REG ,,RZERO,RZERO
      / &         DBUS SDATA & DATINSRT D#31
      ;
      ;(6) THE CONTENTS OF IR0 ARE SWOPPED WITH THE CONTENTS OF ER5.  RETURN
      ;  TAKES PLACE.
058D              NAC ,FALSE,TZERO,CRTN & ALU ,,ABUS,RAMA,OR,OR,ZA & CKALU & CKREG
      / &         REG ,,RZERO,RZERO,R5 & DBUS SALU
      ;
      ;-------------------------------------------------------------------------
      ;
      ;
      ; STORE REGISTER THROUGH MASK
      ; ***************************
      ;
      ; ENTRY POINT FOR DIRECT AND DIRECT INDEXED ADDRESSING
      ; ++++++++++++
      ;
      ; (1) THE CONTENTS OF REGISTER GR1 (RA) ARE ANDED WITH THE CONTENTS OF
      ; REGISTER GR1+1 AND THE RESULTS PLACED IN ER5.
058E OPCODE97:: NAC ,,,JP & ALU ,,,NOP,AND,AND,AB & REG ,B#01,GR1,GR1,R5
```

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

      / &         DBUS SALU & CKREG & BADDRESS OSIX1
      ;
      ;-------------------------------------------------------------------------------
      ;
      ; STORE INTO UPPER BYTE
      ; ********************
      ;
      ; ENTRY POINT FOR DIRECT OR DIRECT INDEXED ADDRESSING
      ; ***********
      ;
      ;(1) JUMP TO OPERAND FETCH ROUTINE FOR DIRECT OR DIRECT INDEXED ADDRESSING
@58F OPCODE9B:: NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS OFDX1
      ; SUBROUTINE TO CARRY OUT THE NECESSARY SHIFTING AND MASKING COMMON TO BOTH
      ; ADDRESSING MODES.
      ;
      ;(2) JUMP SUB TO STUB1
@590          NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS STUB1
      ;
      ;(3) THE RESULT IN ER5 IS PLACED IN MEMORY AT THE SAME ADDRESS AS
      ; IT CAME FROM USING OSGENPUR.
@591          NAC ,,,JP & ALU & BADDRESS OSGENPUR
      ;
      ;
      ; ENTRY POINT FOR INDIRECT AND INDIRECT INDEXED ADDRESSING
      ; ***********
      ;
      ;(1) JUMP SUB TO OFIX1
@592 OPCODE9D:: NAC ,FALSE,TZERO,CRTN & ALU & DBUS SNONE & BADDRESS OFIX1
      ;
      ;(2) JUMP SUB TO STUB1
@593          NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS STUB1
      ;
      ;(3) JUMP TO OSGENPUR
@594          NAC ,,,JP & ALU & DBUS SNONE & BADDRESS OSGENPUR
      ;
      ;
      ;(1) A MASK (0000000011111111) IS PLACED IN QREG.
@595 STUB1:   NAC ,,,CONT & ALU ,,DBUS,QREG,OR,OR,DZ & CKALU & DBUS SDATA
      / &         DATINSRT D#255
      ;
      ;(3) THE DERIVED OPERAND IN ER5 IS MASKED AND LEFT IN THE QREG.
@596          NAC ,,,CONT & ALU ,,ABUS,QREG,AND,AND,DQ & REG ,,,,R5 & CKALU
      / &         DBUS SNONE
      ;
```



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

      ;(4) THE CONTENTS OF GR1 ARE PLACED IN ER5
      ;  THE NEXT LOCATION IS PUSHED ONTO THE STACK AND THE COUNTER IS LOADED WITH
      ;    THE VALUE 6.
@597          NAC ,FALSE,TZERO,PUSH & ALU ,,,NOP,OR,OR,ZA & REG ,,,GR1,R5 & CKREG
      / &         DBUS SALU & DATINSRT D#6
      ;
      ;(5) THE CONTENTS OF GR1 ARE SHIFTED UPWARDS 8 TIMES
@598          NAC ,FALSE,COUNT,RPCT & ALU ,,,RAMD,OR,OR,ZA & REG ,,GR1,GR1
      / &         CKALU & DBUS SNONE & STATSHFT B"0000010000000
      ;
      ;(6) THE CONTENTS OF THE QREG ARE ORED WITH THE CONTENTS OF GR1 AND THE
      ; RESULT LEFT IN GR1.
@599          NAC ,,,CONT & ALU ,,,RAMF,OP,OR,AQ & REG ,,GR1,GR1 & CKALU
      / &         DBUS SNONE
      ;
      ;(7) THE CONTENTS OF GR1 ARE SWOPPED WITH ER5.  RETURN TAKES PLACE.
@59A          NAC ,FALSE,TZERO,CRTN & ALU ,,ABUS,RAMA,OR,OR,DZ & CKALU & CKREG
      / &         REG ,,GR1,GR1,R5 & DBUS SALU
      ;
      ;-------------------------------------------------------------------------------
      ;
      ;
      ; STORE INTO LOWER BYTE
      ; ********************
      ;
      ; ENTRY POINT FOR DIRECT OR DIRECT INDEXED ADDRESSING
      ; ***********
      ;
      ;(1) JUMP SUB TO OFDX1
@59B OPCODE9C:: NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS OFDX1
      ;
      ;(2) JUMP SUB TO SLB1
@59C          NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS SLB1
      ;
      ;(3) JUMP TO OSGENPUR
@59D          NAC ,,,JP & ALU & DBUS SNONE & BADDRESS OSGENPUR
      ;
      ;
      ;(1) A MASK (1111111100000000) IS PLACED IN QREG.
@59E SLB1:    NAC ,,,CONT & ALU ,,DBUS,QREG,OR,OR,DZ & DBUS SDATA & CKALU
      / &         DATINSRT D#6528
      ;
      ;(2) THE DERIVED OPERAND IN ER5 IS MASKED AND THE RESULT LEFT IN ER5.
@59F          NAC ,,,CONT & ALU ,,ABUS,NOP,AND,AND,DQ & REG ,,,R5 & CKREG
      / &         DBUS SALU
```

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

      ;
      ;(3) THE CONTENTS OF GR1 (RA) ARE MASKED BY (0000000011111111) AND PLACED
      ; IN QREG
05A0            NAC ,,,CONT & ALU ,,DBUS,QREG,AND,AND,DA & REG ,,,GR1 & CKALU
      / &       DBUS SDATA & DATINSRT D#255
      ;
      ;(4) THE CONTENTS OF THE QREG ARE ORED WITH THE CONTENTS OF ER5 AND THE
      ; RESULT LEFT IN ER5.  RETURN TAKES PLACE.
05A1            NAC ,FALSE,TZERO,CRTN & ALU ,,ABUS,NOP,OR,OR,DQ & REG ,,,,R5 & CKREG
      / &       DBUS SALU
      ;
      ;
      ; ENTRY POINT FOR INDIRECT OR INDIRECT INDEXED ADDRESSING
      ; ++++++++++++
      ;
      ;(1) JUMP SUB TO OFIX1
05A2  OPCODE9E:: NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS OFIX1
      ;
      ;(2) JUMP SUB TO SLB1
05A3            NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS SLB1
      ;
      ;(3) JUMP TO OSGENPUR
05A4            NAC ,,,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 OF A DIRECT OR DIRECT INDEXED OPERAND.
      ; THE ADDRESS OF THE OPERAND IS PLACED IN ER2.
05A5  OPCODE99:: NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS FETDIADD
      ;
      ;(2)A MASK (0000000011110000) IA PLACED IN QRE3.
05A6            NAC ,,,CONT & ALU ,,DBUS,QREG,OR,OR,DZ & CKALU
      / &       DBUS SDATA & DATINSRT D#240
      ;
      ;(3) THE INSTRUCTION IN ER1 IS MASKED AND LEFT IN THE QREG.
05A7            NAC ,,,CONT & ALU ,,ABUS,QREG,AND,AND,DQ & REG ,,,,R1 & CKALU
      / &       DBUS SALU
      ;
```



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

      ;
      ;(4) LOADING OF THE INSTRUCTION REGISTER TAKES PLACE.
05A8  STM1:     NACF ,,,CONT & ALU ,,,NOP,OR,OR,ZQ & DBUS SALU
      / &       CKREG & REG ,,,,R1
      ;
      ;(5) THE CONTENTS OF GR2 IS PLACED IN ER5. JUMP SUB TAKES PLACE TO
      ;  OSGENPUR.
05A9            NAC ,FALSE,TZERO,CJS & ALU ,,,NOP,OR,OR,ZA & REG ,,,GR2,R5 & DBUS SALU
      / &       CKREG & BADDRESS OSGENPUR
      ;
      ;(5A) THE CONTENTS OF ER1 ARE PLACED IN THE QREG.
05AA            NAC ,,,CONT & ALU ,,ABUS,QREG,OR,OR,DZ & CKALU & REG ,,,,R1
      / &       DBUS SNONE
      ;
      ;(5A) THE CONTENTS OF THE QREG ARE MASKED BY (0000000011110000) BUT NOT LOADED.
05AB            NAC ,,,CONT & ALU ,,DBUS,NOP,AND,AND,DQ & DBUS SDATA & DATINSRT D#240
      ;
      ;
      ;(6) 1 IS ADDED TO THE CONTENTS OF ER2. RETURN TAKES PLACE IF
      ;  THE PREVIOUS OPERATION SET ZLSB.
05AC            NAC ,TRUE,ZLSB,CRTN & ALU ,,ABUS,NOP,ADD,ADD,DZ & REG ,,,,R2 & CKREG
      / &       DBUS SALU & STATSHFT B#0100000000000
      ;
      ;(7) 15 IS SUBTRACTED FROM THE CONTENTS OF QREG.
      ;  THIS MEANS THAT THE N FIELD IS DECREMENTED AND THE GR2 FIELD INCREMENTED.
05AD            NAC ,,,CONT & ALU ,,DBUS,QREG,SUBR,SUBR,DQ & CKALU
      / &       DBUS SDATA & DATINSRT D#14
      ;
      ;(8) JUMP TAKES PLACE TO STM1
05AE            NAC ,,,JP & ALU & DBUS SNONE & BADDRESS STM1
      ;
      ;
      ; MOVE MULTIPLE WORDS, MEMORY TO MEMORY.
      ; **************************************
      ;
      ; ENTRY POINT
      ; ++++++++++++
      ;
      ;(1) THE CONTENTS OF RA+1 ARE PASSED THROUGH THE ALU
05AF  OPCODE93:: NAC ,,,CONT & ALU B#0,,,NOP,OR,OR,ZA & DBUS SNONE
      / &         STATSHFT B#0000000010000 & REG ,,B#01,,GR1
      ;
      ;(2) RETURN TAKES PLACE IF RA WAS ZERO IN THE PREVIOUS INSTRUCTION.
05B0  MOV1:     NAC ,TRUE,MUX,CRTN & ALU & DBUS SNONE & STATSHFT B#0000000000100
      ;
```

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

      ;(3) THE CONTENTS OF RB IS PLACED IN ER2
      ; JUMP SUB TAKES PLACE TO OPGENPUR
05B1              NAC ,FALSE,TZERO,CJS & ALU ,,,NOP,OR,OR,ZA & DBUS SALU & CKREG
      / &         REG ,,GR2,R2 & BADDRESS OPGENPUR
      ;
      ;(4) THE CONTENTS OF RA IS PLACED IN ER2, JUMP SUB TO OSGENPUR
05B2              NAC ,FALSE,TZERO,CJS & ALU ,,,NOP,OR,OR,ZA & DBUS SALU & CKREG
      / &         REG ,,,GR1,R2 & BADDRESS OSGENPUR
      ;
      ;(5) THE CONTENTS OF RA (GR1) ARE INCREMENTED.
05B3              NAC ,,,CONT & ALU ,,,RAMF,ADD,ADD,ZA & REG ,,GR1,GR1 & CKALU
      / &         DBUS SNONE & STATSHFT B#010000000000
      ;
      ;(6) THE CONTENTS OF RB (GR2) ARE INCREMENTED.
05B4              NAC ,,,CONT & ALU ,,,RAMF,ADD,ADD,ZA & REG ,,GR2,GR2 & CKALU
      / &         DBUS SNONE & STATSHFT B#010000000000
      ;
      ;(7) THE CONTENTS OF RA+1 ARE DECREMENTED.  THE MICROSTATUS IS LOADED
      ; ACCORDINGLY.
05B5              NAC ,,,CONT & ALU B#0,,,RAMF,SUBR,SUBR,ZA & REG B#01,B#01,GR1,GR1
      / &         DBUS SNONE & CKALU & STATSHFT B#000000010000
      ;
      ;(8) JUMP TO MOV1 IF NO MICRO INTERRUPT
05B6              NAC ,TRUE,INTRPT,CJP & ALU & DBUS SNONE & BADDRESS MOV1
      ;
      ;(9) THE PC IS DECREMENTED AND JUMP TAKES PLACE TO MICINT SERVICE SEQUENCE.
05B7              NAC ,FALSE,TZERO,CJV & MICINT RDVC & CKMICINT
      / &         ALU ,,ABUS,NOP,SUBS,SUBS,DZ & REG ,,,,R0 & DBUS SALU & CKREG
      / &         STATSHFT B#0000000000000
      ;
      ;-------------------------------------------------------------------------
      ;
      ;-------------------------------------------------------------------------
      ;
      ; PUSH MULTIPLE REGISTERS ONTO STACK
      ; ##################################
      ;
      ;(1) THE INSTRUCTION IN ER1 IS LOADED INTO THE QREG.
05B8 OPCODE9F:: NAC ,,,CONT & ALU ,,ABUS,QREG,OR,OR,DZ & REG ,,,,R1 & CKALU
      ;
      ;(2) THE INSTRUCTION IN ER1 IS SWOPPED WITH THE CONTENTS OF IR0.
05B9              NAC ,FALSE,TZERO,PUSH & ALU ,,ABUS,RAMA,OR,OR,DZ & REG ,,RZERO,RZERO,R1
      / &         CKALU & CKREG & DBUS SALU & DATINSRT D#1
      ;
      ;(3) THE CONTENTS OF IR0 ARE DOWNSHIFTED 3 TIMES.
05BA              NAC ,FALSE,COUNT,RPCT & ALU ,,,RAMD,OR,OR,ZA & REG ,,RZERO,RZERO
```



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

      / &         CKALU & DBUS SNONE & STATSHFT B#0000000000000
      ;
      ;(3A) THE CONTENTS OF IR0 ARE MASKED WITH 0000000000011111 AND DOWNSHIFTED.
05BB              NAC ,,,CONT & ALU ,,DBUS,RAMD,AND,AND,DA & REG ,,RZERO,RZERO
      / &         CKALU & DBUS SDATA & DATINSRT D#31
      ;
      ;(4) THE CONTENTS OF IR0 ARE SWOPPED WITH ER1.
05BC              NAC ,,,CONT & ALU ,,ABUS,RAMA,OR,OR,DZ & REG ,,RZERO,RZERO,R1
      / &         CKALU & CKREG & DBUS SALU
      ;
      ;(5) THE CONTENTS OF ER1 ARE ADDED TO THE CONTENTS OF QREG AND THE RESULT
      ; LEFT IN QREG.
05BD              NAC ,,,CONT & ALU ,,ABUS,QREG,ADD,ADD,DQ & REG ,,,,R1 & CKALU
      / &         DBUS SNONE & STATSHFT B#0000000000000
      ;
      ;(7) THE CONTENTS OF QREG ARE PLACED IN ER8
05BE              NAC ,,,CONT & ALU ,,,NOP,OR,OR,ZQ & REG ,,,,R8 & DBUS SALU & CKREG
      ;
      ;(8) A MASK 0000000011110000 IS LOADED INTO QREG.
05BF PUSHM1:      NAC ,,,CONT & ALU ,,DBUS,QREG,OR,OR,DZ & CKALU & DBUS SDATA
      / &         DATINSRT D#240
      ;
      ;(9) THE CONTENTS OF ER8 ARE ORED WITH THE QREG AND THE RESULT LOADED INTO
      ; THE INSTRUCTION REGISTER.
05C0              NACF ,,,CONT & ALU ,,ABUS,NOP,OR,OR,DQ
      / &         DBUS SALU & REG ,,,,R8
      ;
      ;(10) IR15 (GR1) IS DECREMENTED AND THE RESULT PLACED IN IR15 AND ER2.
05C1              NAC ,,,CONT & ALU ,,,RAMF,SUBR,SUBR,ZA & REG ,,GR1,GR1,R2
      / &         DBUS SALU & CKALU & CKREG
      / &         STATSHFT B#0000000000000
      ;
      ;(11) THE CONTENTS OF GR2 ARE PLACED IN ER5, JUMP TAKES PLACE TO OSGENPUR.
05C2              NAC ,FALSE,TZERO,CJS & ALU ,,,NOP,OR,OR,ZA & REG ,,GR2,GR2,R5
      / &         DBUS SALU & CKREG & BADDRESS OSGENPUR
      ;
      ;(12) THE CONTENTS OF ER1 ARE DECREMENTED.
05C3              NAC ,,,CONT & ALU B#0,,ABUS,NOP,SUBS,SUBS,DZ & REG ,,,,R1
      / &         STATSHFT B#0000000010000 & DBUS SALU & CKREG
      ;
      ;(13) RETURN TAKES PLACE IF THE LAST OPERATION SET MICRO-N.  THE CONTENTS
      ; OF ER8 ARE DECREMENTED.
05C4              NAC ,TRUE,MUX,CRTN & ALU ,,ABUS,NOP,SUBS,SUBS,DZ & REG ,,,,R8
      / &         STATSHFT B#0000000011110 & DBUS SALU & CKREG
      ;
      ;(14) JUMP TO PUSHM1 TAKES PLACE
```

```
05C5            MAC ,,,JP & ALU & DBUS SNONE & BADDRESS PUSHM1
                ;
                ;
                ;-------------------------------------------------------------------------
                ;
                ; POP MULTIPLE REGISTERS OFF STACK
                ; ********************************
                ;
                ;(1) THE NUMBER 15 IS LOADED INTO THE INSTRUCTION REGISTER
05C6 OPCODE8F:: MACF ,,,CONT & ALU & DBUS SDATA & DATINSRT D#15
                ;
                ;(3) THE CONTENTS OF IR15 (THE STACK POINTER) ARE PLACED
                ;   IN ER2.
05C7            MAC ,,,CONT & ALU ,,,NOP,OR,OR,ZA & REG ,,,GR2,R2 & CKREG
                / &         DBUS SALU
                ;
                ;(4) THE CONTENTS OF ER1 ARE SHIFTED ONE PLACE RIGHT AND PLACED IN IR15.
05C8            MAC ,,,CONT & ALU ,,ABUS,RAMD,OR,OR,DZ & REG ,,GR2,,R1 & CKALU
                / &         DBUS SNONE & STATSHFT B#0000000000000
                ;
                ;(5) THE CONTENTS OF IR15 ARE SHIFTED ONE PLACE RIGHT.
05C9            MAC ,,,CONT & ALU ,,,RAMD,OR,OR,ZA & REG ,,GR2,GR2 & CKALU
                / &         DBUS SNONE & STATSHFT B#0000000000000
                ;
                ;(6) THE CONTENTS OF IR15 ARE SHIFTED ONE PLACE RIGHT.
05CA            MAC ,,,CONT & ALU ,,,RAMD,OR,OR,ZA & REG ,,GR2,GR2 & CKALU
                / &         DBUS SNONE & STATSHFT B#0000000000000
                ;
                ;(7) THE CONTENTS OF IR15 ARE MASKED BY 0000000000011111 AND SHIFTED ONE
                ;    PLACE RIGHT.  THE VALUE OF N NOW OCCUPIES THE LEAST SIG FOUR BITS
                ;    OF THE WORD, THE REST BEING ZERO.
05CB            MAC ,,,CONT & ALU ,,DBUS,RAMD,AND,AND,DA & REG ,,GR2,GR2 & CKALU
                / &         DBUS SDATA & DATINSRT D#31
                ;
                ;(8) THE CONTENTS OF IR15 IS LOADED INTO ER6.
05CC            MAC ,,,CONT & ALU ,,,NOP,OR,OR,ZA & REG ,,,GR2,R6 & CKREG & DBUS SALU
                ;
                ;(10) THE LOADING OF THE INSTRUCTION REGISTER TAKES PLACE. JUMP SUB TO
                ;   OPGENPUR.
05CD POPM1:     MACF ,FALSE,TZERO,CJS & ALU & REG ,,,,R1 & DBUS SREG
                / &         BADDRESS OPGENPUR
                ;
                ;(12) THE CONTENTS OF ER5 ARE PLACED IN GR2.
05CE            MAC ,,,CONT & ALU ,,ABUS,RAMF,OR,OR,DZ & REG ,,GR2,,R5 & CKALU
                / &         DBUS SNONE
```



```
                ;
                ;(13) THE CONTENTS OF ER6 ARE PASSED THROUGH THE ALU.
05CF            MAC ,,,CONT & ALU ,,ABUS,NOP,OR,OR,DZ & REG ,,,,R6 & DBUS SNONE
                ;
                ;(14) JUMP TO POPM2 TAKES PLACE IF THE PREVIOUS OPERATION GAVE ZLSB.
05D0            MAC ,TRUE,ZLSB,CJP & ALU & DBUS SNONE & BADDRESS POPM2
                ;
                ;(15) THE CONTENTS OF ER6 ARE DECREMENTED.
05D1            MAC ,,,CONT & ALU ,,ABUS,NOP,SUBS,SUBS,DZ & DBUS SALU & REG ,,,,R6
                / &         CKREG & STATSHFT B#0000000000000
                ;
                ;(16) THE CONTENTS OF ER1 ARE INCREMENTED.
05D2            MAC ,,,CONT & ALU ,,ABUS,NOP,ADD,ADD,DZ & DBUS SALU & REG ,,,,R1
                / &         CKREG & STATSHFT B#0100000000000
                ;
                ;(17) THE CONTENTS OF ER2 ARE INCREMENTED.
05D3            MAC ,,,CONT & ALU ,,ABUS,NOP,ADD,ADD,DZ & DBUS SALU & REG ,,,,R2
                / &         CKREG & STATSHFT B#0100000000000
                ;
                ;(18) JUMP TAKES PLACE TO POPM1.
05D4            MAC ,,,JP & ALU & DBUS SNONE & BADDRESS POPM1
                ;
                ;
                ;(12) THE INSTRUCTION REGISTER IS LOADED WITH 15.
05D5 POPM2:     MACF ,,,CONT & ALU & DBUS SDATA & DATINSRT D#15
                ;
                ;(14) THE CONTENTS OF ER2 ARE INCREMENTED AND PLACED IN IR15, RETURN TAKES
                ; PLACE.
05D6            MAC ,FALSE,TZERO,CRTN & ALU ,,ABUS,RAMF,ADD,ADD,DZ & REG ,,GR2,,R2
                / &         CKALU & DBUS SNONE & STATSHFT B#0100000000000
                ;
                ;
                ;-------------------------------------------------------------------------
                ;
                ; STACK IC AND JUMP TO SUBROUTINE
                ; *******************************
                ;
                ; ENTRY POINT
                ; ***********
                ;
                ;(1)JUMP TAKES PLACE TO FETDIADD.
05D7 OPCODE7E:: MAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS FETDIADD
                ;
                ;(2) THE CONTENTS OF GR1 (RA) ARE DECREMENTED AND LEFT IN GR1.
05D8            MAC ,,,CONT & ALU ,,,RAMF,SUBR,SUBR,ZA & CKALU & REG ,,GR1,GR1
                / &         DBUS SNONE & STATSHFT B#0000000000000
```

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

      ;
      ;(3) THE CONTENTS OF GR1 ARE SWOPPED WITH THE CONTENTS OF ER2.
05B9            MAC ,,,CONT & ALU ,,ABUS,RAMA,OR,OR,DZ & REG ,,GR1,GR1,R2 & CKALU
      / &       CEREG & DBUS SALU
      ;
      ;(4) THE CONTENTS OF ER0 (PC)ARE SWOPPED WITH THE CONTENTS OF GR1.
05BA            MAC ,,,CONT & ALU ,,ABUS,RAMA,OR,OR,DZ & REG ,,GR1,GR1,R0 & CKALU
      / &       CEREG & DBUS SALU
      ;
      ;(5) THE CONTENTS OF GR1 ARE PLACED IN ER5 (THE PC CONTENTS) SO THAT THEY
      ;CAN BE STORED AT THE NEW STACK ADDRESS.
05BB            MAC ,FALSE,TZERO,CJS & ALU ,,,NOP,OR,OR,ZA & REG ,,,GR1,R5 & CEREG
      / &       DBUS SALU & BADDRESS OPGENPUR
      ;
      ;(6) THE CONTENTS OF ER2 (NEW STACK POINTER) AND PLACED IN GR1, RETURN TAKES
      ;  PLACE.
05BC            MAC ,FALSE,TZERO,CRTN & ALU ,,ABUS,RAMF,OR,OR,DZ & REG ,,GR1,,R2
      / &       CKALU & DBUS SNONE
      ;
      ;
      ;-----------------------------------------------------------------------
      ;
      ; UNSTACK IC AND RETURN FROM SUBROUTINE
      ; ************************************
      ;
      ; ENTRY POINT
      ; ***********
      ;
      ;(1) THE CONTENTS OF GR2 ARE PLACED IN ER2.
      ; THE INCREMENTED CONTENTS ARE ALSO RETURNED TO GR2 (RA)
      ; JUMP SUB TAKES PLACE TO OPGENPUR WHICH WILL FETCH THE OPERAND POINTED TO
      ; BY THE STACK POINTER TO ER5.
05BD OPCODE7F:: MAC ,,,CONT & ALU ,,,RAMA,ADD,ADD,ZA & REG ,,GR1,GR1,R2
      / &       CKALU & CEREG & DBUS SALU & STATSHFT B#0010000000000
      ;
      ;(2) JUMP SUB TAKES PLACE TO OPGENPUR WHICH FETCHES THE CONTENTS OF THE
      ; MEMORY LOCATION POINTED TO BY THE STACK POINTER (ORIGINAL) NOW IN ER2.
      ; THE MEMORY DATA WILL BE PLACED IN ER5.
05BE            MAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS OPGENPUR
      ;
      ;(3) THE QREGISTER IS LOADED WITH THE CONTENTS OF ER5
05BF            MAC ,,,CONT & ALU ,,ABUS,QREG,OR,OR,DZ & CKALU & REG ,,,,R5
      / &       DBUS SNONE
      ;
      ;(4) THE CONTENTS OF THE QREG (THE NEW PROGRAM COUNTER) IS PLACED IN ER0
05C0            MAC ,FALSE,TZERO,CRTN & ALU ,,,NOP,OR,OR,ZQ & DBUS SALU & CEREG
```



