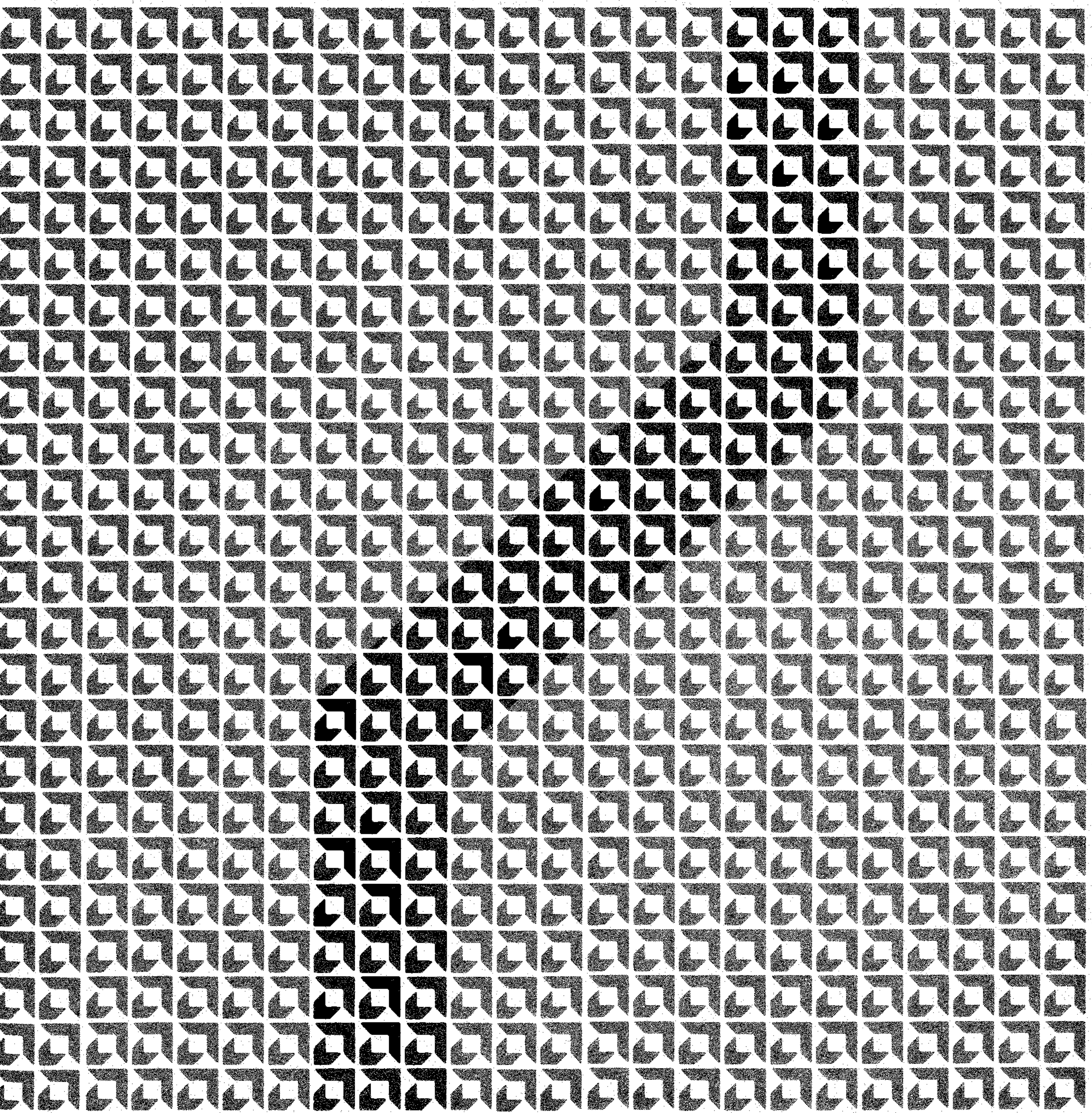
Advanced Micro Devices

Am2900 Learning and Evaluation Kit User's Manual



Advanced Micro Devices, Inc.

Am2900 Evaluation and Learning Kit Instruction Manual

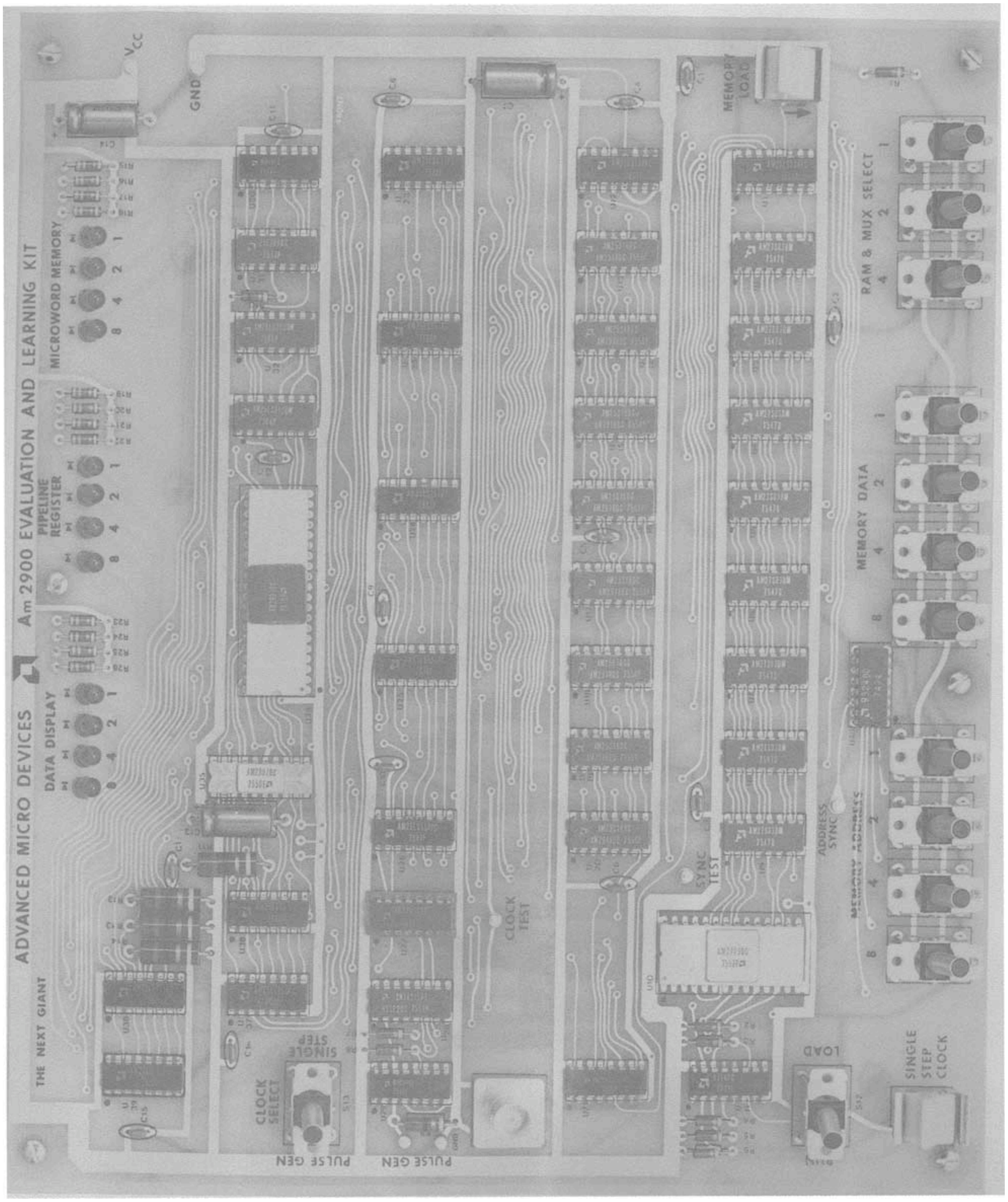

ADVANCED MICRO DEVICES
Am 2900 EVALUATION AND LEARNING KIT
THE NEXT GIANT
DATA DISPLAY
PIPELINE REGISTER
MICROWORD MEMORY
8 4 2 1
VCC
GND
MEMORY LOAD
RAM & MUX SELECT
MEMORY DATA
MEMORY ADDRESS
ADDRESS SYNC
SYNC TEST
CLOCK TEST
CLOCK SELECT
SINGLE STEP
PULSE GEN
LOAD
SINGLE STEP CLOCK

TABLE OF CONTENTS

SECTION I

INTRODUCTION

Advanced Micro Devices has designed an educational tool, the Am2900 Evaluation and Learning Kit, to be used by the design engineer learning microprogramming. The purpose of this kit is twofold. First, it is intended to introduce the design engineer to several of the circuits in the Am2900 Bipolar Microprocessor Family. Second, it is intended to be an instructional tool for the engineer faced with his first microprogramming job. The kit consists of one Am2901 Bipolar Microprocessor, one Am2909 Bipolar Microprogram Sequencer, and several memories, registers, and multiplexers organized in a typical CPU (Central Processing Unit) structure. It should be said at the onset that the purpose of this kit is to introduce the design engineer to the Am2900 Family devices and provide a microprogram learning tool. This kit is NOT a four-bit computer.

The organization of the kit is a reduced example of a typical CPU design. In the microprogram control area, one Am2909 Bipolar Microprogram Sequencer is used to address 16 words of microprogrammed memory. The kit uses eight Am27S03 64-bit RAM's (Random Access Memories) connected as the microprogram memory. This provides a total microprogram memory that is 16 words deep where each word is 32 bits wide. This memory is used as the writeable microprogram memory for holding the microinstructions of the program. The kit contains eight Am2918 Registers used as a 32-bit wide pipeline register at the output of the microprogram memory. The kit also has a 32-word by 8-bit PROM (Programmable Read Only Memory) in the Am2909 control path to generate next microprogram address commands.