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

      / &       REG ,,,,R0
      ;
      ;-----------------------------------------------------------------------
      ;
      ;
      ;
      ;-----------------------------------------------------------------------
      ;-----------------------------------------------------------------------
      ;
      ; EMULATION SEQUENCES FOR SINGLE BIT OPERATIONS.
      ;
      ;
      ; SELBIT -  SELECTS THE BIT FOR OPERATION BY USING THE GR1 FIELD OF THE
      ; INSTRUCTION WORD AND PRODUCING A MASK IN ER0 WHICH HAS ONLY THE RELEVANT
      ; BIT SET TO ONE.  THE BITS ARE NUMBERED FROM THE MOST SIGNIFICANT END - 0 TO 15
      ;
      ;(1) THE CONTENTS OF IR 0 ARE PLACED TEMPORARILY IN ER0.
05E1 SELBIT:  MAC ,,,CONT & ALU ,,,NOP,OR,OR,ZA & REG ,,RZERO,RZERO,R0 & CEREG
      / &       DBUS SALU
      ;(1A) ER10 IS LOADED WITH 16.
05E2            MAC ,,,CONT & ALU & DBUS SDATA & DATINSRT D#16 & REG ,,,,R10 & CEREG
      ;
      ;
      ;(2) IR0 IS LOADED WITH A ONE IN ITS LEAST SIGNIFICANT BIT AND 0 EVERYWHERE
      ; ELSE.
05E3            MAC ,,,CONT & ALU ,,DBUS,RAMF,OR,OR,DZ & DBUS SDATA & DATINSRT D#1
      / &       CKALU & REG ,,RZERO,RZERO
      ;
      ;(3) A MASK B#0000000111100000 IS PLACED IN THE Q REG.
05E4            MAC ,,,CONT & ALU ,,DBUS,QREG,OR,OR,DZ & DBUS SDATA & DATINSRT D#240
      / &       CKALU
      ;
      ;(4) THE INSTRUCTION IN ER1 IS MASKED AND LEFT IN Q REG
      ; THE NEXT ADDRESS IS PUSHED ONTO THE MICRO-SEQUENCER STACK. THE COUNTER IS
      ; NOT LOADED.   THE MICROSTATUS REGISTER IS LOADED FROM THE ALU.
05E5            MAC ,TRUE,TZERO,PUSH & ALU B#0,,ABUS,QREG,AND,AND,DQ & DBUS SNONE
      / &       REG ,,,,R1
      / &       CKALU & STATSHFT B#0000000010000 & CARRYSEL CZERO
      ;
      ;(5) THE CONTENTS OF IR0 ARE SHIFTED DOWN CYCLIC.
05E6            MAC ,,,CONT & ALU ,,,RAMB,OR,OR,ZA & REG ,,RZERO,RZERO & CKALU
      / &       DBUS SNONE & STATSHFT B#0010100000000
      ;
      ; 16 IS SUBTRACTED FROM THE CONTENTS OF THE QREGISTER.
      ; EXIT FROM THE LOOP TAKES PLACE IF
      ; IF MICROSTATUS 2 WAS SET ON THE PREVIOUS LOADING. (EITHER STEP (4) OR PREVIOUS
```

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

     ; EXECUTION OF STEP (6).
05E7           NAC ,TRUE,MUX,LOOP
     / &       ALU D#0,,ABUS,QREG,SUBR,SUBR,DQ & CKALU & DBUS SNONE
     / &       STATSHFT B#0100000000100
     / &       REG ....,R10
     ;
     ;(7) THE CONTENTS OF IR0 AND ER8 ARE SWOPPED -- RETURN TAKES PLACE.
05E8           NAC ,FALSE,TZERO,CRTN & ALU ,,ABUS,RAMA,OR,OR,DZ
     / &       REG ,,RZERO,RZERO,R8 & CKALU & CKREG & DBUS SALU
     ;
     ;
     ;-------------------------------------------------------------------------------
     ;
     ; REGSELBIT -- THIS ROUTINE LOADS ER8 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 (R8).
     ;
     ; THE CONTENTS OF GR1 ARE MASKED BY 0000000000001111 AND PLACED IN THE QREG.
05E9 REGSELBIT: NAC ,,,CONT & ALU ,,DBUS,QREG,AND,AND,DA & REG ,,,GR1 & CKALU
     / &       DBUS SDATA & DATINSRT D#15
     ;
     ;(2) THE CONTENTS OF THE Q REGISTER ARE PASSED THROUGH THE ALU AND THE
     ; MICROSTATUS REGISTER LOADED ACCORDINGLY.  THE CONTENTS OF IR0 ARE PLACED
     ; IN ER8.
05EA           NAC ,,,CONT & ALU D#0,,,RAMA,OR,OR,ZQ & REG ,,RZERO,RZERO,R8
     / &       DBUS SALU & CKREG & STATSHFT B#0000000010000
     ;
     ;(3) IR0 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.
05EB           NAC ,TRUE,TZERO,PUSH & ALU ,,DBUS,RAMF,OR,OR,DZ & DBUS SDATA
     / &         DATINSRT D#1 & CKALU & REG ,,RZERO,RZERO
     ;
     ;(4) THE CONTENTS OF IR0 ARE SHIFTED DOWN CYCLIC.
05EC           NAC ,,,CONT & ALU ,,,RAMD,OR,OR,ZA & REG ,,RZERO,RZERO
     / &       CKALU & DBUS SNONE & STATSHFT B#0010100000000
     ;
     ;(5) THE CONTENTS OF THE QREG ARE DECREMENTED. END OF LOOP TAKES PLACE
     ; IF MICRO Z WAS SET EITHER DURING STEP (2) OR DURING THE PREVIOUS EXECUTION
     ; OF STEP (5).
05ED           NAC ,TRUE,MUX,LOOP & ALU D#0,,,QREG,SUBR,SUBR,ZQ & CKALU & DBUS SNONE
     / &       STATSHFT B#0000000000100
     ;
     ;(6) THE CONTENTS OF IR0 AND ER8 ARE SWOPPED -- RETURN TAKES PLACE
05EE           NAC ,FALSE,TZERO,CRTN & ALU ,,ABUS,RAMA,OR,OR,DZ & REG ,,RZERO,RZERO,R8
     / &       CKALU & CKREG & DBUS SALU
```



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

     ;
     ;-------------------------------------------------------------------------------
     ;
     ; SET BIT
     ; ********
     ;
     ; ENTRY POINT FOR REGISTER ADDRESSING
     ; ++++++++++++
     ;
05EF OPCODE51:: NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS SELBIT
     ;
     ;(2) THE MASK IN ER8 IS ORED WITH THE CONTENTS OF GR2.  THE RESULT IS LEFT
     ; IN GR2 AND RETURN TAKES PLACE.
05F0           NAC ,FALSE,TZERO,CRTN & ALU ,,ABUS,RAMF,OR,OR,DA & CKALU
     / &       REG ,,GR2,GR2,R8 & DBUS SNONE
     ;
     ; ENTRY POINT FOR DIRECT OR DIRECT INDEXED ADDRESSING
     ; ++++++++++++
     ;
05F1 OPCODE50:: NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS OPDX1
     ;
     ;(2) JUMP SUB TO SELBIT
05F2           NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS SELBIT
     ;
     ;(3) THE MASK IN ER8 IS LOADED INTO THE Q REGISTER.
05F3           NAC ,,,CONT & ALU ,,ABUS,QREG,OR,OR,DZ & REG ....,R8 & DBUS SNONE
     / &       CKALU
     ;
     ;(4)  THE DERIVED OPERAND IN ER5 IS ORED WITH THE CONTENTS OF THE Q REGISTER
     ; AND THE RESULT LEFT IN ER5.
05F4           NAC ,,,CONT & ALU ,,ABUS,NOP,OR,OR,DQ & REG ....,R5 & DBUS SALU
     / &       CKREG
     ;
     ;(5) UNCONDITIONAL JUMP TAKES PLACE TO OSGENPUR THE RETURN WILL MARK THE END
     ; OF THE EMULATION SEQUENCE WITH THE STACK CORRECTLY PRESERVED.
05F5           NAC ,,,JP & ALU & DBUS SNONE & BADDRESS OSGENPUR
     ;
     ;
     ; ENTRY POINT FOR INDIRECT AND INDIRECT INDEXED ADDRESSING
     ; ++++++++++++
05F6 OPCODE52:: NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS OPIX1
     ;
     ;(2) JUMP SUB TO SELBIT
05F7           NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS SELBIT
     ;(3) THE CONTENTS OF ER8 (MASK) IS PLACED IN THE QREG.
05F8           NAC ,,,CONT & ALU ,,ABUS,QREG,OR,OR,DZ & REG ....,R8 & CKALU & DBUS SNONE
```

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

     ;
     ;(4) THE DERIVED OPERAND IN ER5 IS ORED WITH THE CONTENTS OF THE Q REGISTER
     ; AND THE RESULT LEFT IN ER5
05F9           NAC ,,,JP & ALU ,,ABUS,NOP,OR,OR,DQ & REG ,,,,R5 & DBUS SALU
     / &       CKREG & BADDRESS OSGENPUR
     ;
     ;
     ;-------------------------------------------------------------------------
     ;
     ; RESET BIT
     ; **********
     ;
     ; ENTRY POINT FOR REGISTER ADDRESSING
     ;
     ;(1) JUMP SUB TO SELBIT
05FA OPCODE54:: NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS SELBIT
     ;
     ;(2) THE CONTENTS OF GR2 ARE ANDED WITH THE NOT OF THE MASK IN ER8. RETURN
     ; TAKES PLACE.
05FB           NAC ,FALSE,TZERO,CRTN & ALU ,,ABUS,RAMP,NOTRS,NOTRS,DA & CKALU
     / &       REG ,,GR2,GR2,R8 & DBUS SNONE
     ;
     ;
     ; ENTRY POINT FOR DIRECT AND DIRECT INDEXED ADDRESSING
     ; ++++++++++++
     ;
     ;(1) JUMP SUB TO OFDX1
05FC OPCODE53:: NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS OFDX1
     ;
     ;(2) JUMP SUB TO SELBIT
05FD           NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS SELBIT
     ;
     ;(3) THE NOT OF THE MASK IN ER8 IS PLACED IN QREG
05FE           NAC ,,,CONT & ALU ,,ABUS,QREG,EXNOR,EXNOR,DZ & REG ,,,,R8 & CKALU
     / &       DBUS SNONE
     ;
     ;(5) THE DERIVED OPERAND IN ER5 IS ANDED WITH THE CONTENTS OF THE Q REG AND
     ; THE RESULT PLACED IN ER5.
05FF           NAC ,,,JP & ALU ,,ABUS,NOP,AND,AND,DQ & REG ,,,,R5 & CKREG
     / &       DBUS SALU & BADDRESS OSGENPUR
     ; ENTRY POINT FOR INDIRECT AND INDIRECT INDEXED ADDRESSING
     ; ++++++++++++
0600 OPCODE55:: NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS OFIX1
     ;
     ;(2) JUMP SUB TO SELBIT
0601           NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS SELBIT
```



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

     ;
     ;(3) THE NOT OF THE MASK IN ER8 IS PLACED IN THE QREG
0602           NAC ,,,CONT & ALU ,,ABUS,QREG,EXNOR,EXNOR,DZ & REG ,,,,R8 & CKALU
     / &       DBUS SNONE
     ;
     ;(4) THE DERIVED OPERAND IN ER5 IS ANDED WITH THE MASK IN QREG.
     ; JUMP TO OSDX1 TAKES PLACE
0603           NAC ,,,JP & ALU ,,ABUS,NOP,AND,AND,DQ & REG ,,,,R5 & CKREG
     / &       DBUS SALU & BADDRESS OSGENPUR
     ;
     ;-------------------------------------------------------------------------
     ;
     ; TEST BIT
     ; *********
     ;
     ; ENTRY POINT FOR REGISTER ADDRESSING
     ; ++++++++++++
0604 OPCODE57:: NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS SELBIT
     ;
     ;(2) THE CONTENTS OF GR2 ARE ANDED WITH THE MASK IN ER8 AND THE MACHINE
     ;STATUS REGISTER LOADED ACCORDING TO THE RESULT.
     ; NB. 1750 DEFINES THAT IF NO BITS IN GR1 ARE SET MACHINE STATUS ZERO IS SET,
     ;     IF THE MOST SIGNIFICANT BIT IS SET, MACHINE STATUS N IS SET,
     ; IF ANY OTHER BITS ARE SET, MACHINE STATUS P IS SET.  THIS WILL TAKE PLACE
     ; CORRECTLY IN THE FOLLOWING INSTRUCTION.
0605           NAC ,FALSE,TZERO,CRTN & ALU ,B00,ABUS,NOP,AND,AND,DA
     / &       REG ,,GR2,GR2,R8 & DBUS SNONE & STATSHFT B"0000000010000
     ;
     / &        CARRYSEL CZERO
     ;
     ; ENTRY POINT FOR DIRECT AND DIRECT INDEXED ADDRESSING
     ; ++++++++++++
0606 OPCODE56:: NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS OFDX1
     ;
     ;(2) JUMPSUB TO SELBIT
0607           NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS SELBIT
     ;
     ;(3) THE DERIVED OPERAND IN ER5 IS PLACED IN QREG.
0608           NAC ,,,CONT & ALU ,,ABUS,QREG,OR,OR,DZ & CKALU & REG ,,,,R5
     / &       DBUS SNONE
     ;
     ;(4) THE MASK IN ER8 IS ANDED WITH THE DERIVED OPERAND AND THE MACHINE
     ;STATUS LOADED AS IN THE REGISTER ADDRESSING.
     ; RETURN TAKES PLACE.
0609           NAC ,FALSE,TZERO,CRTN & ALU ,B00,ABUS,NOP,AND,AND,DQ
     / &       REG ,,,,R8 & DBUS SNONE & STATSHFT B"0000000010000
```

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

     / &        CARRYSEL CZERO
     ;
     ;
     ; ENTRY POINT FOR INDIRECT AND INDIRECT INDEXED ADDRESSING
     ; ***********
0609 OPCODE58:: NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS OPIX1
     ;
     ;(2) JUMP SUB TO SELBIT
0609           NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS SELBIT
     ;
     ;(3) THE DERIVED OPERAND IN ER5 IS PLACED IN QREG
060C           NAC ...,CONT & ALU ,,ABUS,QREG,OR,OR,DZ & CKALU & REG ...,R5
     / &        DBUS SNONE
     ;
     ;(4) THE MASK IN ER8 IS ANDED WITH THE DERIVED OPERAND IN THE QREG.
     ;  MACHINE STATUS IS LOADED ACCORDING TO THE RESULT
     ;  RETURN TAKES PLACE.
060D           NAC ,FALSE,TZERO,CRTN & ALU ,B=0,ABUS,NOP,AND,AND,DQ & DBUS SNONE
     / &        REG ...,R8 & STATSHFT B=0000000010000
     / &        CARRYSEL CZERO
     ;
     ;-------------------------------------------------------------------------
     ;
     ; TEST AND SET BIT
     ; ****************
     ;
     ; ENTRY POINT FOR DIRECT AND DIRECT INDEXED ADDRESSING
     ; ***********
060E OPCODE59:: NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS OFDX1
     ;
     ;(2) JUMP SUB TO SELBIT
060F           NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS SELBIT
     ;
     ;(3) THE MASK IN ER8 IS PLACED IN QREG.
0610           NAC ...,CONT & ALU ,,ABUS,QREG,OR,OR,DZ & CKALU & REG ...,R8
     / &        DBUS SNONE
     ;
     ;(4) THE DERIVED OPERAND IN ER5 IS ANDED WITH THE MASK IN QREG. THE MACCHINE
     ; STATUS IS LOADED ACCORDING TO THE RESULT.
0611           NAC ...,CONT & ALU ,B=0,ABUS,NOP,AND,AND,DQ & REG ...,R5 & DBUS SNONE
     / & STATSHFT B=0000000010000
     / &        CARRYSEL CZET
     ;
     ;(5) THE DERIVED OPERAND IN ER5 IS ORED WITH THE MASK IN THE QREG. THE RESULT
     ; IS PLACED IN ER5.  JUMP PIPELINE TAKES PLACE TO OSGENPUR
0612           NAC ...,JP & ALU ,,ABUS,NOP,OR,OR,DQ & REG ...,R5 & CKREG & DBUS SALU
```



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

     / &        BADDRESS OSGENPUR
     ;
     ;-------------------------------------------------------------------------
     ;
     ; SET VARIABLE BIT IN REGISTER
     ; ****************************
     ;
     ;ENTRY POINT
     ;***********
     ;
     ;(1) JUMP TAKES PLACE TO REGSELBIT WHICH PROVIDES A MASK IN ER8 WHICH
     ; SELECTS THE REQUIRED BIT ACCORDING TO THE CONTENTS OF GR1.
0613 OPCODE5A:: NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS REGSELBIT
     ;
     ;(2) THE MASK IN ER8 IS ORED WITH THE CONTENTS OF REGISTER GR2, RETURN TAKES
     ; PLACE.
0614           NAC ,FALSE,TZERO,CRTN & ALU ,,ABUS,RAMP,OR,OR,DA & REG ,,GR2,GR2,R8
     / &        CKALU & DBUS SNONE
     ;
     ;-------------------------------------------------------------------------
     ;
     ; RESET VARIABLE BIT IN REGISTER
     ; ******************************
     ;
     ; ENTRY POINT
     ; ***********
     ;
     ;(1) JUMP SUB TO REGSELBIT (DESCRIBED ABOVE).
0615 OPCODE5C:: NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS REGSELBIT
     ;
     ;(2) THE NOT OF THE MASK IN ER8 IS ANDED WITH THE CONTENTS OF REGISTER GR2
     ; THE RESULT IS PLACED IN GR2.
     ; RETURN TAKES PLACE.
0616           NAC ,FALSE,TZERO,CRTN & ALU ,,ABUS,RAMP,NOTRS,NOTRS,DA & REG ,,GR2,GR2,R8
     / &        CKALU & DBUS SNONE
     ;
     ;-------------------------------------------------------------------------
     ;
     ; TEST VARIABLE BIT IN REGISTER
     ; *****************************
     ;
     ; ENTRY POINT
     ; ***********
     ;
     ;(1) JUMP SUB TO REGSELBIT
0617 OPCODE5E:: NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS REGSELBIT
```

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

      ;
      ;(2) THE CONTENTS OF GR2 ARE ANDED WITH THE MASK IN ER6 AND MACHINE STATUS
      ; IS LOADED ACCORDING TO THE RESULT.  THE CONTENTS OF GR2 ARE NOT CHANGED.
      ; RETURN TAKES PLACE.
 0618             NAC ,FALSE,TZERO,CRTN & ALU ,D#0,ABUS,NOP,AND,AND,DA
      / &         RBS ...GR2,R6 & DBUS SNONE & STATSHFT D#0000000010000
      / &         CARRYSEL CZERO
      ;
      ;-----------------------------------------------------------------------
      ;
      ;
      ;
      ;-----------------------------------------------------------------------
      ;
      ; JUMP ON CONDITION
      ; ^^^^^^^^^^^^^^^^^
      ;
      ; ENTRY POINT FOR DIRECT AND DIRECT INDEXED ADDRESSING
      ; ++++++++++++
      ;(0) JUMP SUB TO FETDIADD.  THIS FETCHES THE SECOND WORD OF THE INSTRUCTION
      ;   (THE LABEL) INTO ER2 AND INDEXES IT WITH THE CONTENTS OF THE INDEX REGISTER
      ;   IF ONE IS SPECIFIED.
 0619 OPCODE70:: NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS FETDIADD
      ;
      ;
      ;(1)THE CONTENTS OF IR0 ARE PLACED IN ER8.
 061A JC1:       NAC ...CONT & ALU ..,NOP,OR,OR,ZA & RBS ...RZERO,R8 & DBUS SALU
      / &         CRER8
      ;
      ;(2)THE INSTRUCTION IN ER1 IS PLACED IN IR0.
 061B             NAC ...CONT & ALU ..ABUS,RAMF,OR,OR,DZ & RBS ..RZERO,RZERO,R1
      / &         CRALU & DBUS SNONE
      ;
      ;(3)THE INSTRUCTION IS MASKED BY (0000000011110000).
 061C             NAC ...CONT & ALU ..DBUS,RAMF,AND,AND,DA & RBS ..,RZERO,RZERO
      / &         CRALU & DBUS SDATA & DATINSHT D#240
      ;
      ;(4) THE CONTENTS OF IR0 ARE DOWNSHIFTED.
      ;  THE NEXT INSTRUCTION ADDRESS IS PUSHED INTO THE STACK AND THE COUNTER
      ; LOADED WITH 5.
 061D             NAC ,FALSE,TZERO,PUSH & ALU ...RAMD,OR,OR,ZA & RBS ..,RZERO,RZERO
      / &         CRALU & DBUS SNONE & DATINSHT D#5
      ; NB. D#8 GIVES THE CORRECT STATSHFT FIELD FOR THE DOWNSHIFT.
      ;
      ;
      ;(5) THE CONTENTS OF IR0 ARE DOWNSHIFTED 7 TIMES CYCLICLY.
```



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

 061E             NAC ,FALSE,COUNT,RFCT & ALU ...RAMD,OR,OR,ZA & RBS ..RZERO,RZERO
      / &         CRALU & STATSHFT D#0010100000000
      ;
      ;(7) THE CONTENTS OF THE STATUS REGISTER ARE ANDED WITH THE CONTENTS OF
      ; IR0.
 061F             NAC ...CONT & ALU ..DBUS,NOP,AND,AND,DA & RBS ..,RZERO,RZERO
      / &         DBUS SSTATUS & STATSHFT D#0000000100000
      ;
      ;(8) THE CONTENTS OF ER6 ARE PLACED BACK IN IR0, RETURN TAKES PLACE IF
      ; ZMSB WAS SET.
 0620             NAC ,TRUE,ZMSB,CRTN & ALU ..ABUS,RAMF,OR,OR,DZ & CRALU & RBS ..RZERO,,R8
      / &         DBUS SNONE
      ;
      ;(3)THE QREG IS LOADED WITH THE CONTENTS OF ER2.
 0621             NAC ...CONT & ALU ..ABUS,QREG,OR,OR,DZ & RBS ....,R2 & CRALU
      / &         DBUS SNONE
      ;
      ;(4) THE PC IS LOADED WITH THE CONTENTS OF THE QREG. RETURN TAKES PLACE.
 0622             NAC ,FALSE,TZERO,CRTN & ALU ...NOP,OR,OR,ZQ & RBS ....,R8 & CRER8
      / &         DBUS SALU
      ;
      ;  ENTRY POINT FOR INDIRECT OR INDIRECT INDEXED ADDRESSING
      ;  ++++++++++++
      ;
      ;(0) JUMP SUB TO OFDI1.  THIS FETCHES THE SECOND WORD OF THE INSTRUCTION TO
      ; ER8, INDEXES IT WITH THE CONTENTS OF AN INDEX REGISTER IF SPECIFIED AND
      ; OPTAINS AN OPERAND FROM THE RESULTING ADDRESS PLACING IT IN ER5.
      ; THE PROGRAM COUNTER IS INCREMENTED TO TAKE ACCOUNT OF THE TWO WORD INSTRUCTION
 0623 OPCODE71:: NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS OFDI1
      ;
      ;(2) THE DERIVED OPERAND IN ER5 IS PLACED IN THE QREG.
 0624             NAC ...CONT & ALU ..ABUS,QREG,OR,OR,DZ & CRALU & RBS ....,R5
      / &         DBUS SNONE
      ;
      ;(3) THE QREG CONTENTS ARE PLACED IN ER2. JUMP TO JC1 TAKES PLACE.
 0625             NAC ...,JP & ALU ...NOP,OR,OR,ZQ & RBS ....R2 & CRER8 & DBUS SALU
      / &         BADDRESS JC1
      ;
      ;-----------------------------------------------------------------------
      ;
      ; JUMP TO SUBROUTINE
      ; ^^^^^^^^^^^^^^^^^^
      ;
      ; ENTRY POINT FOR DIRECT OR DIRECT INDEXED ADDRESSING
      ; ++++++++++++
      ;
```

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

      ;(1) JUMP TO FETDIADD.  THIS FETCHES THE SECOND WORD OF THE TWO WORD INSTRUCTION
      ; INTO ER2.  THE RESULT IS THEN INDEXED BY THE CONTENTS OF THE INDEX REGISTER
      ; IF SPECIFIED.  THE PROGRAM COUNTER CONTENTS IN ER0 ARE INCREMENTED TO TAKE
      ; ACCOUNT OF THE TWO WORD INSTRUCTION.  NB THE PROGRAM COUNTER WILL ALREADY
      ; POINT TO THE MEMORY LOCATION AFTER THE CURRENT INSTRUCTION SINCE IT IS
      ; INCREMENTED ONCE IN THE FETCH SEQUENCE.
 0626 OPCODE72:: NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS FETDIADD
      ;
      ;(2) THE CONTENTS OF ER2 (THE NEW PC VALUE) ARE PLACED IN GR1 (RA).
 0627            NAC ,,,CONT & ALU ,,ABUS,RAMF,OR,OR,DZ & REG ,,GR1,,R2 & CKALU
      / &        DBUS SNONE
      ;
      ;(3) THE CONTENTS OF GR1 (RA) ARE SWOPPED WITH THE CONTENTS OF ER0, RETURN
      ; TAKES PLACE.
 0628            NAC ,FALSE,TZERO,CRTN & ALU ,,ABUS,RAMA,OR,OR,DZ & REG ,,GR1,GR1,R0
      / &        CKREG & CKALU & DBUS SALU
      ;
      ;-------------------------------------------------------------------------------
      ;
      ; SUBTRACT ONE AND JUMP
      ; ********************
      ;
      ; ENTRY POINT
      ; ***********
      ;(0) JUMP SUB TO FETDIADD TAKES PLACE.
 0629 OPCODE73:: NAC ,FALSE,TZERO,CJS & ALU & DBUS SNONE & BADDRESS FETDIADD
      ;
      ;(1) THE CONTENTS OF GR1 (RA) ARE DECREMENTED, THE MACHINE STATUS REGISTER
      ; IS LOADED ACCORDINGLY.
 062A            NAC ,,,CONT & ALU ,B#0,,RAMF,SUBR,SUBR,ZA & REG ,,GR1,GR1
      / &        CKALU & DBUS SNONE & STATSHFT B#0000000010000 & MACSTEN B#0000
      / &        CARRYSEL CZERO
      ;
      ;(2) RETURN TAKES PLACE IF MACHINE STATUS ZERO IS SET.
 062B            NAC ,TRUE,MUX,CRTN & ALU & DBUS SNONE & STATSHFT B#0000000100100
      ;
      ;
      ;(4) THE QREG IS LOADED WITH THE CONTENTS OF ER2.
 062C            NAC ,,,CONT & ALU ,,ABUS,QREG,OR,OR,DZ & REG ,,,,R2 & CKALU
      / &        DBUS SNONE
      ;
      ;(5) THE PC (ER0) IS LOADED WITH THE CONTENTS OF QREG.
      ;  RETURN TAKES PLACE.
 062D            NAC ,FALSE,TZERO,CRTN & ALU ,,,NOP,OR,OR,ZQ & REG ,,,,R0 & CKREG
      / &        DBUS SALU
      ;
```



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

      ;-------------------------------------------------------------------------------
      ;
      ; BRANCH UNCONDITIONALLY
      ; **********************
      ;
      ;(0) THE PROGRAM COUNTER CONTENTS ARE DECREMENTED.
 062E OPCODE74:: NAC ,,,CONT & ALU ,,ABUS,NOP,SU2S,SUBS,DZ & CKREG & DBUS SALU
      / &        REG ,,,,R0 & STATSHFT B#0000000000000
      ;
      ;(1) THE CONTENTS OF ER1 IS PLACED IN QREG.
 062F            NAC ,,,CONT & ALU ,,ABUS,QREG,OR,OR,DZ & REG ,,,,R1 & CKALU
      / &        DBUS SNONE
      ;
      ;(2) THE CONTENTS OF QREG ARE MASKED WITH 0000000010000000 BUT NO REGISTERS
      ; ARE LOADED.
 0630            NAC ,,,CONT & ALU ,,DBUS,NOP,AND,AND,DQ & DBUS SDATA
      / &        DATINSRT D#128
      ;
      ;(3) JUMP TO BR1 TAKES PLACE IF THE MASKED BIT WAS SET (IE LSB NOT =0)
 0631            NAC ,FALSE,ZLSB,CJP & ALU & DBUS SNONE & BADDRESS BR1
      ;
      ;(4) CONTINUE KNOWING THAT INSTRUCTION COUNTER RELATIVE OPERAND IS POSITIVE.
      ; THE CONTENTS OF QREG ARE MASKED BY (0000000001111111) AND LEFT IN QREG
 0632            NAC ,,,CONT & ALU ,,DBUS,QREG,AND,AND,DQ & CKALU & DBUS SDATA
      / &        DATINSRT D#127
      ;
      ;(5) THE CONTENTS OF QREG ARE ADDED TO ER0 (PC), RETURN TAKES PLACE.
 0633            NAC ,FALSE,TZERO,CRTN & ALU ,,ABUS,NOP,ADD,ADD,DQ & CKREG
      / &        REG ,,,,R0 & DBUS SALU & STATSHFT B#0000000000000
      ;
      ;(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.
 0634 BR1:       NAC ,,,CONT & ALU ,,DBUS,QREG,OR,OR,DQ & CKALU & DBUS SDATA
      / &        DATINSRT D#65280
      ;
      ;(7) THE CONTENTS OF QREG ARE ADDED TO THE CONTENTS OF ER0 (PC), RETURN
      ; TAKES PLACE.
 0635            NAC ,FALSE,TZERO,CRTN & ALU ,,ABUS,NOP,ADD,ADD,DQ & DBUS SALU
      / &        CKREG & REG ,,,,R0 & STATSHFT B#0000000000000
      ;
      ;-------------------------------------------------------------------------------
      ;
      ; BRANCH IF EQUAL TO ZERO
```

```
AMD05/29 AMDASM MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EMSL

      ; ************************
      ;
      ; ENTRY POINT
      ; ***********
      ;
      ;(1) THE STATUS REGISTER CONTENTS ARE LOADED INTO THE QREG
0636 OPCODE75:: NAC ,,,CONT & ALU ,,DBUS,QREG,OR,OR,DZ & DBUS SSTATUS & CKALU
      / &        STATSHFT B#0000001000000
      ;
      ;(2) THE CONTENTS OF THE Q REGISTER ARE MASKED BY (0010000000000000) SO THAT
      ; ONLY THE ZERO BIT IS LEFT.
0637            NAC ,,,CONT & ALU ,,DBUS,NOP,AND,AND,DQ & DBUS SDATA
      / &       DATINSRT D#8192
      ;
      ;(3) RETURN TAKES PLACE IF THE BIT WAS ZERO (IE ZMSB IS SET)
0638            NAC ,TRUE,ZMSB,CRTN & ALU & DBUS SNONE
      ;
      ;(4) JUMP TO OPCODE74
0639            NAC ,,,JP & ALU & DBUS SNONE & BADDRESS OPCODE74
      ;
      ;-------------------------------------------------------------------------
      ;
      ; BRANCH IF LESS THAN ZERO
      ; ************************
      ;
      ;(1) THE STATUS REGISTER CONTENTS ARE LOADED INTO THE QREG.
063A OPCODE76:: NAC ,,,CONT & ALU ,,DBUS,QREG,OR,OR,DZ & DBUS SSTATUS & CKALU
      / &        STATSHFT B#0000001000000
      ;
      ;(2) THE CONTENTS OF THE QREG ARE MASKED WITH (0000100000000000) SO THAT
      ; ONLY THE NEGATIVE BIT IS LEFT.
063B            NAC ,,,CONT & ALU ,,DBUS,NOP,AND,AND,DQ & DBUS SDATA
      / &       DATINSRT D#4096
      ;
      ;(3) RETURN TAKES PLACE IF THE BIT WAS ZERO (IE IF ZMSB IS SET.)
063C            NAC ,TRUE,ZMSB,CRTN & ALU & DBUS SNONE
      ;
      ;(4) JUMP TO OPCODE74
063D            NAC ,,,JP & ALU & DBUS SNONE & BADDRESS OPCODE74
      ;
      ;-------------------------------------------------------------------------
      ;
      ; BRANCH IF LESS THAN OR EQUAL TO ZERO
      ; ************************************
      ;
      ;(1) THE STATUS REGISTER CONTENTS ARE LOADED INTO THE QREG.
```



```
AMD05/29 AMDASM MICRO ASSEMBLER, V1.0
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EMSL

063E OPCODE78:: NAC ,,,CONT & ALU ,,DBUS,QREG,OR,OR,DZ & DBUS SSTATUS & CKALU
      / &        STATSHFT B#0000001000000
      ;
      ;(2) THE CONTENTS OF THE QREG ARE MASKED WITH (0011000000000000) SO THAT
      ; THE NEGATIVE BIT AND ZERO BIT ARE LEFT.
063F            NAC ,,,CONT & ALU ,,DBUS,NOP,AND,AND,DQ & DBUS SDATA
      / &       DATINSRT D#12288
      ;
      ;(3) RETURN TAKES PLACE IF BOTH THE BITS WERE ZERO (IE ZMSB IS SET.)
0640            AC ,TRUE,ZMSB,CRTN & ALU & DBUS SNONE
      ;
      ;(4) JUMP TO OPCODE74
0641            NAC ,,,JP & ALU & DBUS SNONE & BADDRESS OPCODE74
      ;
      ;-------------------------------------------------------------------------
      ;
      ; BRANCH IF GREATER THAN ZERO
      ; ***************************
      ;
      ;(1) THE STATUS REGISTER CONTENTS ARE LOADED INTO THE QREG.
0642 OPCODE79:: NAC ,,,CONT & ALU ,,DBUS,QREG,OR,OR,DZ & DBUS SSTATUS & CKALU
      / &        STATSHFT B#0000001000000
      ;
      ;(2) THE CONTENTS OF THE QREG ARE MASKED WITH 0100000000000000 TO, LEAVE
      ; ONLY THE POSITIVE BIT.
0643            NAC ,,,CONT & ALU ,,DBUS,NOP,AND,AND,DQ & DBUS SDATA
      / &       DATINSRT D#16384
      ;
      ;(3) RETURN TAKES PLACE IF THE POSITIVE BIT WAS NOT SET. (IE ZMSB WAS SET)
0644            NAC ,TRUE,ZMSB,CRTN & ALU & DBUS SNONE
      ;
      ;(4) JUMP TO OPCODE74
0645            NAC ,,,JP & ALU & DBUS SNONE & BADDRESS OPCODE74
      ;
      ;-------------------------------------------------------------------------
      ;
      ; BRANCH IF NOT EQUAL TO ZERO
      ; ***************************
      ;
      ;(1) THE STATUS REGISTER CONTENTS ARE LOADED INTO THE QREG.
0646 OPCODE7A:: NAC ,,,CONT & ALU ,,DBUS,QREG,OR,OR,DZ & DBUS SSTATUS & CKALU
      / &        STATSHFT B#0000001000000
      ;
      ;(2) THE CONTENTS OF THE QREG ARE MASKED WITH (0101000000000000) SO THAT
      ; THE NEGATIVE AND POSITIVE BITS ARE LEFT.
0647            NAC ,,,CONT & ALU ,,DBUS,NOP,AND,AND,DQ & DBUS SDATA
```

1 MAY 1981

AMD05/29 AMDASM MICRO ASSEMBLER, V1.0 PAGE 88
EMULATION SEQUENCES FOR SINGLE LENGTH SUBSET FOR 1750. (EMSL