The ALU (Arithmetic Logic Unit) in the Am2900 kit utilizes one four-bit Am2901 Bipolar Microprocessor slice. One Am25LS08 is used as the status register which holds the four status flags. The ALU section also uses two Am25LS253 Multiplexers for shifting data into either the RAM shift matrix or the Q shift matrix under microprogrammed control. This allows the execution of various types of microinstructions and the results of these operations can be learned in concept and in practice.

The microprogram memory is loaded with data by using a combination of toggle switches that select the memory to be loaded, apply the address for the memory, and also apply the data to be loaded into the memory. The contents of the microprogram memory can be viewed conveniently, four bits at a time, on four LED (Light Emitting Diode) lamps. Four-bit fields associated with the Am2901, Am2907, Am2909, and Am25LS08 can be viewed on four additional LED lamps. Likewise, four-bit fields of the Am2918 Pipeline Register can be viewed on a separate LED display. This provides the student with the ability to learn the machine operation in a step-by-step program sequence.

All components required in the assembly of the Am2900 Evaluation and Learning Kit are supplied in the kit package. The only item that need be supplied by the user of the kit is a +5V power supply capable of delivering approximately 2 amperes of current. The assembly diagram and assembly instructions in this book show the location of each of the components on the printed circuit board. Each component should be assembled in its position and then soldered in place. The user should assemble the board in steps and check out the assembly as it proceeds. The recommended technique will be to load some of the components onto the printed circuit (PC) board and solder them in place. A +5V power supply will be connected to the PC board and a test procedure followed to ensure proper operation of the devices as installed. This procedure is defined in detail in Section IV during kit assembly.

In order to utilize the Am2900 Family efficiently, a basic understanding of microprogramming is essential. The simplest method of microprogramming is for each microinstruction step to be implemented in a sequential manner. The EXECUTION of the microprogram allows the CPU to operate on a microinstruction in a particular memory location. Then, the microprogram controller increments to the next memory location for the next microinstruction. After each microinstruction fetch, the pipeline register will contain the memory location contents for execution. In order to provide the basic understanding of microprogramming, Section II on this topic is included in this book. Section III describes the basic operation of the Am2900 Evaluation and Learning Kit applying the basics from Section II.

SECTION II

UNDERSTANDING THE BASIC THEORY OF MICROPROGRAMMING

INTRODUCTION

With the advent of the Am2901 four-bit microprocessor slice and the Am2909 bipolar microprogram sequencer, the design engineer can upgrade the performance of existing systems or implement new systems taking advantage of the latest state-of-the-art technology in Low-Power Schottky integrated circuits. These devices, however, utilize a new concept in machine design not familiar to many design engineers. This technique is called microprogramming.

Basically, a microprogrammed machine is one in which a coherent sequence of microinstructions is used to execute various commands required by the machine. If the machine is a computer, each sequence of microinstructions can be made to execute a macroinstruction. All of the little elemental tasks performed by the machine in executing the macroinstruction are called microinstructions. The storage area for these microinstructions or microprogram signals is usually called the microprogram memory.

A microinstruction usually has two primary parts. These are: (1) the definition and control of all micro-operations to be carried out and (2) the definition and control of the address of the next microinstruction to be executed.

The definition of the various micro-operations to be carried out usually includes such things as ALU source operand selection, ALU function, ALU destination, carry control, shift control, interrupt control, data-in and data-out control, and so forth. The definition of the next microinstruction function usually includes identifying the source selection of the next microinstruction address and, in some cases, supplying the actual value of that microinstruction address.

Microprogrammed machines are usually distinguished from non-microprogrammed machines in the following manner. Older, non-microprogrammed machines implemented the control function by using combinations of gates and flip-flops connected in a somewhat random fashion in order to generate the required timing and control signals for the machine. Microprogrammed machines, on the other hand, are normally considered highly ordered and more organized with regard to the control function field. In its simplest definition, a microprogram control unit consists of the microprogram memory and the structure required to determine the address of the next microinstruction.

UNDERSTANDING THE MICROPROGRAM MEMORY

The microprogram memory is simply a N word by M bit memory used to hold the various microinstructions. Figure 1 depicts the word number definition of an N word microprogrammed memory. For an N word memory, the address locations are usually defined as location 0 through location N-1. For example, a 256-word microprogram memory will have address locations 0 through 255.