```
      / &         DATINSRT D#20480
      ;
      ;(3) RETURN TAKES PLACE IF NEITHER OF THESE TWO BITS WERE SET. IE ZMSB IS SET.
 0648             NAC ,TRUE,ZMSB,CRTN & ALU & DBUS SNONE
      ;
      ;(4) JUMP TO OPCODE74
 0649             NAC ...JP & ALU & DBUS SNONE & BADDRESS OPCODE74
      ;
      ;-------------------------------------------------------------------------
      ;
      ; BRANCH IF GREATER THAN OR EQUAL TO ZERO
      ; ***************************************
      ;
      ;(1) THE CONTENTS OF THE STATUS REGISTER ARE LOADED INTO THE QREG.
 064A OPCODE79:: NAC ...CONT & ALU ..DBUS,QREG,OR,OR,DZ & CKALU & DBUS SSTATUS
      / &         STATSHFT B#0000001000000
      ;
      ;(2) THE CONTENTS OF THE QREG ARE MASKED BY (0110000000000000) SO THAT
      ; ONLY THE POSITIVE AND ZERO BITS ARE LEFT.
 064B             NAC ...CONT & ALU ..DBUS,NOP,AND,AND,DQ & DBUS SDATA
      / &         DATINSRT D#24576
      ;
      ;(3) RETURN TAKES PLACE IF BOTH BITS WERE ZERO. IE ZMSB IS SET.
 064C             NAC ,TRUE,ZMSB,CRTN & ALU & DBUS SNONE
      ;
      ;(4) JUMP TO OPCODE74
 064D             NAC ...JP & ALU & DBUS SNONE & BADDRESS OPCODE74
      ;
      ;-------------------------------------------------------------------------
      ;
      ;
      ;
      ;
      ;
      ;-------------------------------------------------------------------------
      ;
      ; THIS IS THE END OF THE EMULATION SEQUENCES.
      ;
      ;
      ;
      END
```

REFERENCES

No.	Author	Title, etc
1	USAF	MIL-STD-[illegible] Aerospace computer instruction set architecture. [illegible], Washington DC 20360, 1 March 1980
2	A.A. Callaway S.J. Shrimpton	RAL contract with the USAF MIL-STD-1750 instruction set architecture standardisation programme. RAL Technical Memorandum FS 286 (1979)
3	Advanced Micro Devices	[illegible]
4	Advanced Micro Computers	A [illegible] system. [illegible]

Fig 1

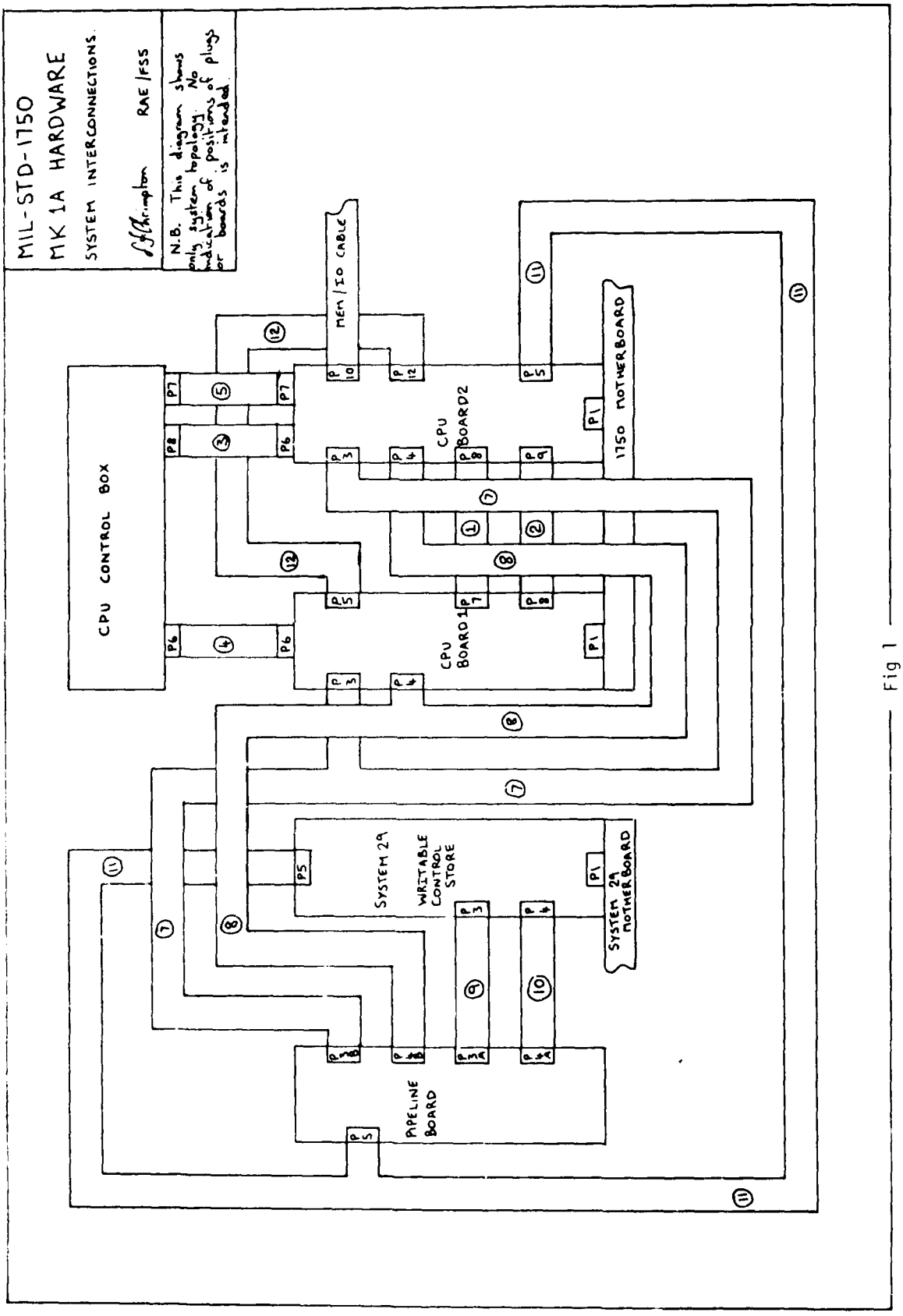

MIL-STD-1750
MK 1A HARDWARE
SYSTEM INTERCONNECTIONS
RAE/FSS
N.B. This diagram shows only system topology. No indication of positions of plugs or boards is intended.
CPU CONTROL BOX
MEM/IO CABLE
CPU BOARD2
CPU BOARD1
1750 MOTHERBOARD
SYSTEM 2A
WRITABLE CONTROL STORE
SYSTEM 2A MOTHERBOARD
PIPELINE BOARD

Fig 1

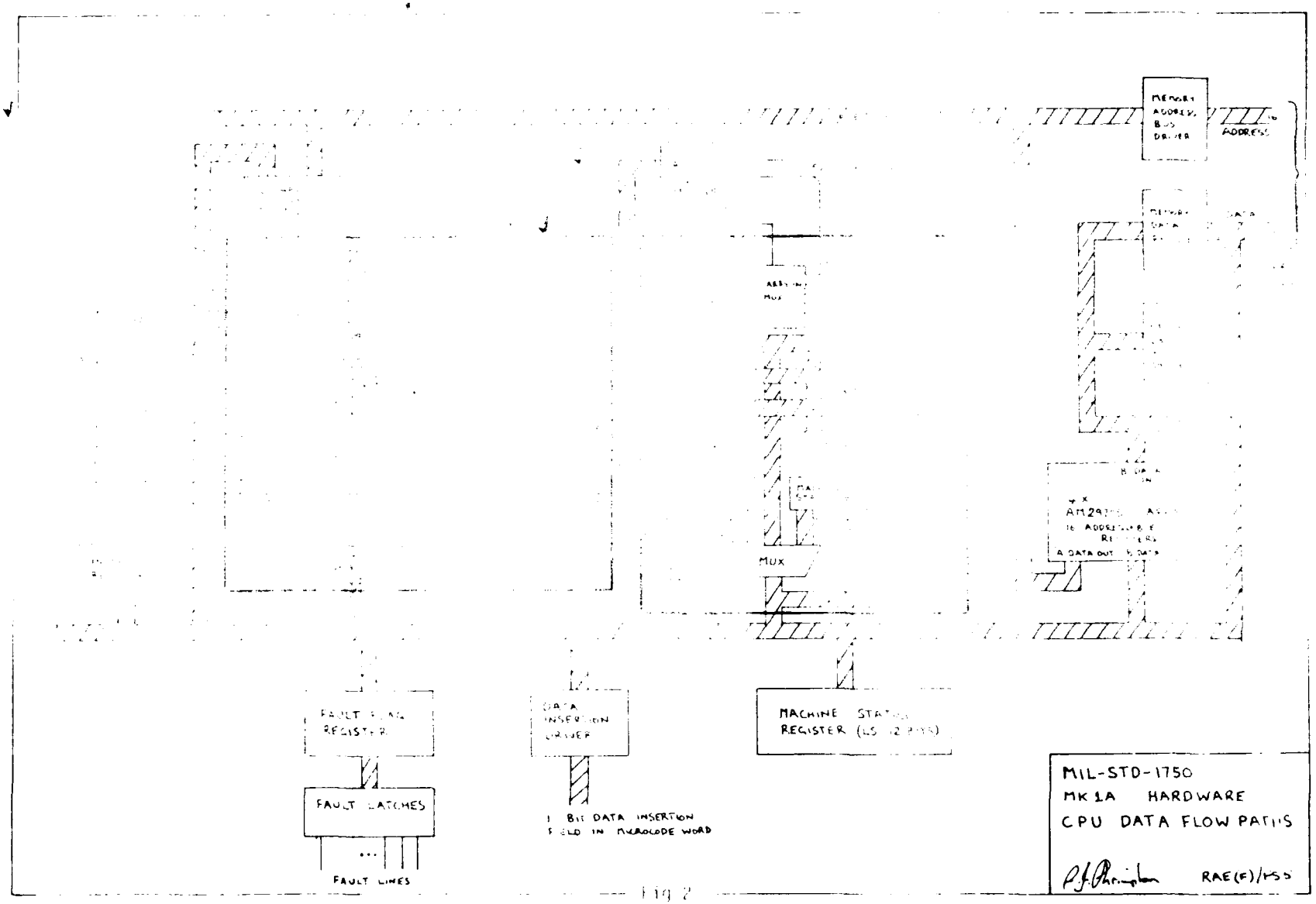

MEMORY ADDRESS BUS DRIVER
ADDRESS
MUX
FAULT FLAG REGISTER
FAULT LATCHES
FAULT LINES
DATA INSERTION DRIVER
1 BIT DATA INSERTION FIELD IN MICROCODE WORD
MACHINE STATUS REGISTER (LS 12 BITS)
MIL-STD-1750
MK 1A HARDWARE
CPU DATA FLOW PATHS
RAE(F)/HSS

Fig 2

Fig 2

Fig 3

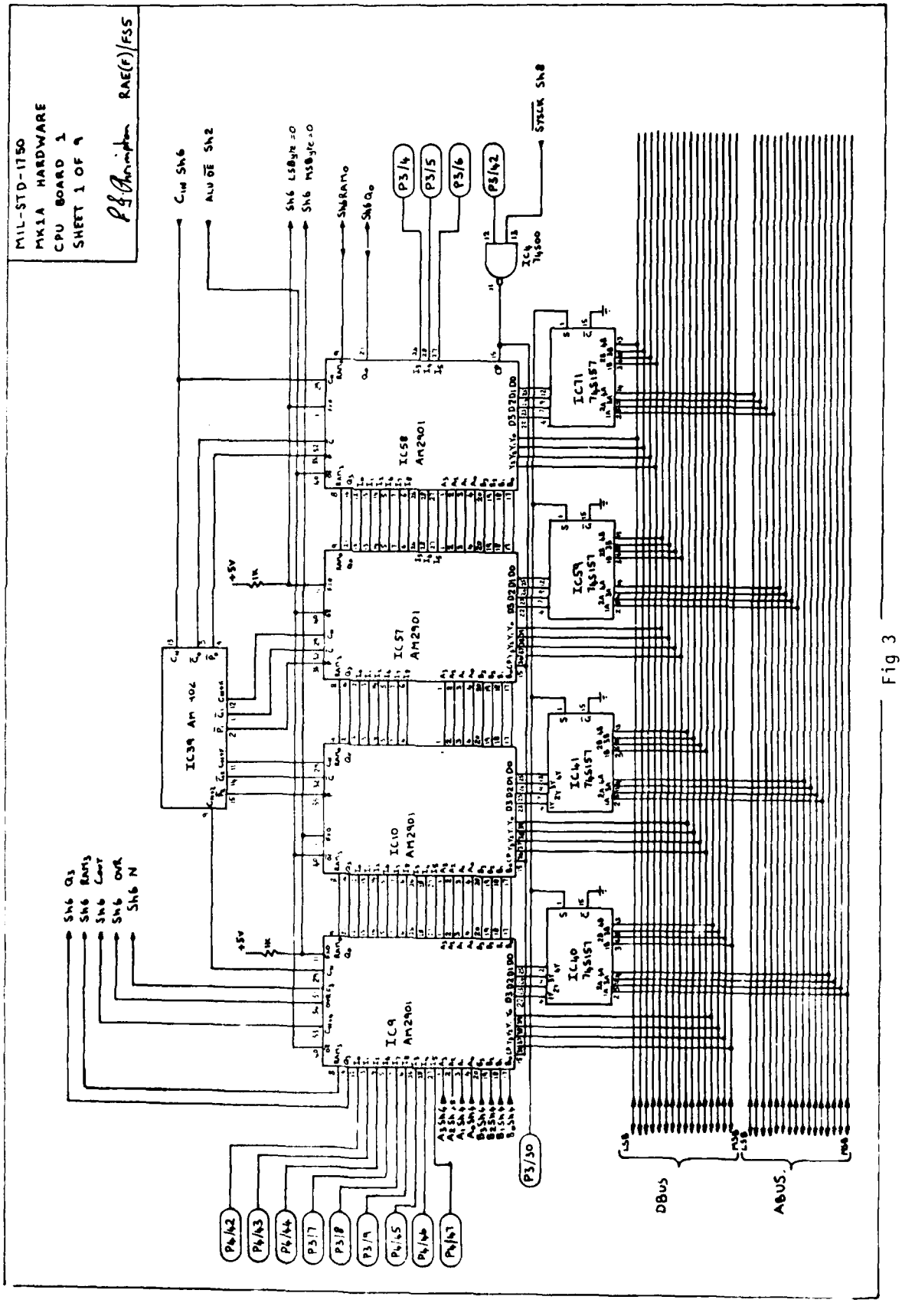

MIL-STD-1750
MK1A HARDWARE
CPU BOARD 1
SHEET 1 OF 9
P.G. Hampton RAE(F)/FSS
IC39 Am 402
IC58 Am2901
IC57 Am2901
IC10 Am2901
IC9 Am2901
IC71 74S157
IC59 74S157
IC41 74S157
IC40 74S157
IC4 74S00
+5V
1K
DBUS
ABUS

Fig 3

Fig 4

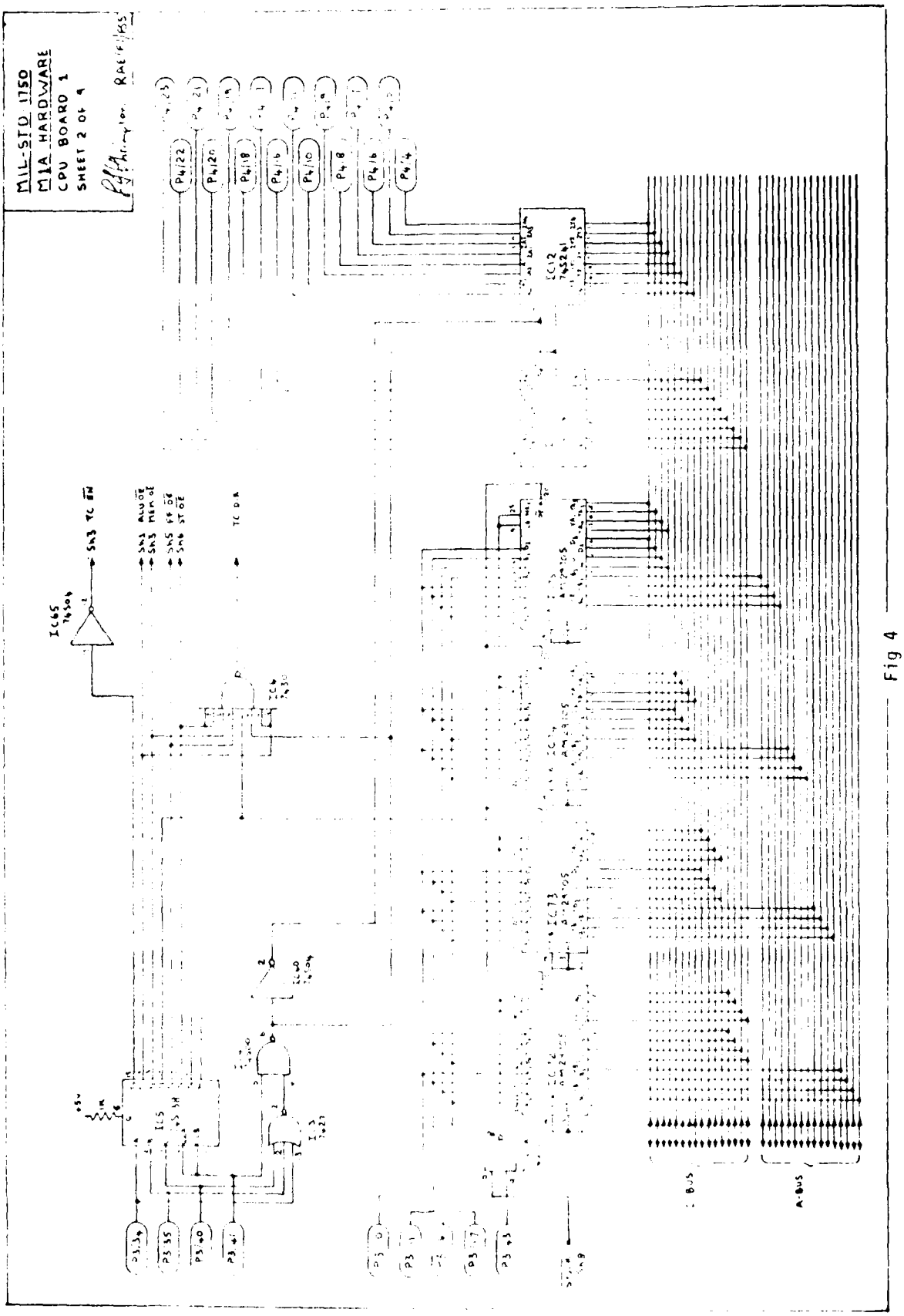

MIL-STD 1750
MIA HARDWARE
CPU BOARD 1
SHEET 2 OF 9
IC12
74S241
IC65
74S04
IC66
74S00
IC67
74S20
IC73
IC74
IC75
A-BUS
I-BUS

Fig 4

Fig 5

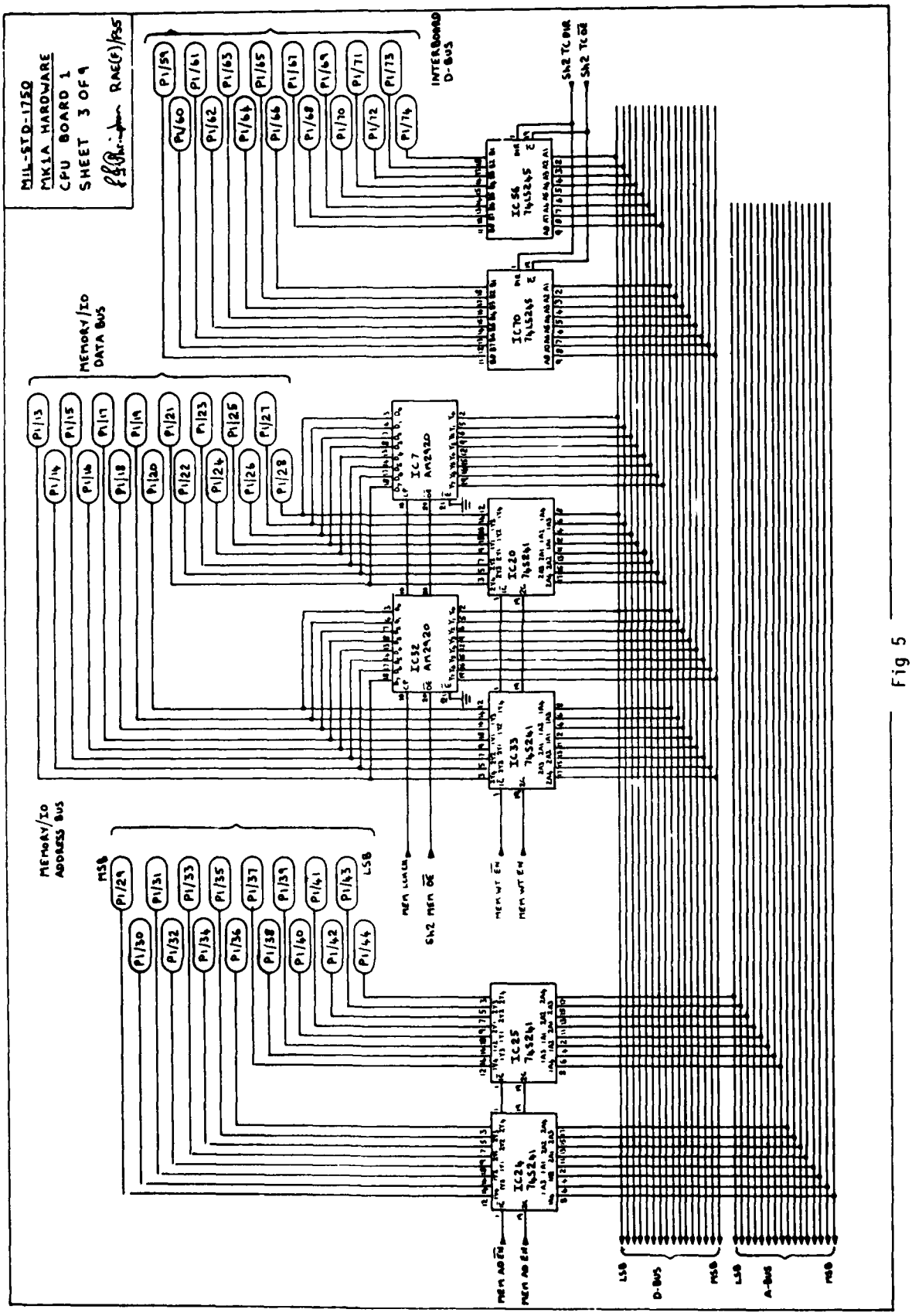

MIL-STD-1750
MIKIA HARDWARE
CPU BOARD 1
SHEET 3 OF 9
MEMORY/IO DATA BUS
MEMORY/IO ADDRESS BUS
INTERBOARD D-BUS
IC56 74S245