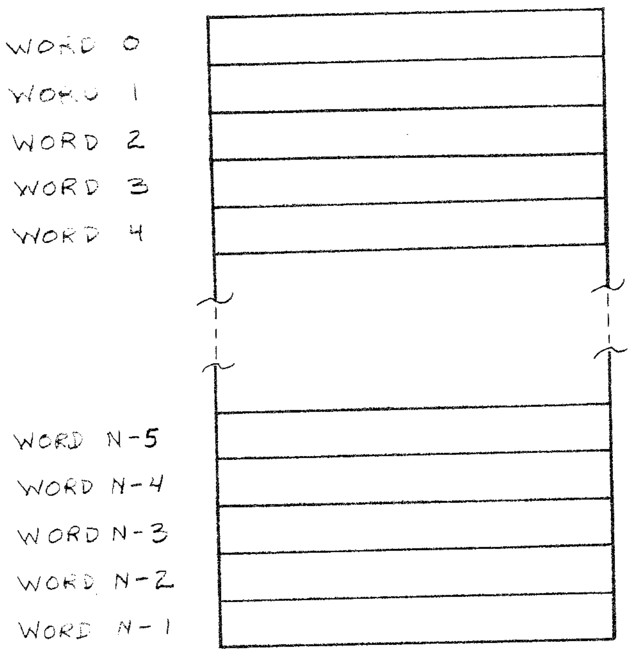


FIGURE 1: Organization of an N-word Microprogram Memory

Each word of the microprogram memory consists of M bits. These M bits are usually broken into various field definitions. A typical example of field definition is shown in Figure 2 where a 32-bit word made up of nine fields is depicted. It should be noted that the fields can consist of various numbers of bits. For example, field number 1 is 5 bits wide, field number 2 is 8 bits wide, field number 6 is 1 bit wide, and so forth. It is the definition of the various fields of a microprogram word that is usually referred to as FORMATTING. Thus, Figure 2 shows the format of a 32-bit microinstruction.

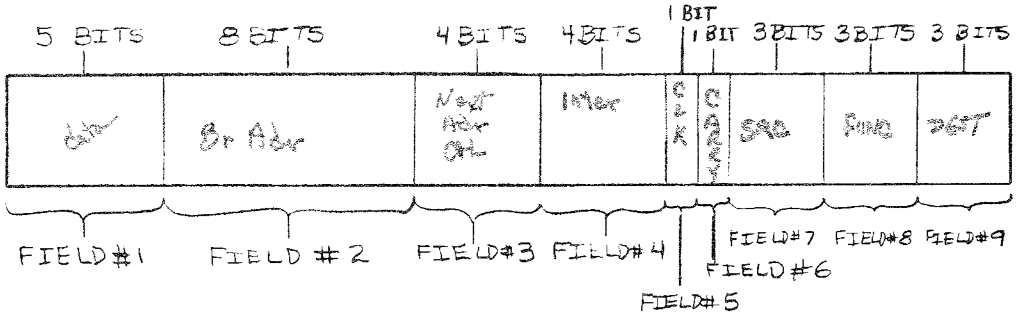


FIGURE 2: Definition of the Fields within one 32-bit MICROPROGRAM Memory Word

Examples of the uses of the field as defined in Figure 2 are as follows:

Field 1	General purpose
Field 2	Branch address
Field 3	Next address control
Field 4	Interrupt control
Field 5	Fast clock/slow clock select
Field 6	Carry control
Field 7	ALU source operand control
Field 8	ALU function control
Field 9	ALU destination control

The above field definition is an example of how microinstruction fields are defined in a typical machine.

SEQUENCING THROUGH MICROINSTRUCTIONS

Once the microprogram format has been defined, it is necessary to execute sequences of these microinstructions if the machine is to perform any real function. In its simplest form, all that is required to sequence through a series of microinstructions is a microprogram address counter. Such a simplified microprogram memory address control is shown in Figure 3. The microprogram address counter simply increments by one on each clock cycle to select the address of the next microinstruction. Figure 4 shows an example of what might be occurring in this mode. For example, if the microprogram address counter contains address 23, the next clock cycle will increment the counter and it will select address 24. The counter will continue to increment on each clock cycle thereby selecting address 25, address 26, address 27, and so forth. If this were the only control available, the machine would not be very flexible but it would be able to execute a fixed pattern of microinstructions.

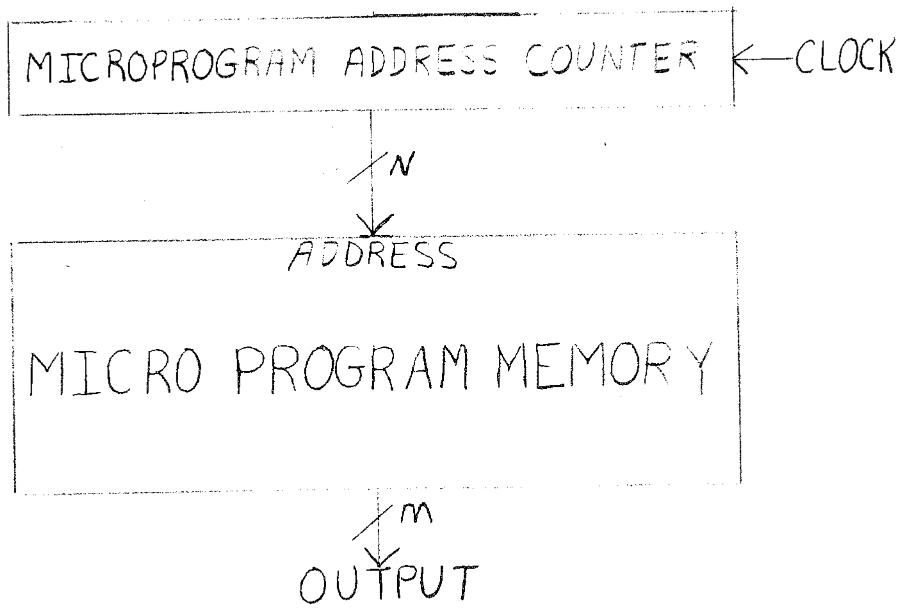


FIGURE 3: Microprogram Memory Address Control Using a Counter

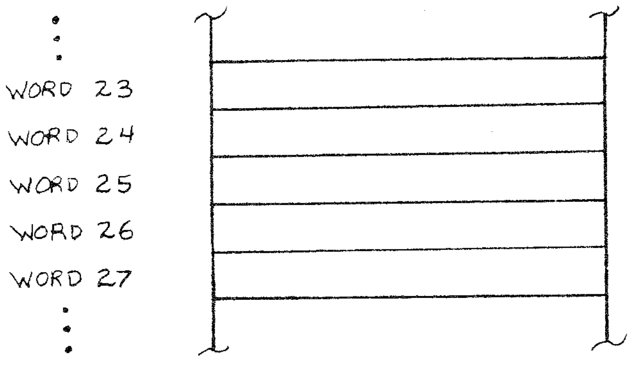


FIGURE 4: Executing Sequential Microprogram Memory Words

The technique of continuing from one microinstruction to the next sequential microinstruction is usually referred to as CONTINUE or EXECUTE. Thus, in microprogram control definition, we will use the CONTINUE or EXECUTE statement to mean simply incrementing to the next microinstruction.

MICROPROGRAM BRANCHING OR JUMPING

If the microprogram control unit is to have the ability to select other than the next microinstruction, the control unit must be able to load a BRANCH address. Figure 5 shows a control unit architecture whereby a BRANCH address can be parallel loaded into the microprogram address counter. The load control is a single bit field within the microprogram word format. Let us call this one-bit field the microprogram address counter load enable bit. When this bit is at logic 0, a load will be inhibited and when this bit is logic 1, a load will be enabled. If the load is enabled, the branch address contained within the microprogram memory will be parallel loaded into the microprogram address counter. This results in the ability to perform an N-way branch. For example, if the branch address field is eight bits wide, a branch to any address in the memory space from word 0 through word 255 can be performed. Note that there may be many other fields in the microprogram.

Let us examine the programming that would be required by the microprogram control unit described in Figure 5. Referring to the microprogram memory map of Figure 6, assume the current address i n the microprogram address counter is 51. Since the load control is logic 0, the counter will increment on the next clock cycle and will contain address 52. In fact, the counter will continue to increment through words 53, 54, 55, 56 and 57. When the microprogram address counter contains address 57 as shown in Figure 6, the load control bit changes to a logic 1. When this happens, the microprogram address counter will parallel load the branch address supplied to its parallel data inputs. An example of such a counter is the Am25LS161 binary counter. The branch address shown in the contents of word 57 in Figure 6 is address 90. On the next clock cycle, 90 will be loaded into the microprogram address counter. Thus, the next microinstruction executed will be defined by the contents of the microinstruction word at address 90. Figure 6 now defines the load control bit at instruction 90 to be logic 0. Thus, on the next clock cycle, the counter will increment to address 91. When address 92 is reached, another BRANCH is defined. The branch address at word 92 is address 13.

The simple branching control feature described in Figure 5 and Figure 6 allows a microprogram memory controller to execute sequential microinstructions or perform a BRANCH to any address either before or after the address currently contained in the microprogram address counter.