IC70 74S245
IC7 AM2920
IC32 AM2920
IC20 74S241
IC33 74S241
IC25 74S241
IC24 74S241
D-BUS
A-BUS
MSB
LSB

Fig 5

Fig 6

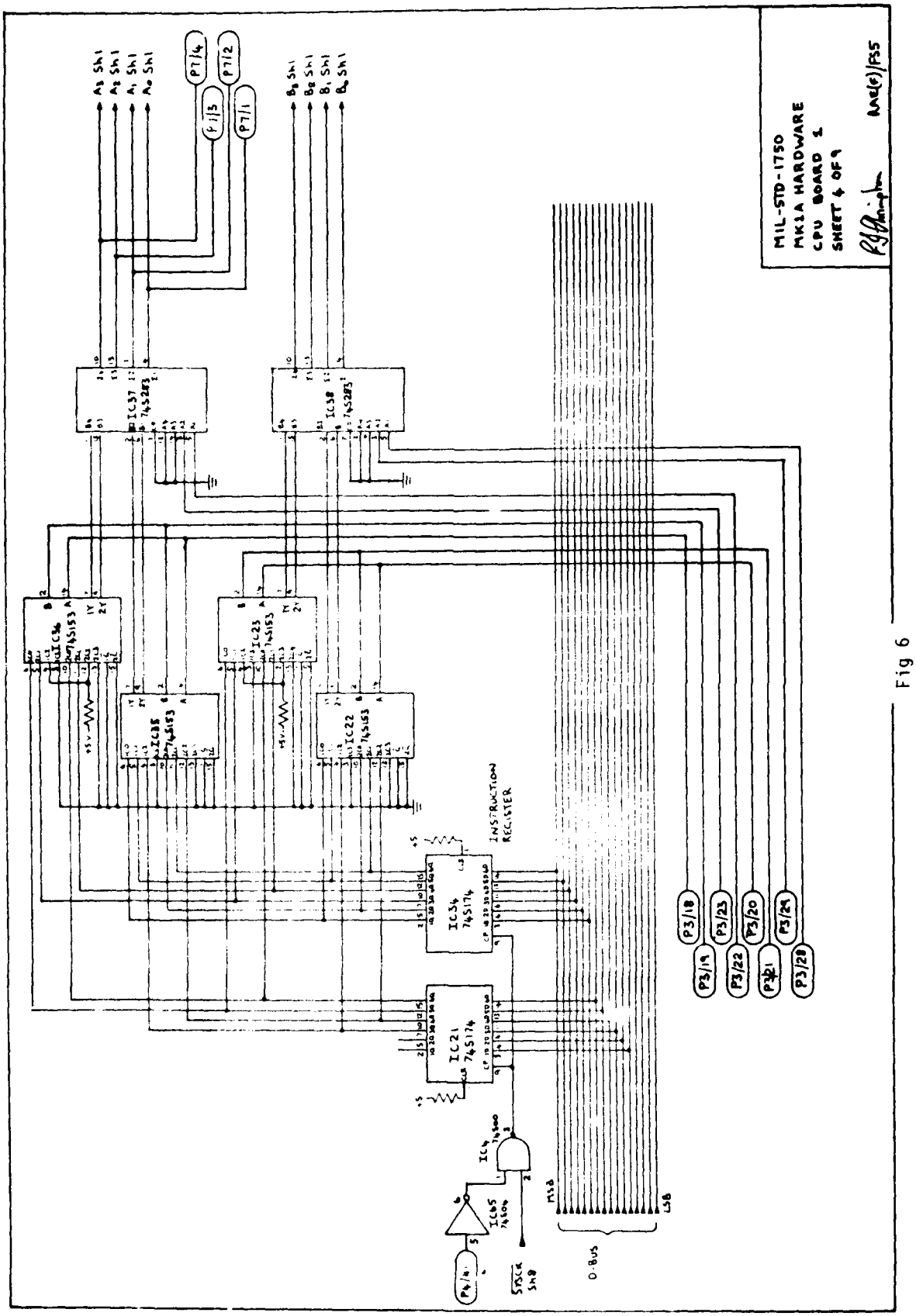

MIL-STD-1750
MK1A HARDWARE
CPU BOARD 1
SHEET 4 OF 9
INSTRUCTION REGISTER
IC37 74S283
IC38 74S283
IC36 74S153
IC23 74S153
IC35 74S153
IC22 74S153
IC34 74S174
IC21 74S174
IC4 74S00
IC45 74S04
P7/4
P7/3
P7/2
P7/1
P3/18
P3/19
P3/23
P3/22
P3/20
P3/21
P3/24
P3/28
P4/8
STACK SW9
D-BUS

Fig 6

Fig 7

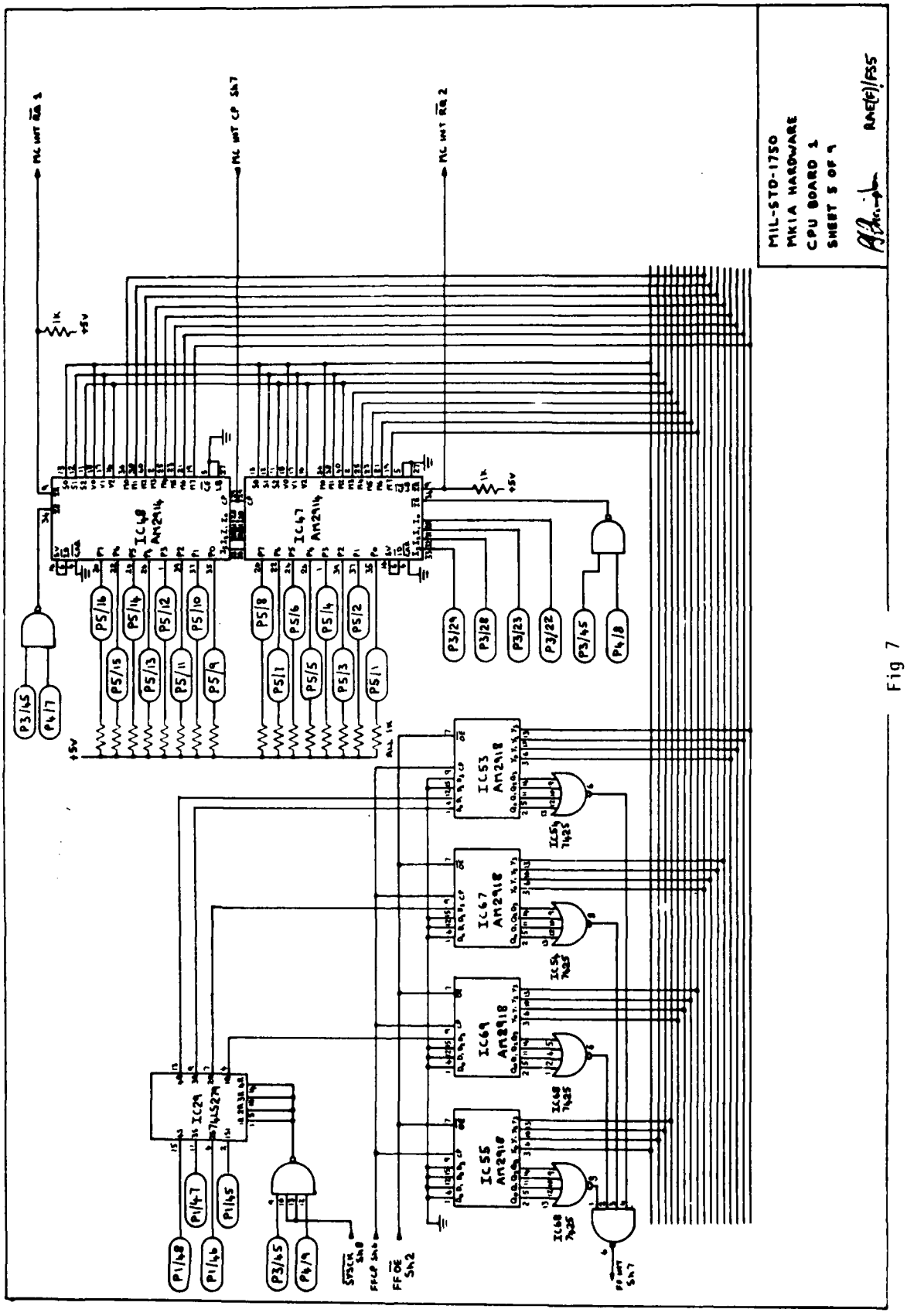

MIL-STD-1750
MK1A HARDWARE
CPU BOARD 1
SHEET 5 OF 9
IC48 AM2914
IC47 AM2914
IC53 AM2918
IC67 AM2918
IC69 AM2918
IC55 AM2918
IC24 74S240
IC54 7425
IC68 7425
IC55 7425
P3/24
P3/28
P3/23
P3/22
P3/45
P4/8
P3/45
P4/7
P1/48
P1/46
P1/47
P1/45
P3/45
P4/9
+5V
1K

Fig 7

Fig 8

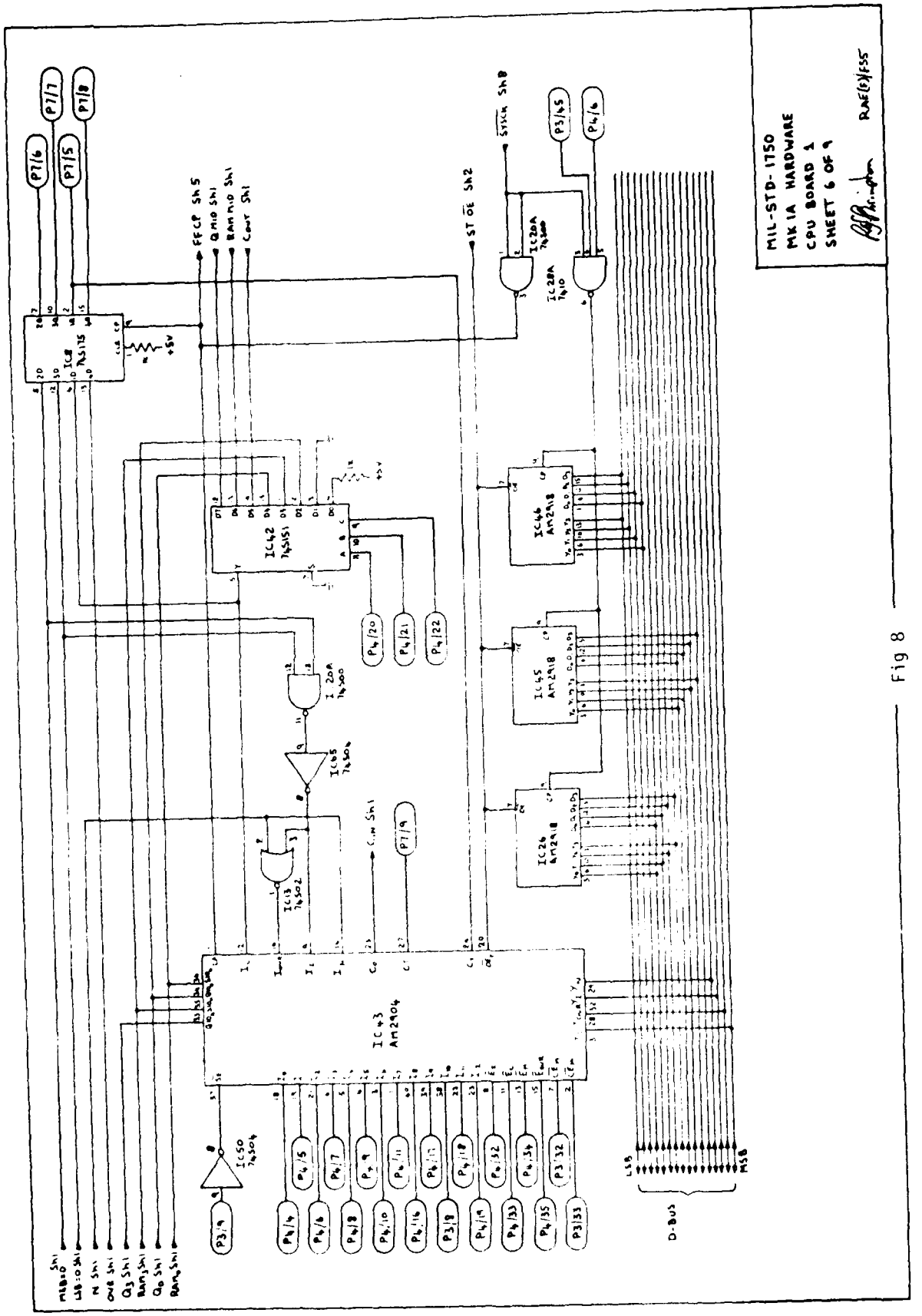

MIL-STD-1750
MK 1A HARDWARE
CPU BOARD 1
SHEET 6 OF 9
RAE(F)FSS
IC43 Am2904
IC42 74S151
IC46 Am2918
IC45 Am2918
IC24 Am2918
IC28A 74S08
IC28B 74S10
IC26 74S00
IC13 74S02
IC65 74S04
IC50 74S04
IC8 74S175
P7/6
P7/7
P7/5
P7/8
P3/45
P4/6
P4/20
P4/21
P4/22
P7/9
P3/9
P4/4
P4/5
P4/6
P4/7
P4/8
P4/9
P4/10
P4/11
P4/12
P4/13
P3/8
P4/14
P4/32
P4/33
P4/34
P4/35
P3/32
P3/33
D-BUS
LSB
MSB

Fig 8

Fig 9

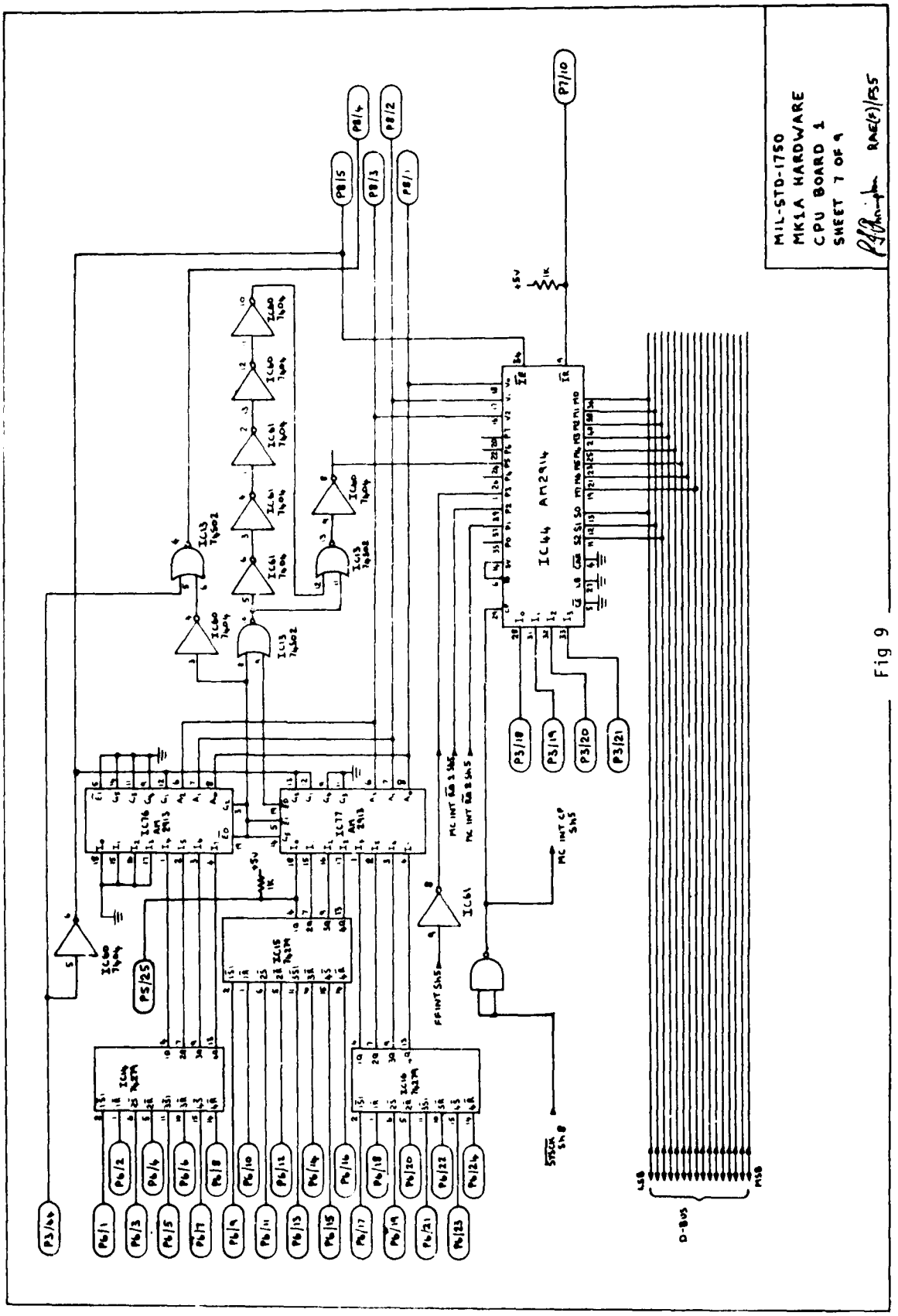

MIL-STD-1750
MK1A HARDWARE
CPU BOARD 1
SHEET 7 OF 9
IC44 AM2914
IC76 AM 2913
IC77 AM 2913
IC80 74S04
IC81 74S04
IC83 74S02
IC85 74S38
IC86 74S38
IC87 74S38
IC88 74S38
IC61
+5V
1K
P7/10
P8/4
P8/2
P8/5
P8/3
P8/1
P3/18
P3/19
P3/20
P3/21
P3/04
P5/25
P6/1
P6/2
P6/3
P6/4
P6/5
P6/6
P6/7
P6/8
P6/9
P6/10
P6/11
P6/12
P6/13
P6/14
P6/15
P6/16
P6/17
P6/18
P6/19
P6/20
P6/21
P6/22
P6/23
P6/24
D-BUS
MC INT CP SNS
FF INT SNS
STATUS SNS

Fig 9

Fig 10

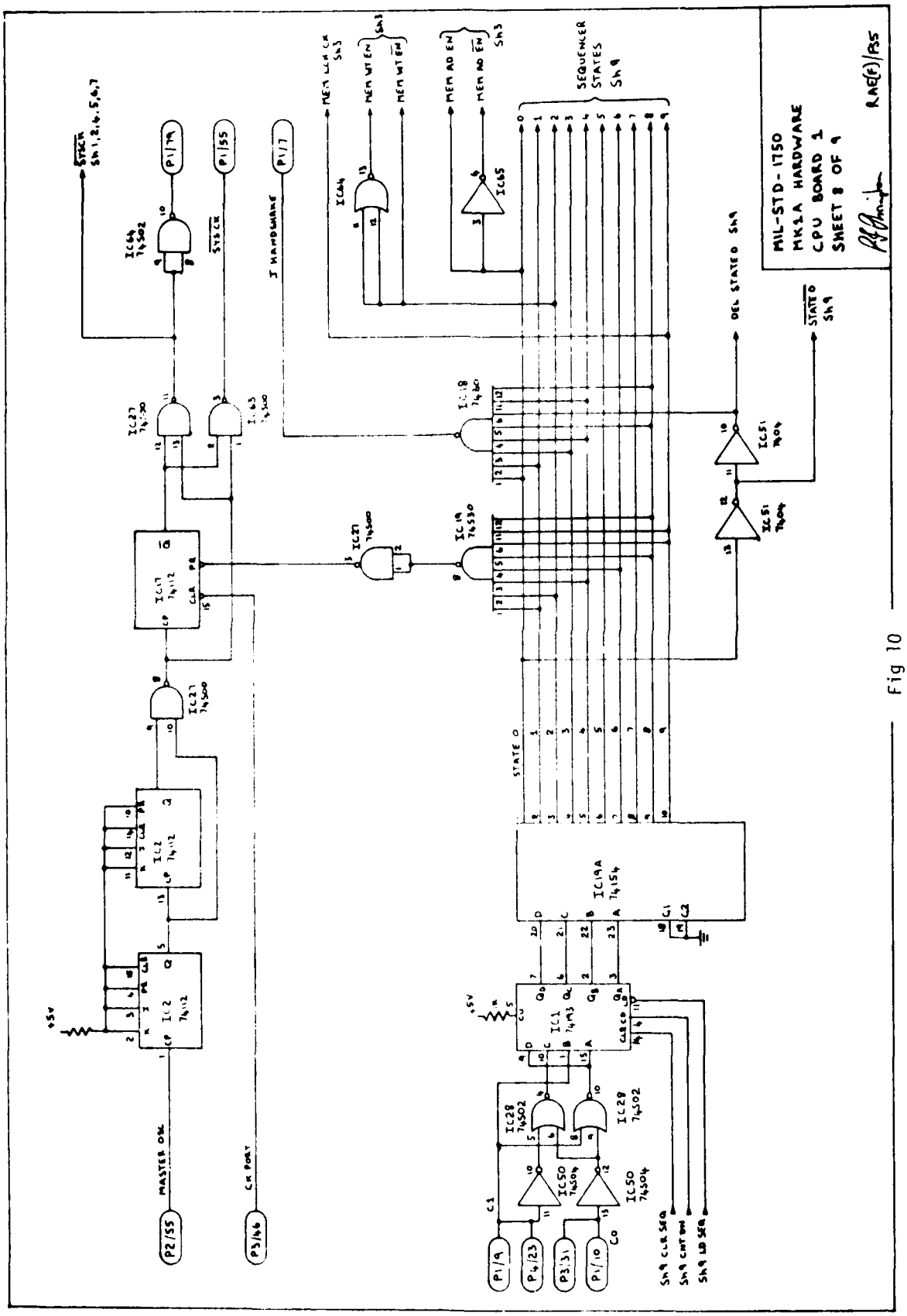

MIL-STD-1750
MK1A HARDWARE
CPU BOARD 1
SHEET 8 OF 9
RAE(E)/BS
SEQUENCER STATES SH9
MEM AD EN SH3
MEM WT EN SH3
MEM LOK CK SH3
STATE 0 SH9
DEL STATE 0 SH9
MASTER CLK
CLK PORT
P1/79
P1/55
P1/7
P2/55
P3/46
P1/9
P4/23
P3/31
P1/10
IC1 74193
IC2 74112
IC19A 74154
IC28 74S02
IC50 74S04
IC51 7404
IC64 74S02
IC65 7404
IC18 74S60
IC19 74S30
IC21 74S00
IC27 74S00

Fig 10

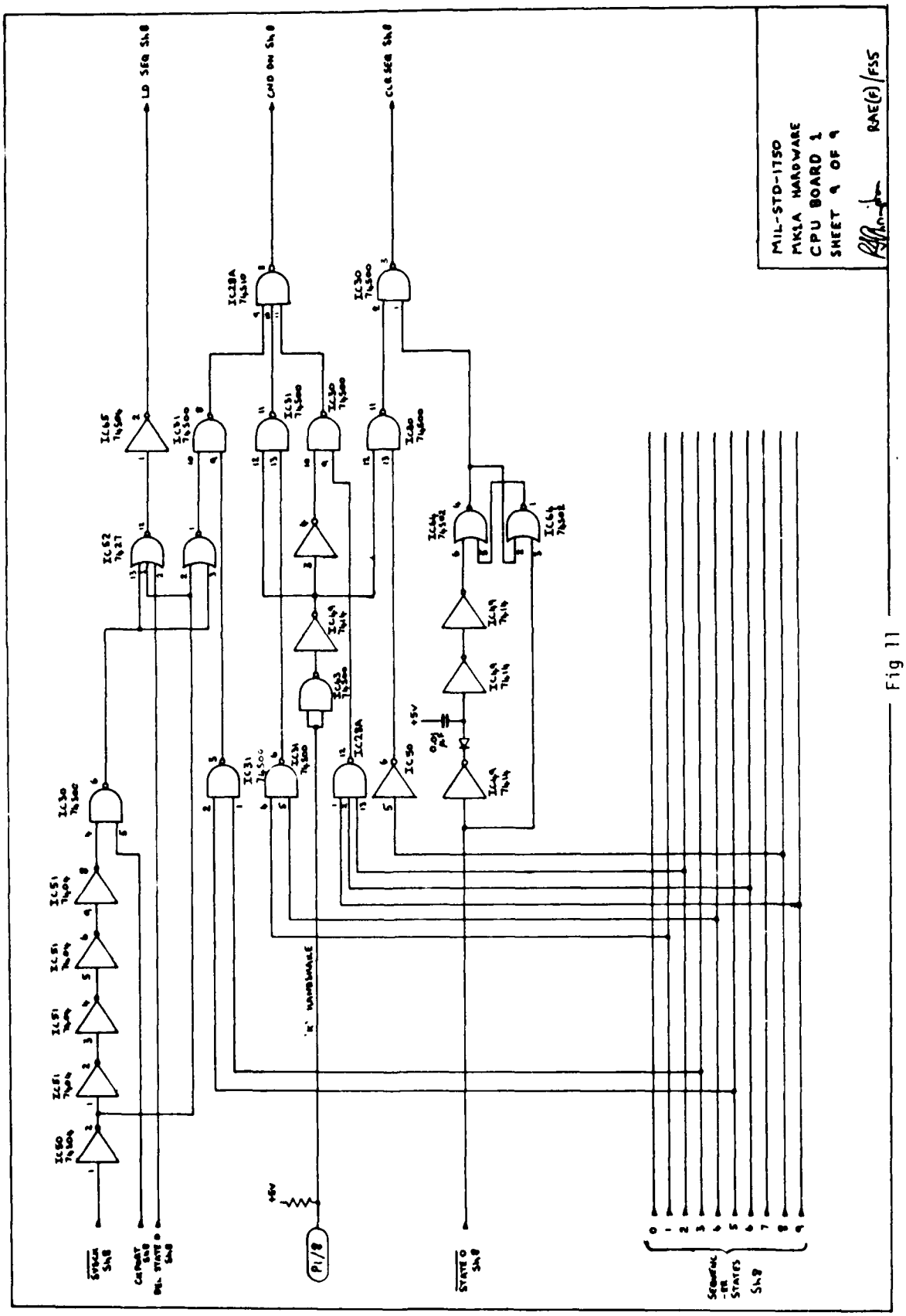

MIL-STD-1750
MKIA HARDWARE
CPU BOARD 1
SHEET 9 OF 9
RAE(F)/FSS
LD SEQ Sh8
CND BR Sh8
CLR SEQ Sh8
+5V
0.01 μF
+5V
P1/8
STATE 0 Sh8
0 1 2 3 4 5 6 7 8 9

Fig 11

Fig 12

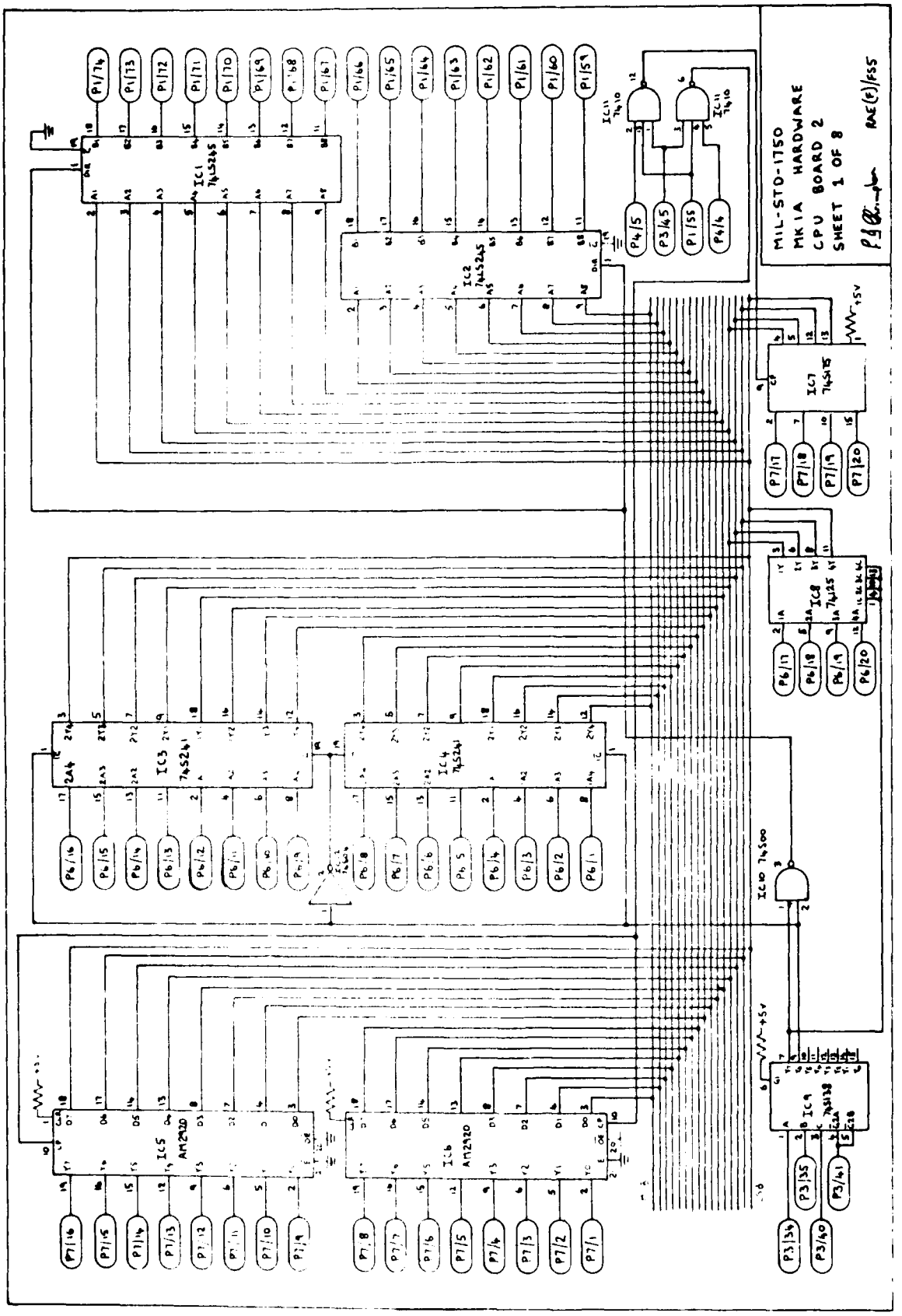

MIL-STD-1750
MK1A HARDWARE
CPU BOARD 2
SHEET 1 OF 8
IC1 74LS245
IC2 74LS245
IC3 74S241
IC4 74S241
IC5 AM2920
IC6 AM2920
IC7 74S175
IC8 74S125
IC9 74S138
IC10 74S00
IC11 74S10
+5V

Fig 12

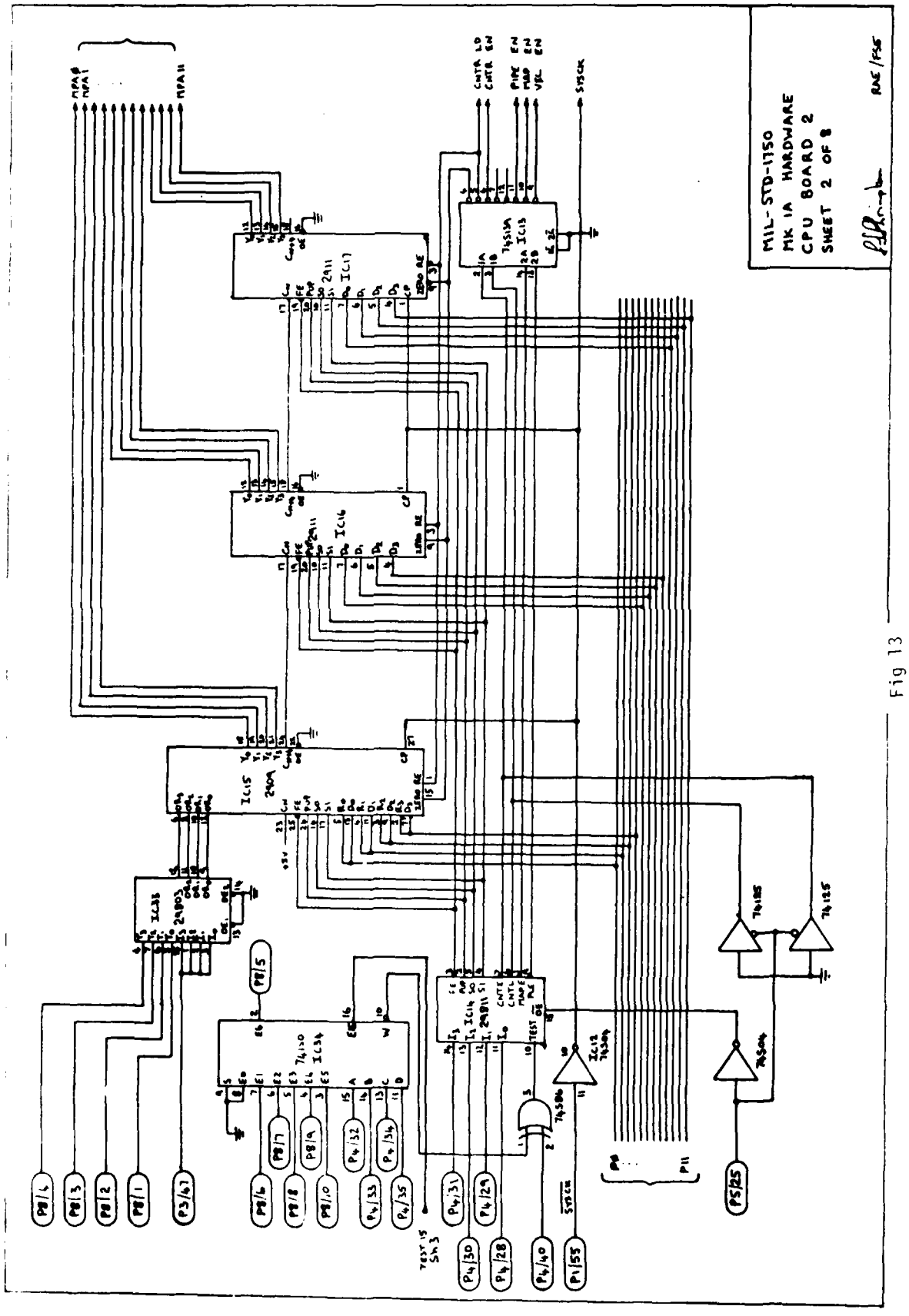

MIL-STD-1750
MK 1A HARDWARE
CPU BOARD 2
SHEET 2 OF 8
RAE/FSC
NPA0
NPA1
NPA11
SYSCK
IC17
IC16
IC15
IC14
IC13
IC12
IC11
74S138
74S30
74S04
74S125
P8/4
P8/3
P8/2
P8/1
P3/67
P8/5
P8/6
P8/7
P8/8
P8/9
P8/10
P4/32
P4/33
P4/34
P4/35
P4/31
P4/30
P4/29
P4/28
P4/40
P1/55
P5/25
P0
P11

Fig 13

Fig 14

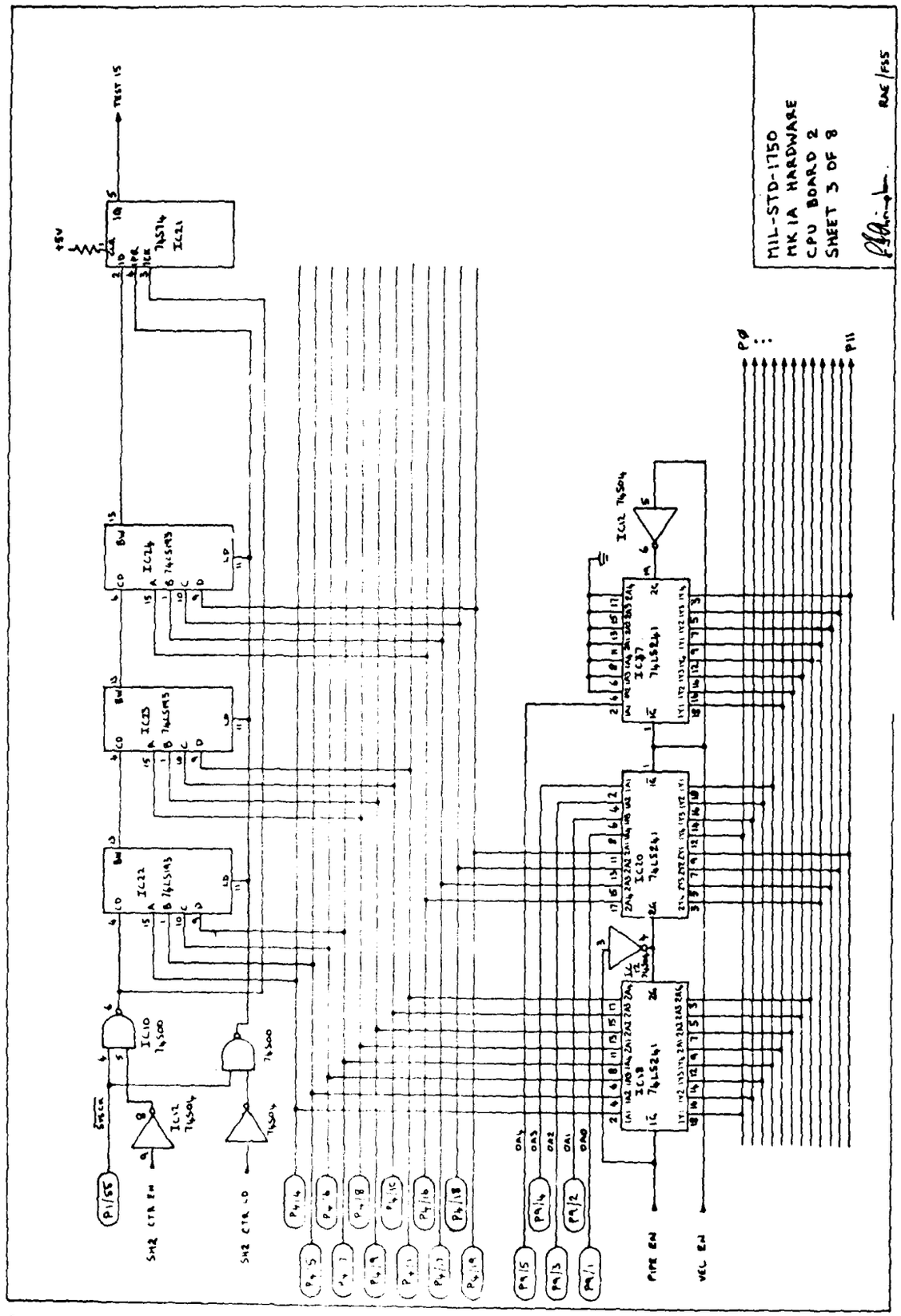

MIL-STD-1750
MK 1A HARDWARE
CPU BOARD 2
SHEET 3 OF 8
RAE/FSS
TEST 15
+5V
IC21 74S74
IC24 74LS193
IC23 74LS193
IC22 74LS193
IC10 74S00
74S00
IC12 74S04
74S04
P1/15
SH2 CTR EN
SH2 CTR LD
P4/4
P4/5
P4/6
P4/7
P4/8
P4/9
P4/10
P4/11
P4/12
P4/13
P4/14
P4/15
P4/16
P4/17
P4/18
P4/19
IC12 74S04
IC17 74LS241
IC18 74LS241
IC10 74LS241
IC11 74LS241
PIPE EN
VEC EN
P0
P11
DA3
DA2
DA1
DA0
P8/15
P8/14
P8/13
P8/12
P8/1

Fig 14

Fig 15

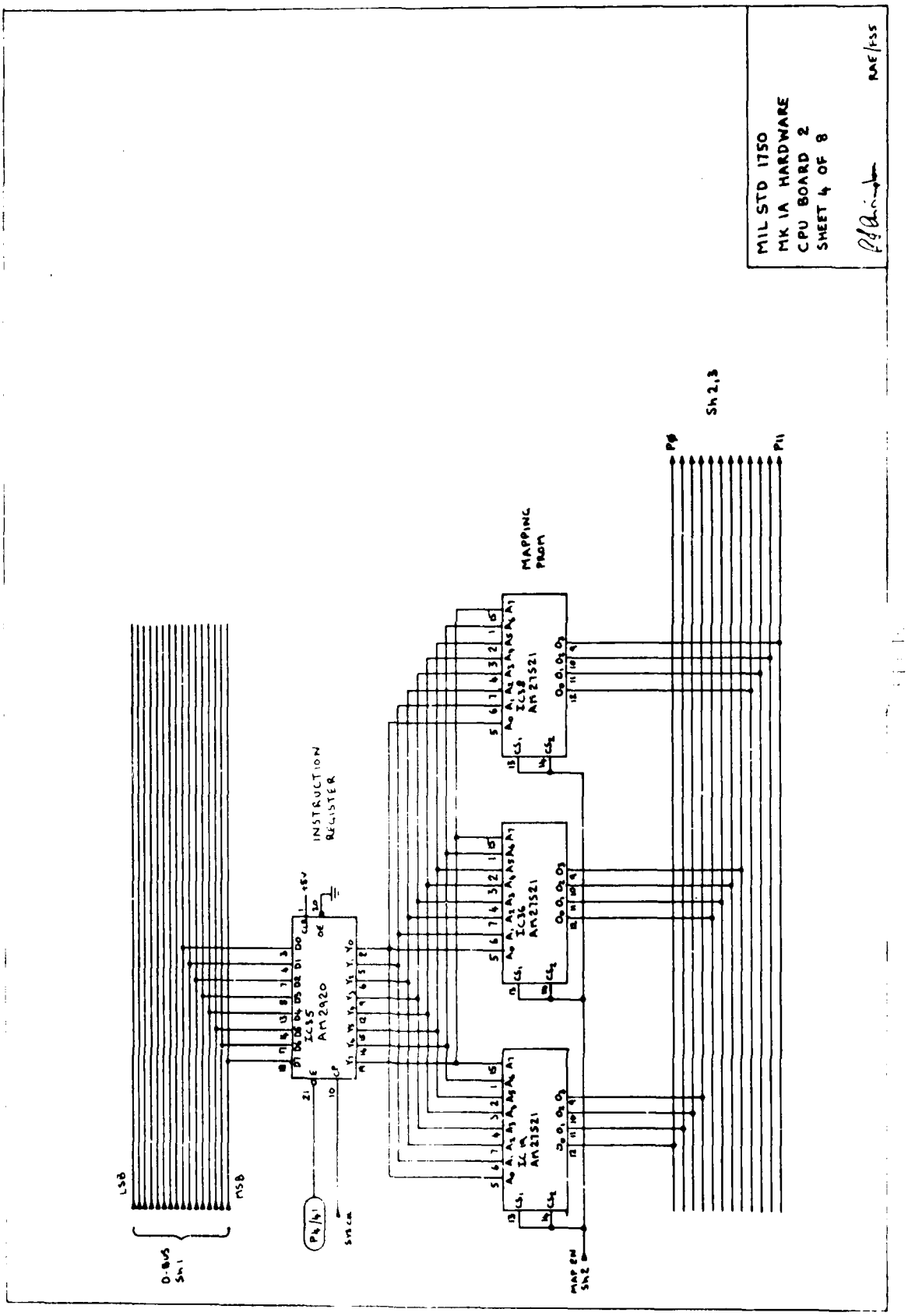

MIL STD 1750
MK 1A HARDWARE
CPU BOARD 2
SHEET 4 OF 8
RAE/1755
INSTRUCTION REGISTER
MAPPING PROM
IC35 AM2920
IC36 AM21521
IC37 AM21521
IC38 AM21521
LSB
MSB
D-BUS Sh1
P4/1
Sh1,2
Sh2,3
P0
P11
MAP EN Sh2

Fig 16

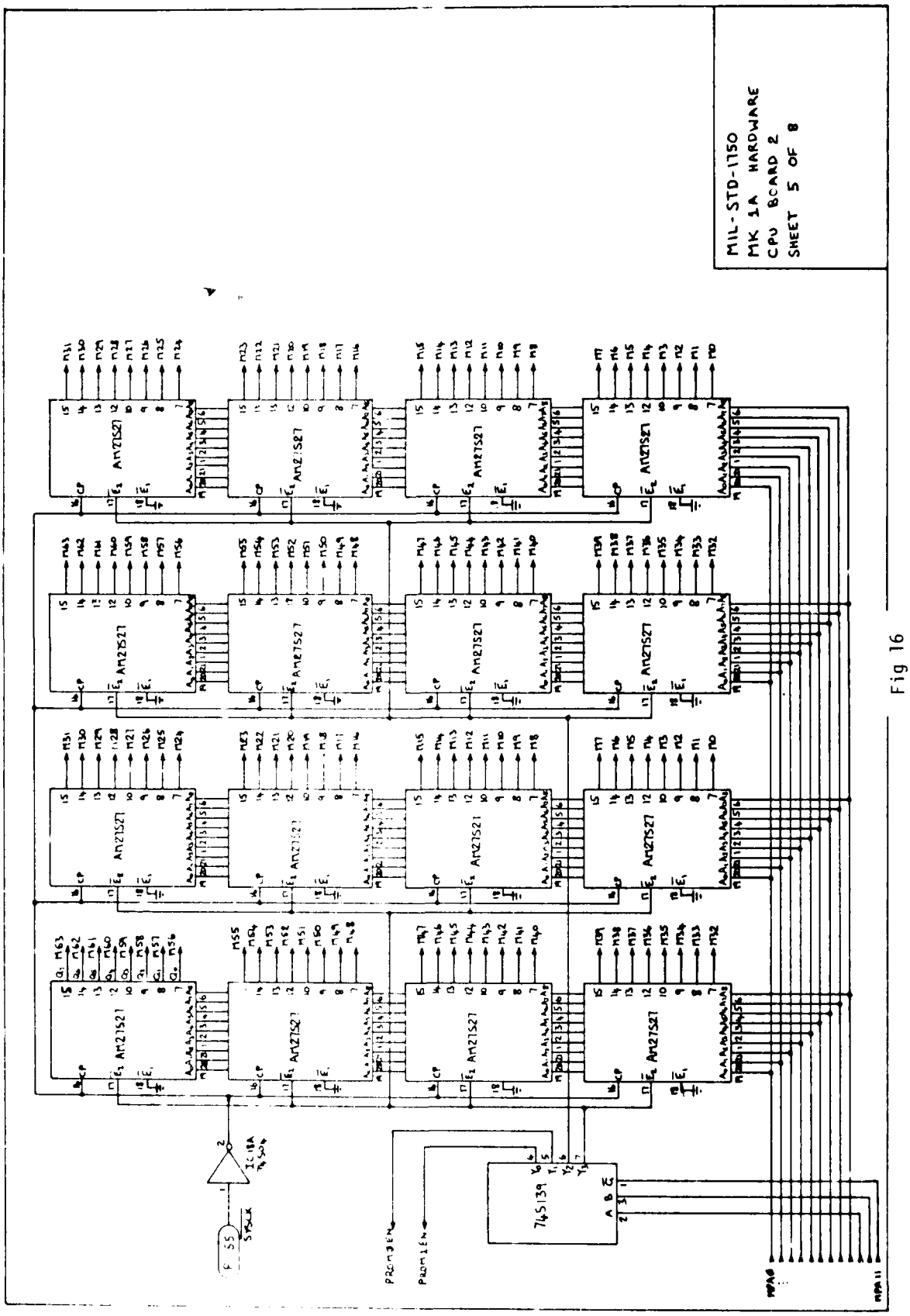

MIL-STD-1750
MK 1A HARDWARE
CPU BOARD 2
SHEET 5 OF 8
Am27S27
74S139
PROM EN
PROM EN

Fig 16

Fig 17

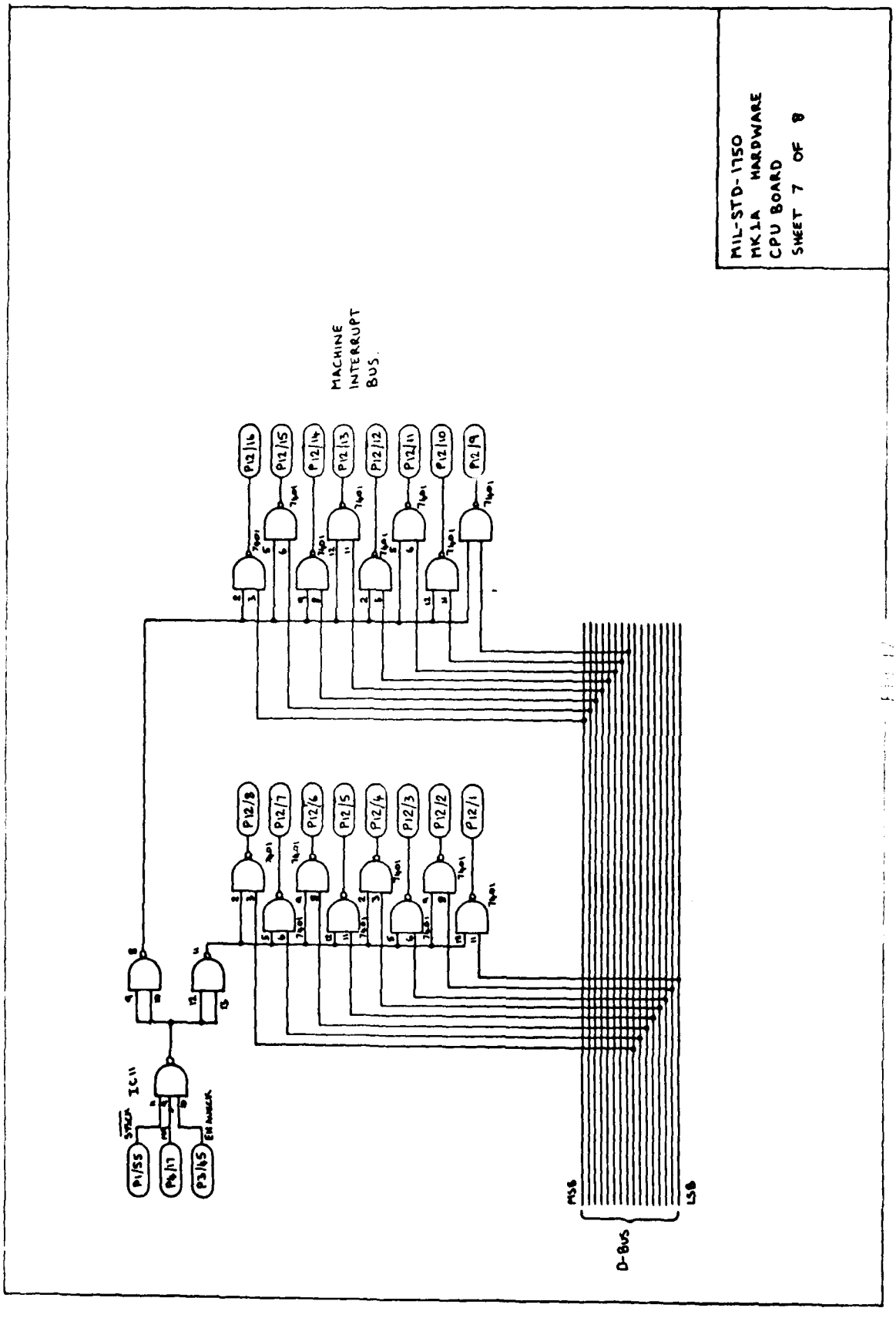

MIL-STD-1750
MK1A HARDWARE
CPU BOARD
SHEET 7 OF 8
MACHINE
INTERRUPT
BUS
P12/16
P12/15
P12/14
P12/13
P12/12
P12/11
P12/10
P12/9
P12/8
P12/7
P12/6
P12/5
P12/4
P12/3
P12/2
P12/1
IC11
P1/55
P3/45
MSB
LSB
D-Bus

Fig 18

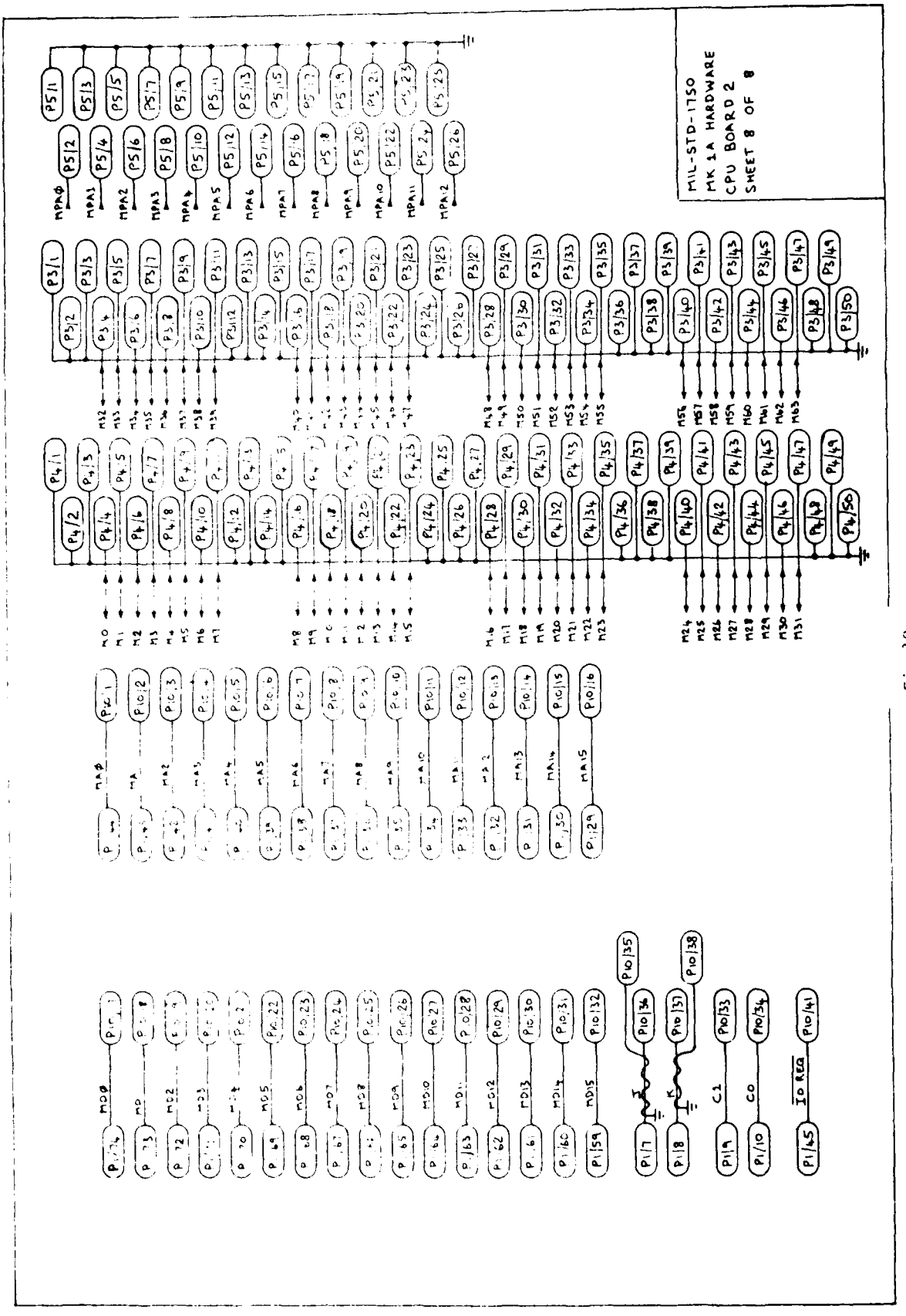


Fig 18

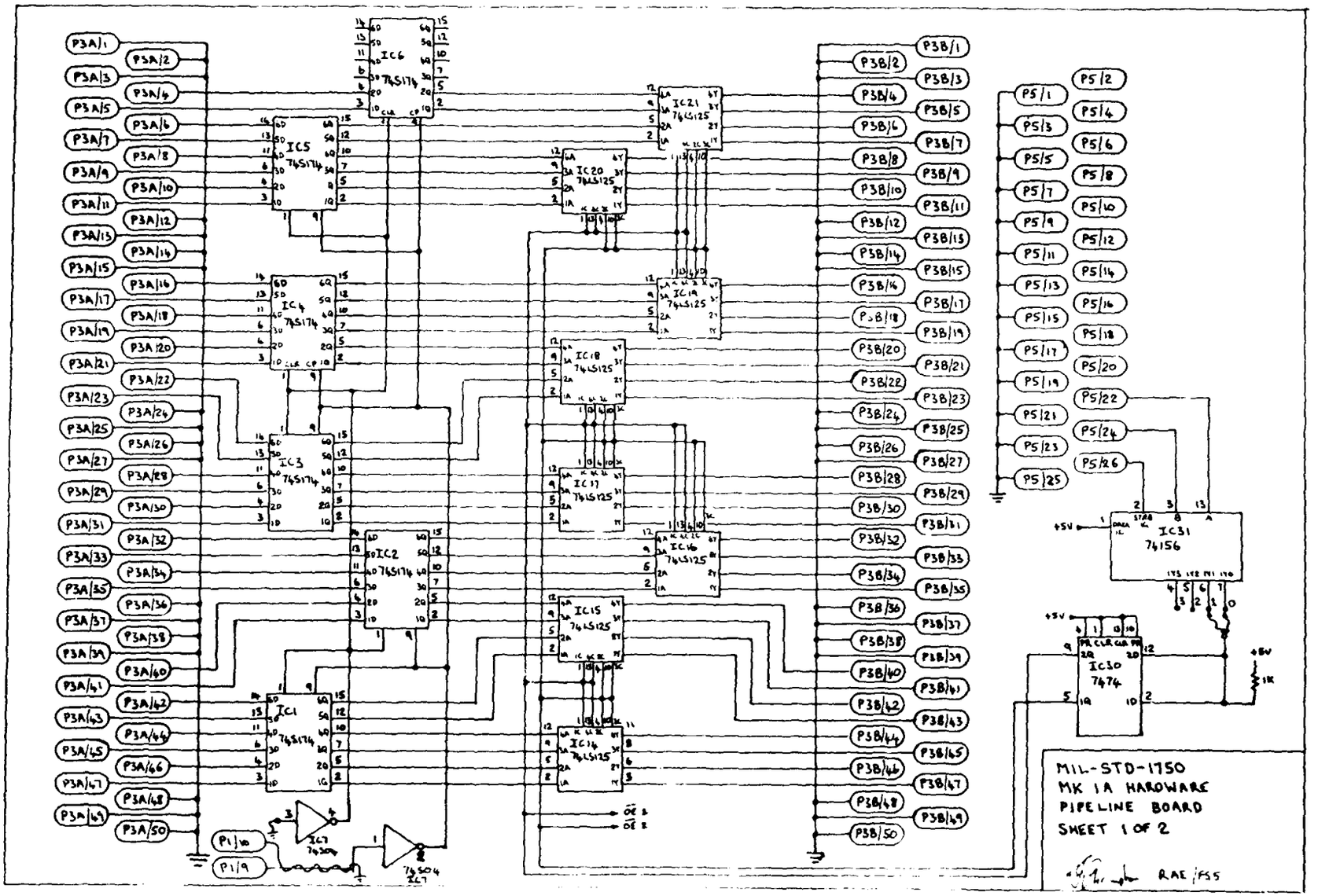

P3A/1
P3A/2
P3A/3
P3A/4
P3A/5
P3A/6
P3A/7
P3A/8
P3A/9
P3A/10
P3A/11
P3A/12
P3A/13
P3A/14
P3A/15
P3A/16
P3A/17
P3A/18
P3A/19
P3A/20
P3A/21
P3A/22
P3A/23
P3A/24
P3A/25
P3A/26
P3A/27
P3A/28
P3A/29
P3A/30
P3A/31
P3A/32
P3A/33
P3A/34
P3A/35
P3A/36
P3A/37
P3A/38
P3A/39
P3A/40
P3A/41
P3A/42
P3A/43
P3A/44
P3A/45
P3A/46
P3A/47
P3A/48
P3A/49
P3A/50
IC6 74S174
IC5 74S174
IC4 74S174
IC3 74S174
IC2 74S174
IC1 74S174
IC7 74S04
P1/10
P1/9
IC21 74LS125
IC20 74LS125
IC19 74LS125
IC18 74LS125
IC17 74LS125
IC16 74LS125
IC15 74LS125
IC14 74LS125
P3B/1
P3B/2
P3B/3
P3B/4
P3B/5
P3B/6
P3B/7
P3B/8
P3B/9
P3B/10
P3B/11
P3B/12
P3B/13
P3B/14
P3B/15
P3B/16
P3B/17
P3B/18
P3B/19
P3B/20
P3B/21
P3B/22
P3B/23
P3B/24
P3B/25
P3B/26
P3B/27
P3B/28
P3B/29
P3B/30
P3B/31
P3B/32
P3B/33
P3B/34
P3B/35
P3B/36
P3B/37
P3B/38
P3B/39
P3B/40
P3B/41
P3B/42
P3B/43
P3B/44
P3B/45
P3B/46
P3B/47
P3B/48
P3B/49
P3B/50
P5/1
P5/2
P5/3
P5/4
P5/5
P5/6
P5/7
P5/8
P5/9
P5/10
P5/11
P5/12
P5/13
P5/14
P5/15
P5/16
P5/17
P5/18
P5/19
P5/20
P5/21
P5/22
P5/23
P5/24
P5/25
P5/26
+5V
IC31 74156
IC30 7474
1K
MIL-STD-1750
MK 1A HARDWARE
PIPELINE BOARD
SHEET 1 OF 2
RAE/FSS

Fig 10

Fig 20

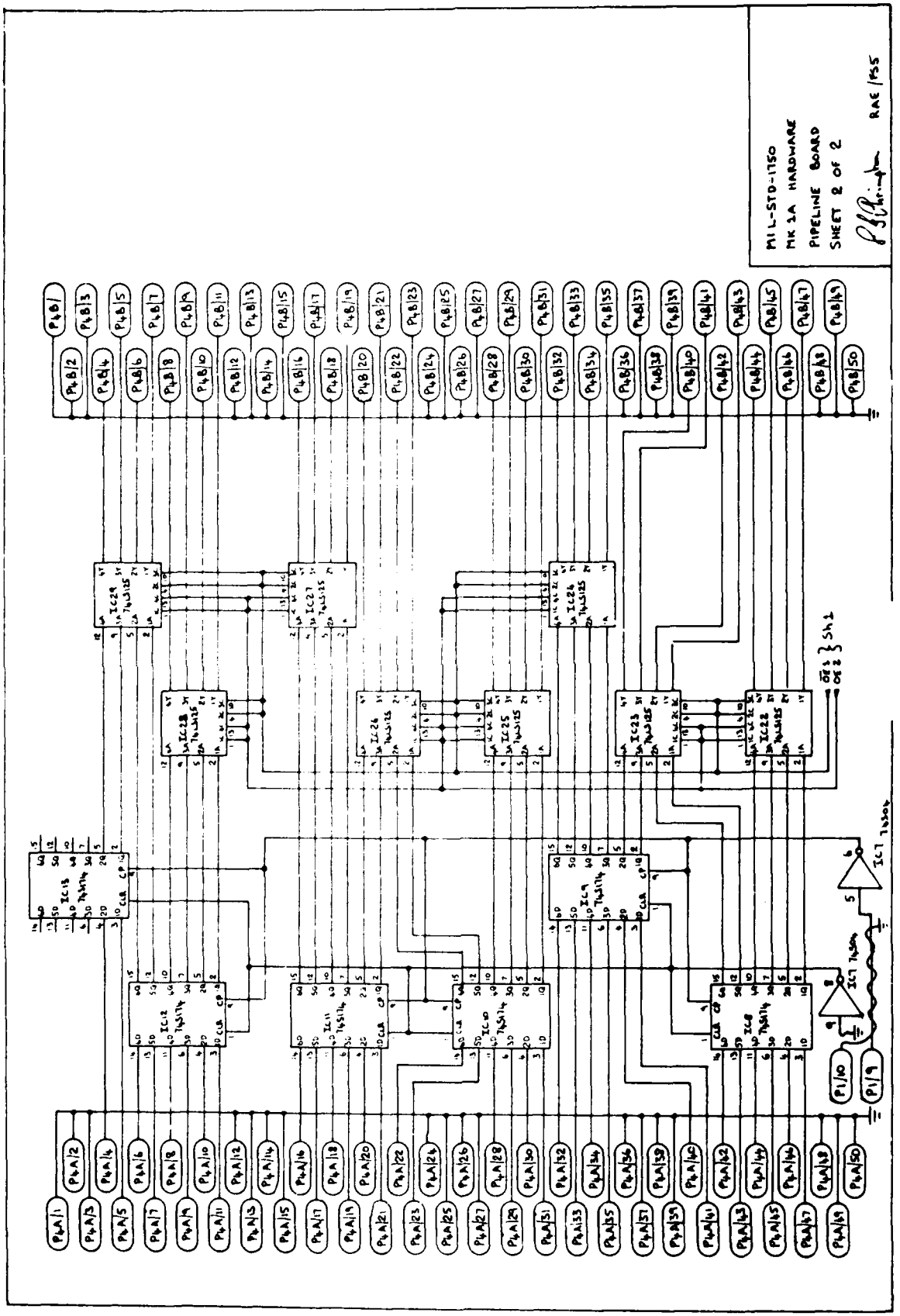

MIL-STD-1750
MK 1A HARDWARE
PIPELINE BOARD
SHEET 2 OF 2
RAE/MSS

Fig 20

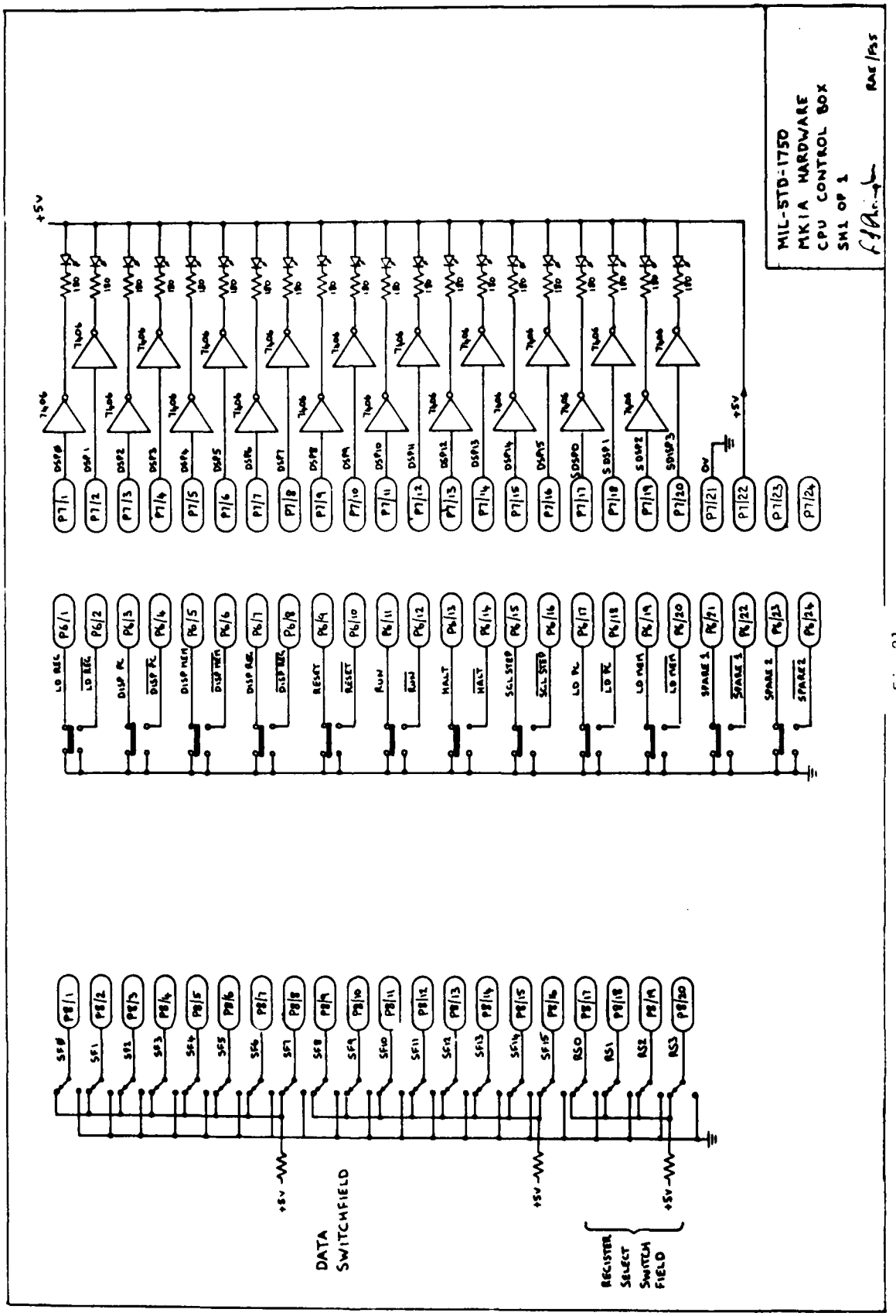

MIL-STD-1750
MKIA HARDWARE
CPU CONTROL BOX
SHT OF
DATA
SWITCHFIELD
REGISTER
SELECT
SWITCH
FIELD

Fig 21

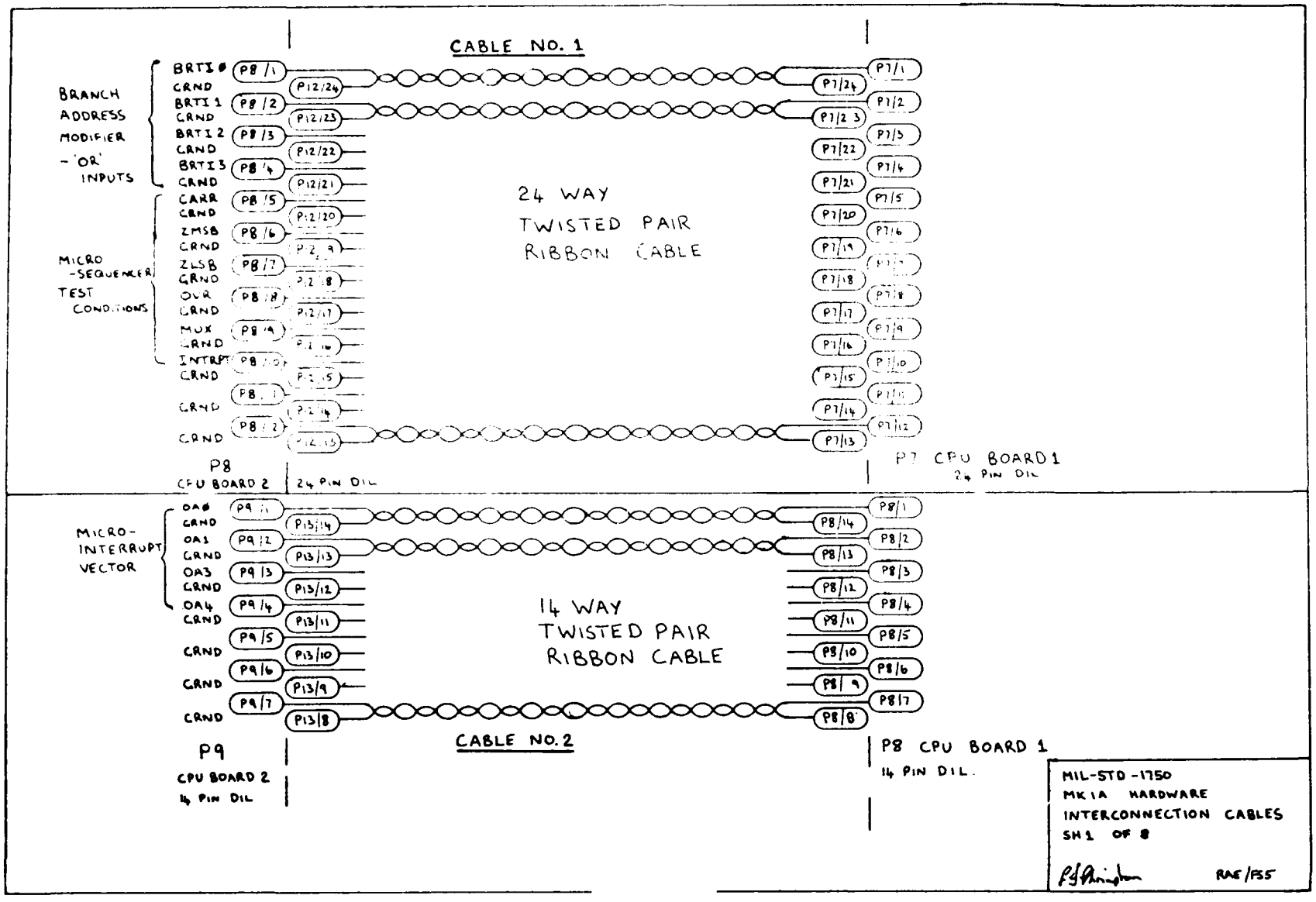

CABLE NO. 1
BRANCH ADDRESS MODIFIER - 'OR' INPUTS
MICRO-SEQUENCER TEST CONDITIONS
BRTI 0
BRTI 1
BRTI 2
BRTI 3
CARR
ZMSB
ZLSB
OVR
MUX
INTRPT
GRND
24 WAY TWISTED PAIR RIBBON CABLE
P8 CPU BOARD 2
24 PIN DIL
P7 CPU BOARD 1
24 PIN DIL
MICRO-INTERRUPT VECTOR
OA0
OA1
OA3
OA4
14 WAY TWISTED PAIR RIBBON CABLE
CABLE NO. 2
P9
CPU BOARD 2
14 PIN DIL
P8 CPU BOARD 1
14 PIN DIL
MIL-STD-1750
MK1A HARDWARE
INTERCONNECTION CABLES
SH 1 OF 8
RAE/PSS

Fig 22

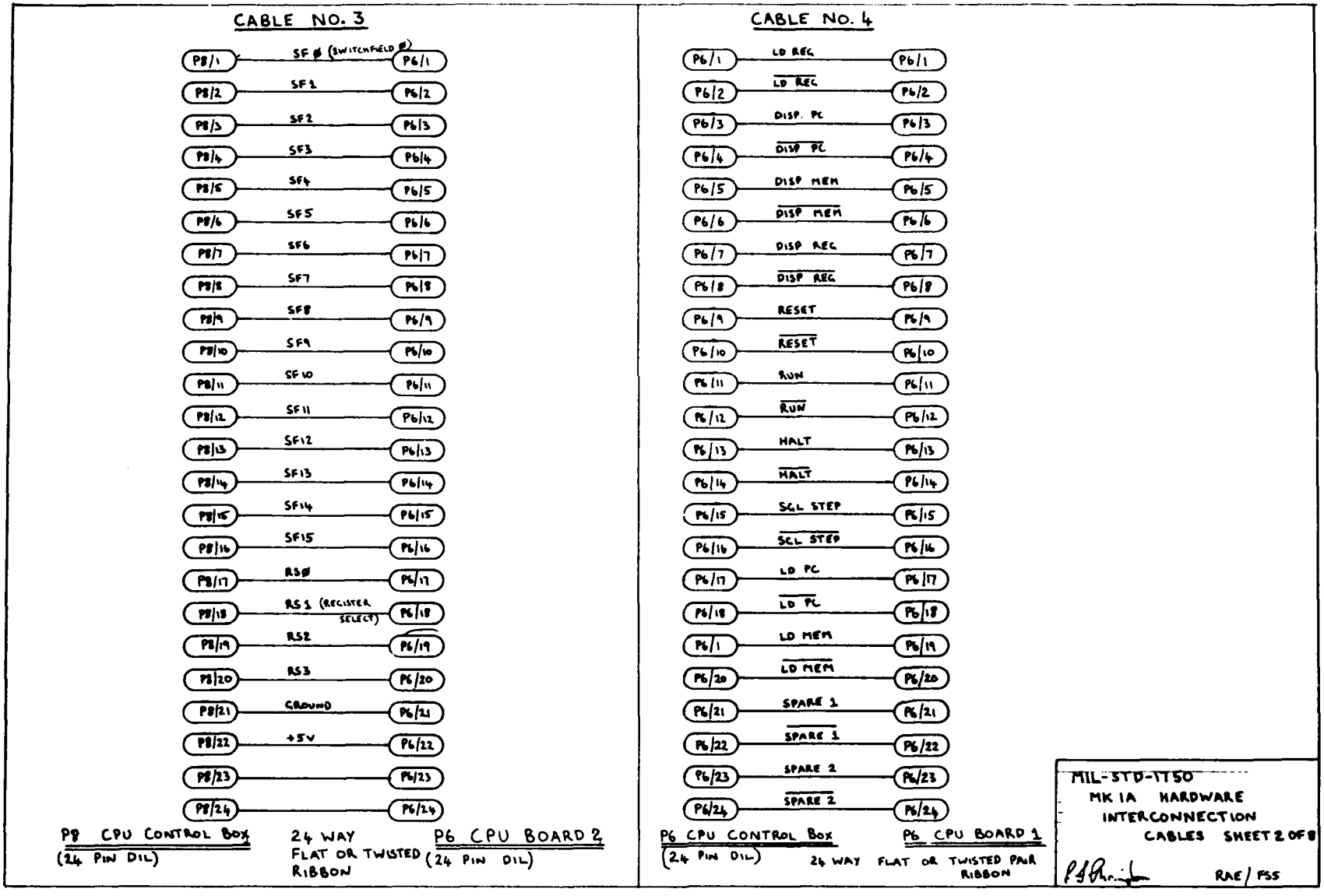
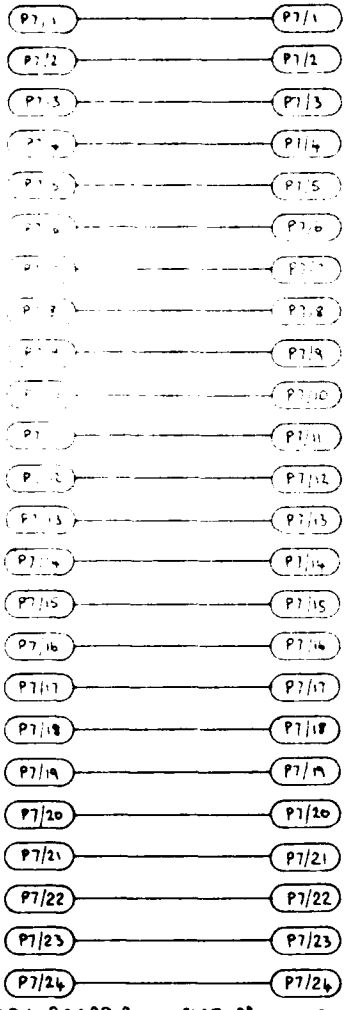

CABLE NO. 3
P8/1 SF0 (SWITCHFIELD 0) P6/1
P8/2 SF1 P6/2
P8/3 SF2 P6/3
P8/4 SF3 P6/4
P8/5 SF4 P6/5
P8/6 SF5 P6/6
P8/7 SF6 P6/7
P8/8 SF7 P6/8
P8/9 SF8 P6/9
P8/10 SF9 P6/10
P8/11 SF10 P6/11
P8/12 SF11 P6/12
P8/13 SF12 P6/13
P8/14 SF13 P6/14
P8/15 SF14 P6/15
P8/16 SF15 P6/16
P8/17 RS0 P6/17
P8/18 RS1 (REGISTER SELECT) P6/18
P8/19 RS2 P6/19
P8/20 RS3 P6/20