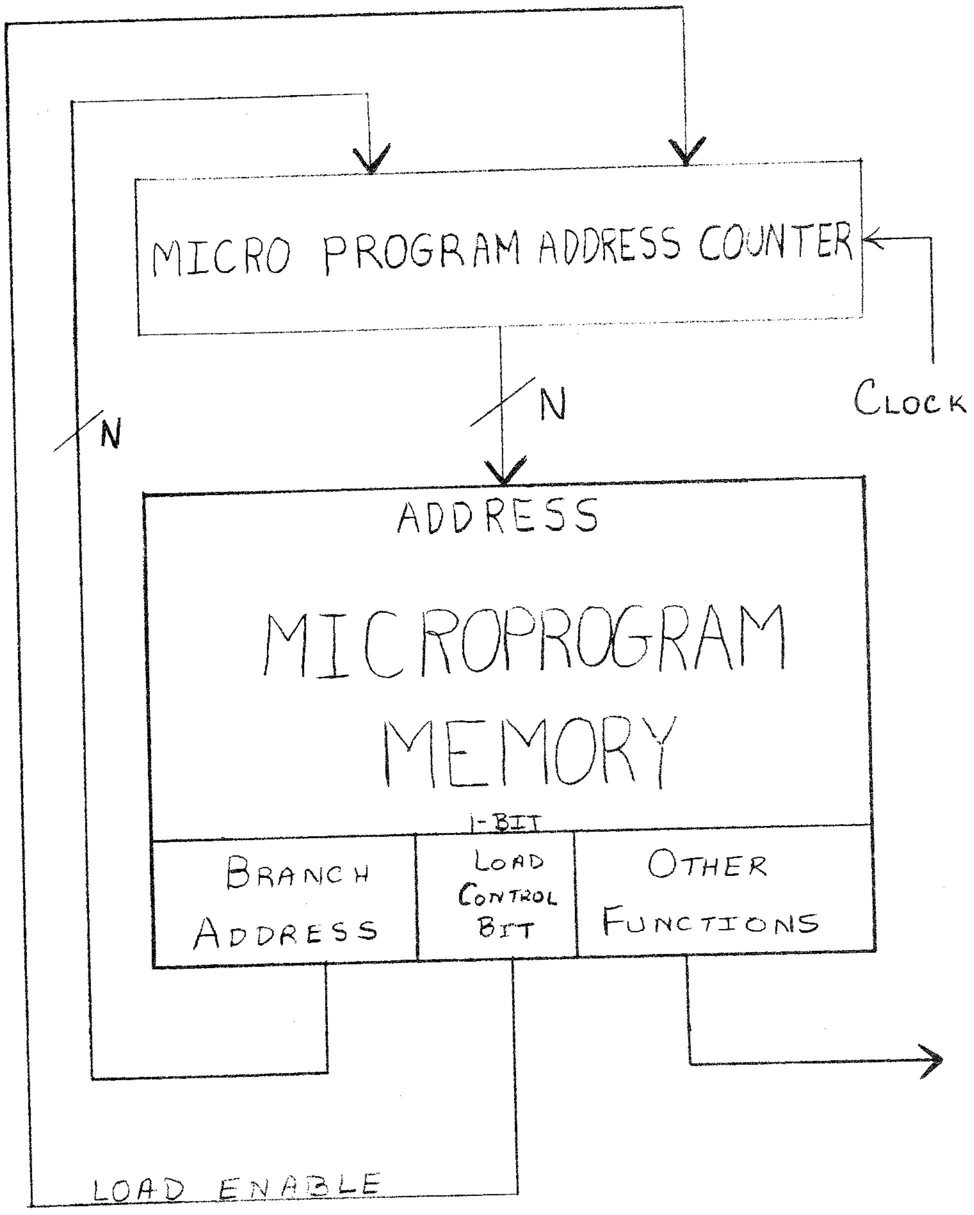


FIGURE 5: Branching or Jumping in a Microprogram Memory Space Requires a Branch Address and a Load Control Signal