P8/21 GROUND P6/21
P8/22 +5V P6/22
P8/23 P6/23
P8/24 P6/24
P8 CPU CONTROL BOX (24 PIN DIL)
24 WAY FLAT OR TWISTED RIBBON
P6 CPU BOARD 2 (24 PIN DIL)
CABLE NO. 4
P6/1 LD REC P6/1
P6/2 LD REC P6/2
P6/3 DISP PC P6/3
P6/4 DISP PC P6/4
P6/5 DISP MEM P6/5
P6/6 DISP MEM P6/6
P6/7 DISP REG P6/7
P6/8 DISP REG P6/8
P6/9 RESET P6/9
P6/10 RESET P6/10
P6/11 RUN P6/11
P6/12 RUN P6/12
P6/13 HALT P6/13
P6/14 HALT P6/14
P6/15 SGL STEP P6/15
P6/16 SGL STEP P6/16
P6/17 LD PC P6/17
P6/18 LD PC P6/18
P6/1 LD MEM P6/19
P6/20 LD MEM P6/20
P6/21 SPARE 1 P6/21
P6/22 SPARE 1 P6/22
P6/23 SPARE 2 P6/23
P6/24 SPARE 2 P6/24
P6 CPU CONTROL BOX (24 PIN DIL)
24 WAY FLAT OR TWISTED PAIR RIBBON
P6 CPU BOARD 1
MIL-STD-1750
MK 1A HARDWARE INTERCONNECTION CABLES SHEET 2 OF 8
RAE/FSS

Fig 23

Fig 23

CABLE NO. 5

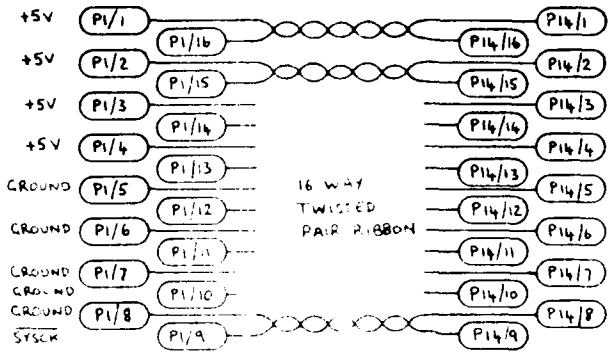


P7 CPU BOARD 2 (24 PIN DIL)

FLAT OR TWISTED 24 WAY RIBBON

P7 CPU CONTROL BOX (24 PIN DIL)

CABLE NO 6



P1 PIPELINE BOARD (16 PIN DIL)

P14 COMPUTER CONTROL UNIT BOARD (16 PIN DIL)

MIL-STD-1750
MK1A HARDWARE
INTERCONNECTION CABLES
SHEET 3 OF 8

RAE/FSS

Fig 24

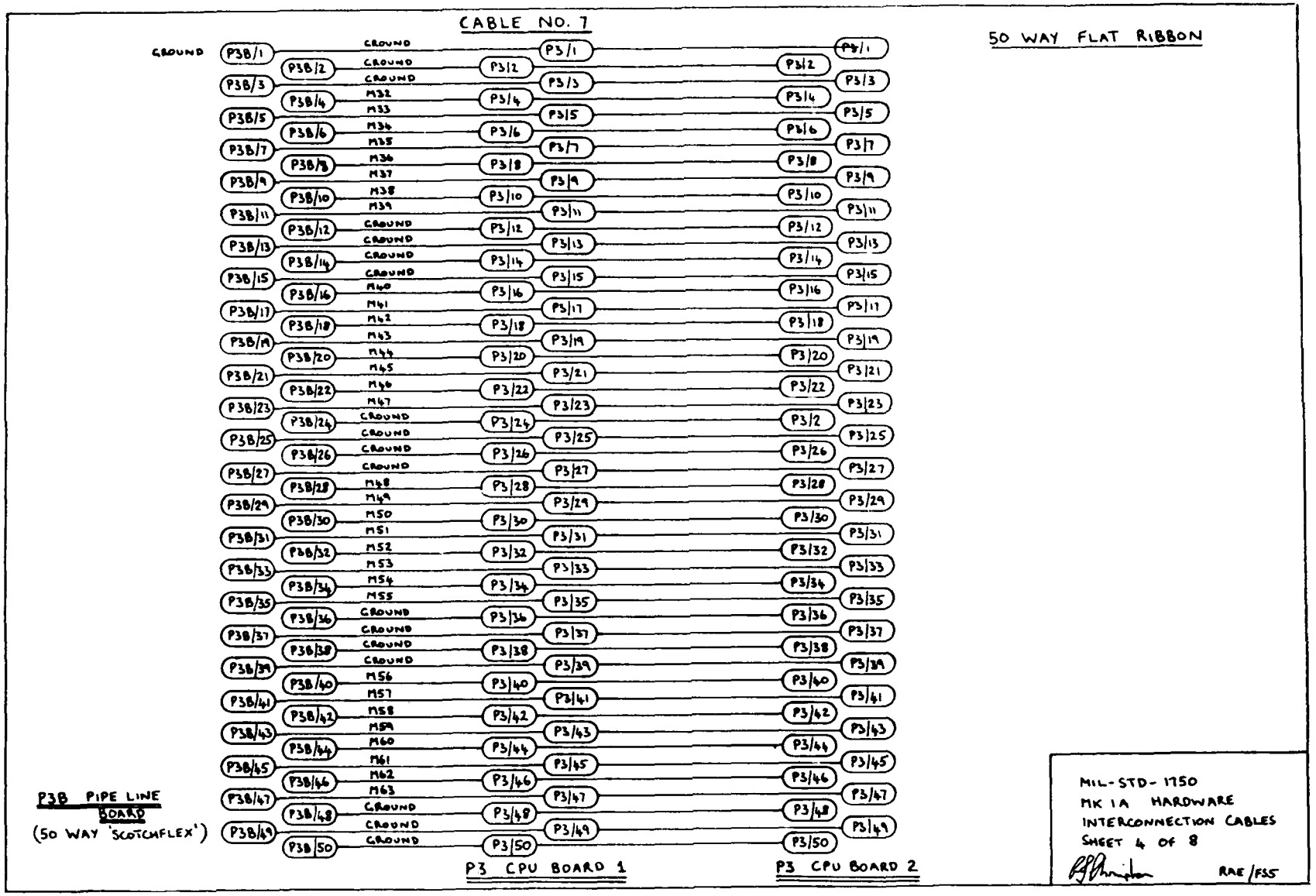

CABLE NO. 7
50 WAY FLAT RIBBON
GROUND
P3B/1
P3B/2
P3B/3
P3B/4
P3B/5
P3B/6
P3B/7
P3B/8
P3B/9
P3B/10
P3B/11
P3B/12
P3B/13
P3B/14
P3B/15
P3B/16
P3B/17
P3B/18
P3B/19
P3B/20
P3B/21
P3B/22
P3B/23
P3B/24
P3B/25
P3B/26
P3B/27
P3B/28
P3B/29
P3B/30
P3B/31
P3B/32
P3B/33
P3B/34
P3B/35
P3B/36
P3B/37
P3B/38
P3B/39
P3B/40
P3B/41
P3B/42
P3B/43
P3B/44
P3B/45
P3B/46
P3B/47
P3B/48
P3B/49
P3B/50
GROUND
M32
M33
M34
M35
M36
M37
M38
M39
GROUND
GROUND
GROUND
M40
M41
M42
M43
M44
M45
M46
M47
GROUND
GROUND
GROUND
M48
M49
M50
M51
M52
M53
M54
M55
GROUND
GROUND
GROUND
M56
M57
M58
M59
M60
M61
M62
M63
GROUND
GROUND
GROUND
P3/1
P3/2
P3/3
P3/4
P3/5
P3/6
P3/7
P3/8
P3/9
P3/10
P3/11
P3/12
P3/13
P3/14
P3/15
P3/16
P3/17
P3/18
P3/19
P3/20
P3/21
P3/22
P3/23
P3/24
P3/25
P3/26
P3/27
P3/28
P3/29
P3/30
P3/31
P3/32
P3/33
P3/34
P3/35
P3/36
P3/37
P3/38
P3/39
P3/40
P3/41
P3/42
P3/43
P3/44
P3/45
P3/46
P3/47
P3/48
P3/49
P3/50
P3B PIPE LINE BOARD
(50 WAY 'SCOTCHFLEX')
P3 CPU BOARD 1
P3 CPU BOARD 2
MIL-STD-1750
MK 1A HARDWARE
INTERCONNECTION CABLES
SHEET 4 OF 8
RAE/FSS

Fig 25

Fig 25

Fig 26

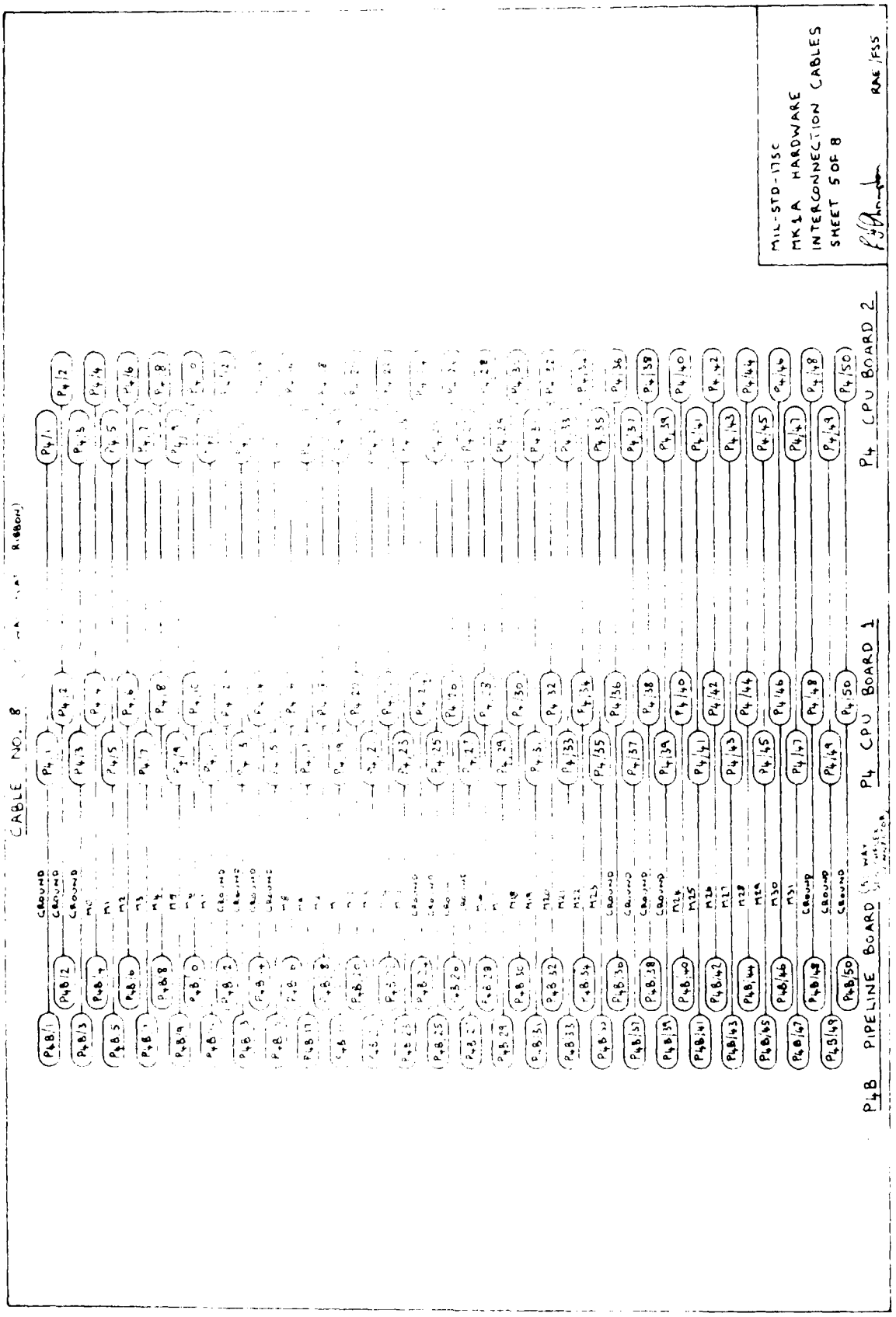

CABLE NO. 8
P4B PIPELINE BOARD
P4 CPU BOARD 1
P4 CPU BOARD 2
MIL-STD-1553
MK1A HARDWARE
INTERCONNECTION CABLES
SHEET 5 OF 8
RAE/FSS

Fig 26

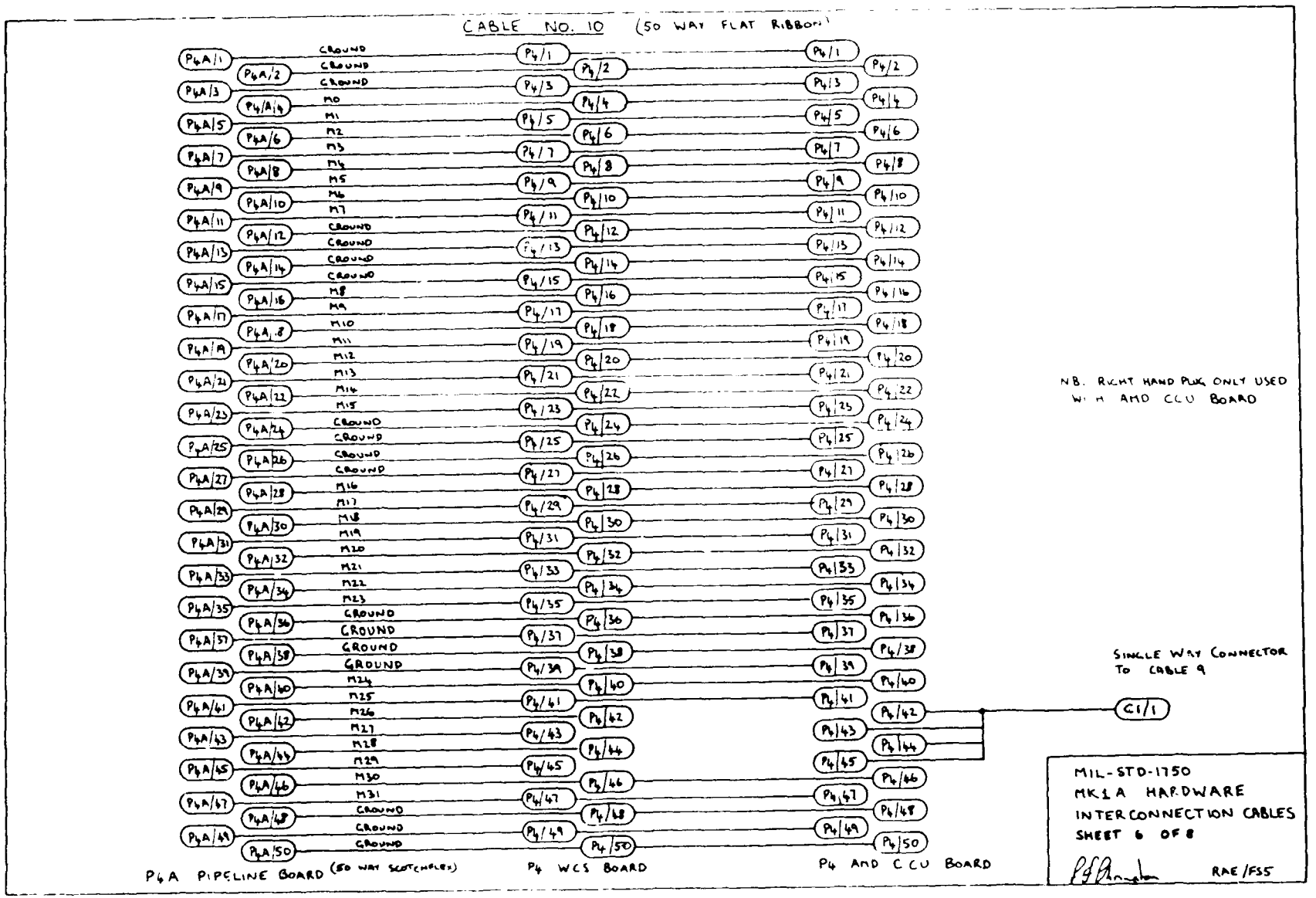

CABLE NO. 10 (50 WAY FLAT RIBBON)
P4A/1 GROUND
P4A/2 GROUND
P4A/3 GROUND
P4A/4 M0
P4A/5 M1
P4A/6 M2
P4A/7 M3
P4A/8 M4
P4A/9 M5
P4A/10 M6
P4A/11 M7
P4A/12 GROUND
P4A/13 GROUND
P4A/14 GROUND
P4A/15 GROUND
P4A/16 M8
P4A/17 M9
P4A/18 M10
P4A/19 M11
P4A/20 M12
P4A/21 M13
P4A/22 M14
P4A/23 M15
P4A/24 GROUND
P4A/25 GROUND
P4A/26 GROUND
P4A/27 GROUND
P4A/28 M16
P4A/29 M17
P4A/30 M18
P4A/31 M19
P4A/32 M20
P4A/33 M21
P4A/34 M22
P4A/35 M23
P4A/36 GROUND
P4A/37 GROUND
P4A/38 GROUND
P4A/39 GROUND
P4A/40 M24
P4A/41 M25
P4A/42 M26
P4A/43 M27
P4A/44 M28
P4A/45 M29
P4A/46 M30
P4A/47 M31
P4A/48 GROUND
P4A/49 GROUND
P4A/50 GROUND
P4/1 P4/2 P4/3 P4/4 P4/5 P4/6 P4/7 P4/8 P4/9 P4/10 P4/11 P4/12 P4/13 P4/14 P4/15 P4/16 P4/17 P4/18 P4/19 P4/20 P4/21 P4/22 P4/23 P4/24 P4/25 P4/26 P4/27 P4/28 P4/29 P4/30 P4/31 P4/32 P4/33 P4/34 P4/35 P4/36 P4/37 P4/38 P4/39 P4/40 P4/41 P4/42 P4/43 P4/44 P4/45 P4/46 P4/47 P4/48 P4/49 P4/50
P4/1 P4/2 P4/3 P4/4 P4/5 P4/6 P4/7 P4/8 P4/9 P4/10 P4/11 P4/12 P4/13 P4/14 P4/15 P4/16 P4/17 P4/18 P4/19 P4/20 P4/21 P4/22 P4/23 P4/24 P4/25 P4/26 P4/27 P4/28 P4/29 P4/30 P4/31 P4/32 P4/33 P4/34 P4/35 P4/36 P4/37 P4/38 P4/39 P4/40 P4/41 P4/42 P4/43 P4/44 P4/45 P4/46 P4/47 P4/48 P4/49 P4/50
NB. RIGHT HAND PLUG ONLY USED WITH AMD CCU BOARD
SINGLE WAY CONNECTOR TO CABLE 9
C1/1
P4A PIPELINE BOARD (50 WAY SCOTCHFLEX)
P4 WCS BOARD
P4 AMD CCU BOARD
MIL-STD-1750
MK1A HARDWARE
INTERCONNECTION CABLES
SHEET 6 OF 8
RAE/FSS

Fig 27

Fig 27

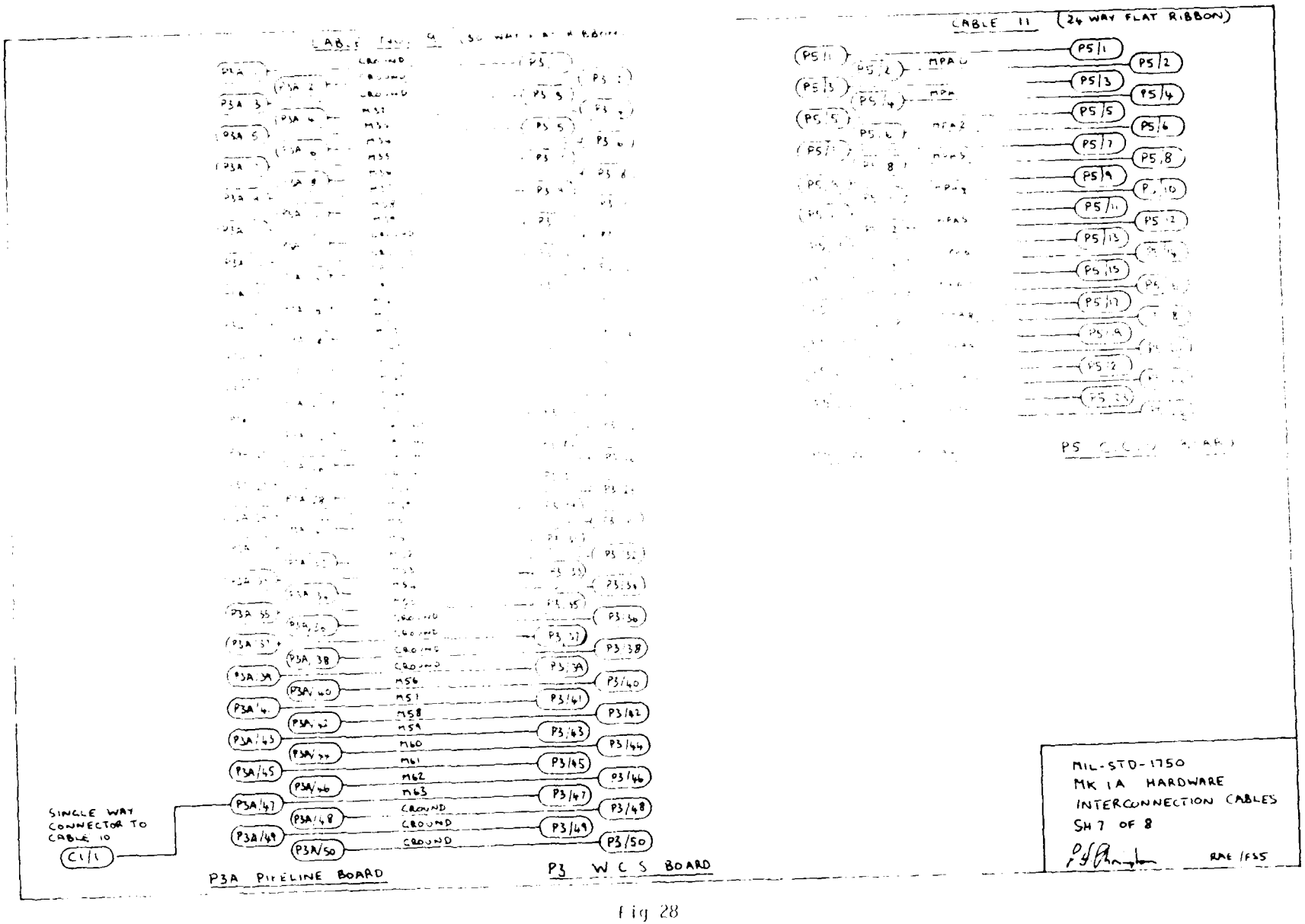

CABLE 11 (24 WAY FLAT RIBBON)
P5/1
P5/2
P5/3
P5/4
P5/5
P5/6
P5/7
P5/8
P5/9
P5/10
P5/11
P5/12
P5/13
P5/14
P5/15
P5/16
P5/17
P5/18
P5/19
P5/20
P5/21
P5/22
P5/23
P5/24
P3A/1
P3A/2
P3A/3
P3A/4
P3A/5
P3A/6
P3A/7
P3A/8
P3A/9
P3A/10
P3A/11
P3A/12
P3A/13
P3A/14
P3A/15
P3A/16
P3A/17
P3A/18
P3A/19
P3A/20
P3A/21
P3A/22
P3A/23
P3A/24
P3A/25
P3A/26
P3A/27
P3A/28
P3A/29
P3A/30
P3A/31
P3A/32
P3A/33
P3A/34
P3A/35
P3A/36
P3A/37
P3A/38
P3A/39
P3A/40
P3A/41
P3A/42
P3A/43
P3A/44
P3A/45
P3A/46
P3A/47
P3A/48
P3A/49
P3A/50
GROUND
MS6
MS5
MS4
MS3
MS2
MS1
MS0
MA0
MA1
MA2
MA3
MA4
MA5
MA6
MA7
MA8
MA9
MA10
MA11
MA12
MA13
MA14
MA15
MD0
MD1
MD2
MD3
MD4
MD5
MD6
MD7
MD8
MD9
MD10
MD11
MD12
MD13
MD14
MD15
P3/1
P3/2
P3/3
P3/4
P3/5
P3/6
P3/7
P3/8
P3/9
P3/10
P3/11
P3/12
P3/13
P3/14
P3/15
P3/16
P3/17
P3/18
P3/19
P3/20
P3/21
P3/22
P3/23
P3/24
P3/25
P3/26
P3/27
P3/28
P3/29
P3/30
P3/31
P3/32
P3/33
P3/34
P3/35
P3/36
P3/37
P3/38
P3/39
P3/40
P3/41
P3/42
P3/43
P3/44
P3/45
P3/46
P3/47
P3/48
P3/49
P3/50
SINGLE WAY CONNECTOR TO CABLE 10
C1/1
P3A PIPELINE BOARD
P3 W C S BOARD
MIL-STD-1750
MK 1A HARDWARE
INTERCONNECTION CABLES
SH 7 OF 8
RAE IF35

Fig 28

Fig 28

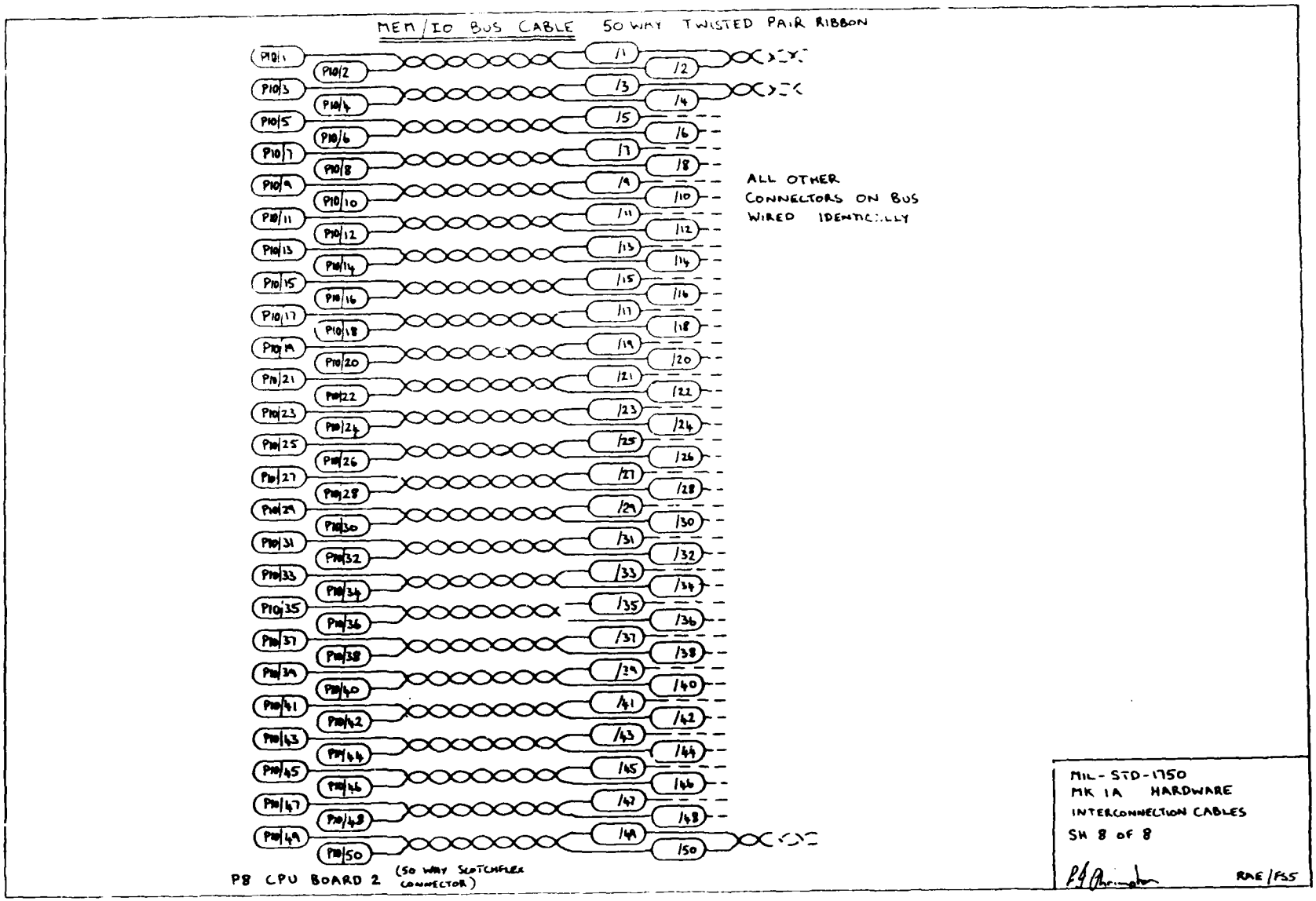

MEM/IO BUS CABLE 50 WAY TWISTED PAIR RIBBON
P10/1
P10/2
P10/3
P10/4
P10/5
P10/6
P10/7
P10/8
P10/9
P10/10
P10/11
P10/12
P10/13
P10/14
P10/15
P10/16
P10/17
P10/18
P10/19
P10/20
P10/21
P10/22
P10/23
P10/24
P10/25
P10/26
P10/27
P10/28
P10/29
P10/30
P10/31
P10/32
P10/33
P10/34
P10/35
P10/36
P10/37
P10/38
P10/39
P10/40
P10/41
P10/42
P10/43
P10/44
P10/45
P10/46
P10/47
P10/48
P10/49
P10/50
/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
ALL OTHER CONNECTORS ON BUS WIRED IDENTICALLY
P8 CPU BOARD 2 (50 WAY SCOTCHFLEX CONNECTOR)
MIL-STD-1750
MK 1A HARDWARE
INTERCONNECTION CABLES
SH 8 of 8
RAE/FSS

Fig 29

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	3. Agency Reference	4. Report Security Classification/Marking
	RAE TM FS 403	N/A	UNCLASSIFIED

5. DRIC Code for Originator	6. Originator (Corporate Author) Name and Location
7673000W	Royal Aircraft Establishment, Farnborough, Hants, UK
5a. Sponsoring Agency's Code	6a. Sponsoring Agency (Contract Authority) Name and Location
N/A	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	9a. Author 2	9b. Authors 3, 4	10. Date	Pages	Refs.
Shrimpton, S.J.			May 1981	149	4

11. Contract Number	12. Period	13. Project	14. Other Reference Nos.
N/A	N/A		

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.

FS910/1

RAE Form A143