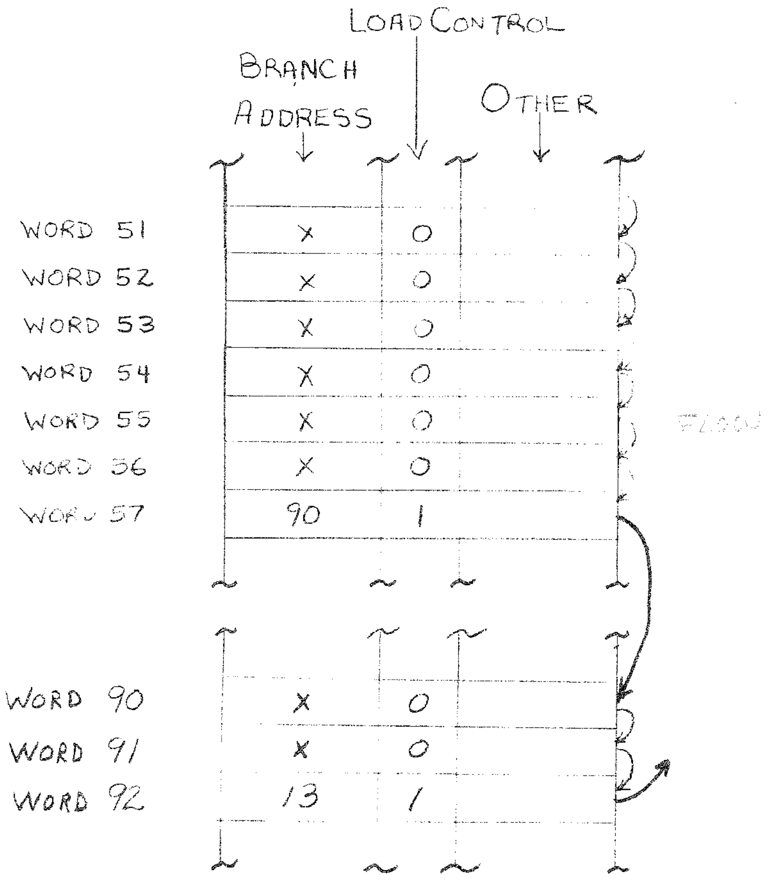


FIGURE 6: Branching or Jumping in a Microprogram Control Space

CONDITIONAL BRANCHING

While the BRANCH or JUMP instruction has added some flexibility to the sequencing of microprogram instructions, the controller still lacks any decision-making capability. This decision-making capability is provided by the CONDITIONAL BRANCH instruction. Figure 7 shows a functional block diagram of a microprogram memory/address controller providing the capability to branch on two different conditions. In this example, the load select control is a two-bit field used to control a four-input multiplexer. When the two-bit field is equivalent to binary zero, the multiplexer selects the zero input which forces the load control inactive. Thus, the CONTINUE microprogram control instruction is executed. When the two-bit load select field contains binary one, the D_1 input of the multiplexer is selected. Now, the load control is a function of the Condition 1 input. If Condition 1 is logic 0, the microprogram address counter increments and if Condition 1 is logic 1, the branch address will be parallel loaded in the next clock cycle. This operation is defined as a CONDITIONAL BRANCH. If the load select input contains binary 2, the D_2 input is selected and the same function is performed with respect to the Condition 2 input. If the load select field contains binary 3, the D_3 input of the multiplexer is selected. Since the D_3 input is tied to logic HIGH, this forces the microprogram address counter to the load mode independent of anything else. Thus, the branch address is loaded into the microprogram address counter on the next clock cycle and an UNCONDITIONAL JUMP is executed. This load select control function definition is shown in Table I.

TABLE I

Load Select Control Function

S_1	S_0	Function
0	0	Continue
0	1	Jump Condition 1 True
1	0	Jump Condition 2 True
1	1	Jump Unconditional

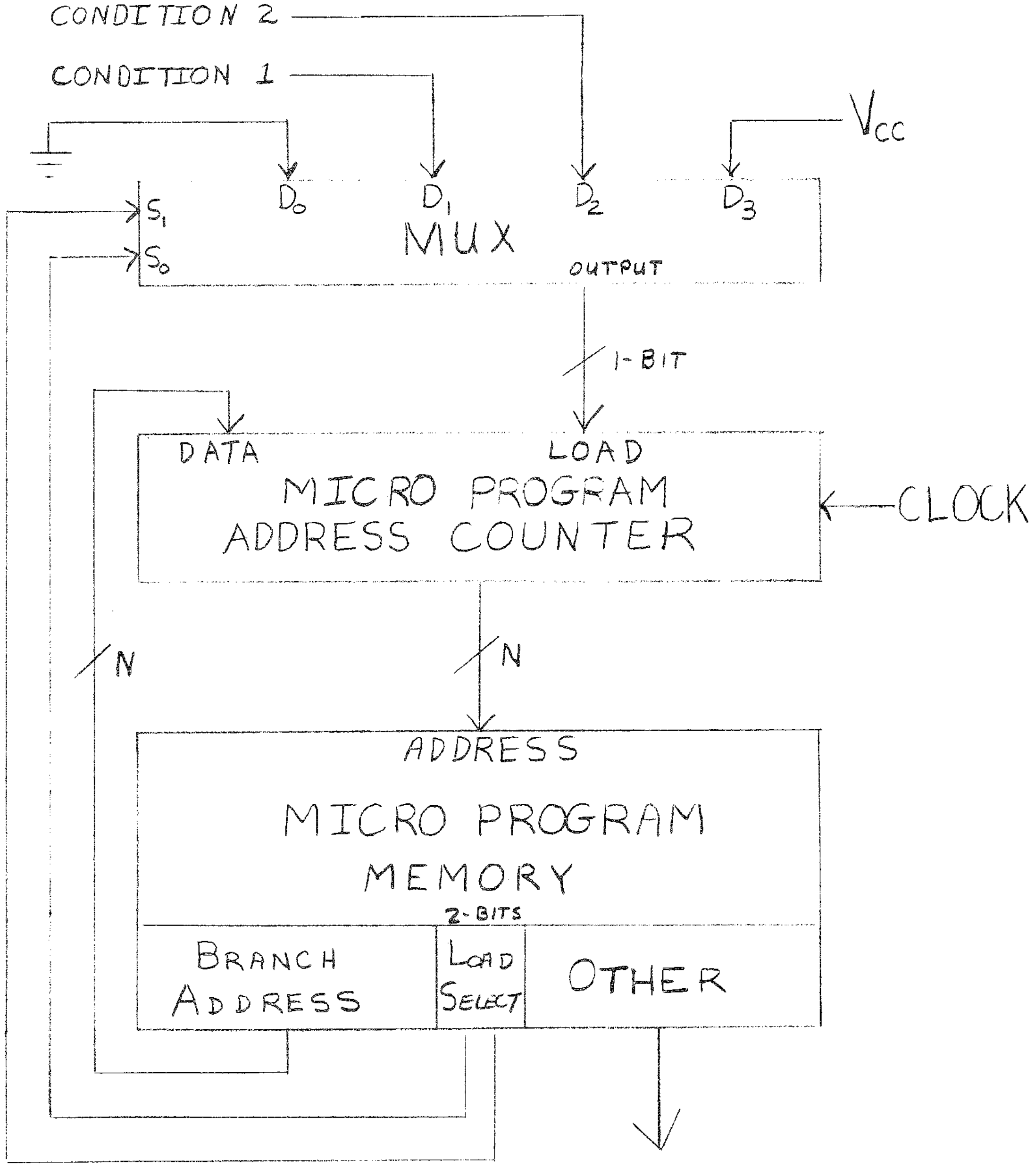


FIGURE 7: A Two-bit Control Field can be used to Select CONTINUE, BRANCH, or CONDITIONAL BRANCH

A microprogram memory map showing the use of conditional branching in microprogram control is depicted in Figure 8. In this map, we assume the microprogram address counter is currently pointing at word 30. The load select field causes the address counter to increment to word 31 on the next clock cycle. At word 31, a CONDITIONAL BRANCH is encountered. If the Condition 1 input is logic 0, the next word selected will be word 32. If the Condition 1 input is logic 1, the next word selected will be word 38. The example of Figure 8 shows other CONDITIONAL BRANCHES at word 32, 35, and 36. It also shows an UNCONDITIONAL BRANCH at word 39. It should be noted that words 35 and 36 result in CONDITIONAL BRANCHES to the same address, address 95. The essence of this function is that if either Condition 1 or Condition 2 is true as tested on sequential microinstructions, the program branches to the sequence beginning at address 95.

An example of the power of CONDITIONAL BRANCHING is shown in Figure 9A. Here, ignoring the start-up problem, let us assume the microprogram address counter contains the word 0. The instruction being executed is a conditional test on Condition 1 input. If Condition 1 is true, branch to word 4 and if Condition 1 is false, continue. Word 1 performs a test on Condition 2. This test results in a branch to word 8 if true, or a continue if false. Word 2 performs a condition test on Condition 1. If true, a branch to word 12 occurs, if false a continue to word 3 occurs. Word 3 is an unconditional branch to the starting address at word 0. In this example, if a jump occurs to either word 4, 8, or 12, the net result is that the next four instructions are executed and then an unconditional branch to word 0 occurs. The flow table for this 16-word microprogram is shown in Figure 9B.

The microprogram control flow described in Figures 9A and 9B represent a simple state machine design controller that looks for either Condition 1 to occur, Condition 2 to occur, or not Condition 2 followed by Condition 1 occurring before a reset to word 0. In the event any of the conditions do occur, a small series of tasks (four microinstructions) are to be executed and then the machine returned to word 0. If none of the test conditions occur, the machine is reset at word 3 and returned to the test at word 0. At power-up, an asynchronous reset could be applied to the microprogram address counter to initialize the machine. Although Figure 9A and Figure 9B describe an extremely simple microprogram control unit, it is representative of the microprogram control power contained in only three basic microinstructions. These microinstructions are the CONTINUE instruction, the UNCONDITIONAL JUMP instruction, and the CONDITIONAL JUMP instruction.

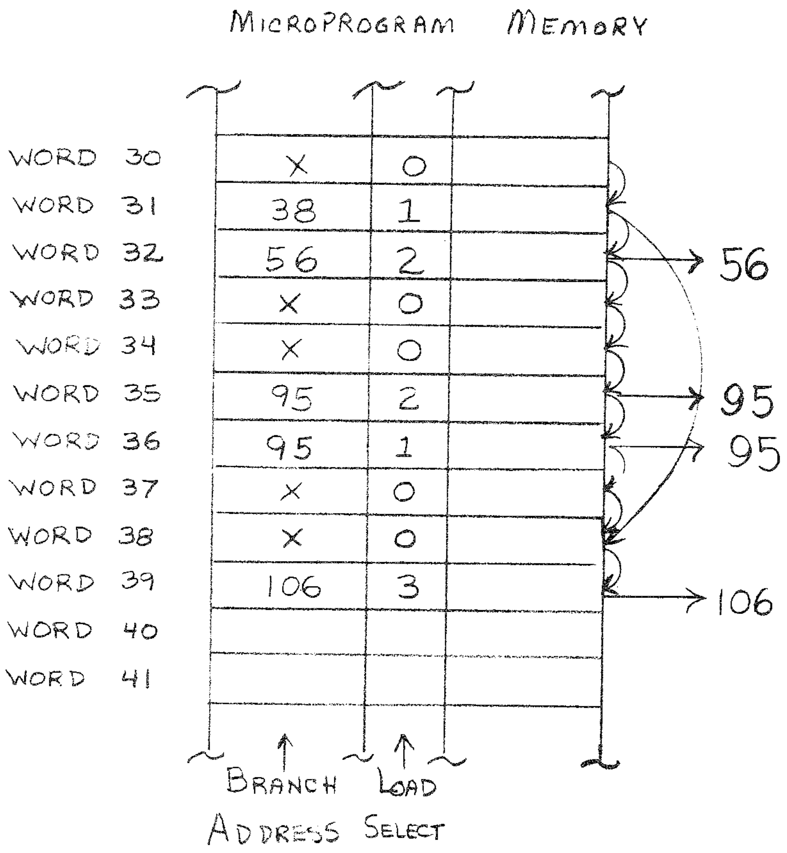


FIGURE 8: CONDITIONAL BRANCHING Using a Two-bit Select Field as shown in Figure 7

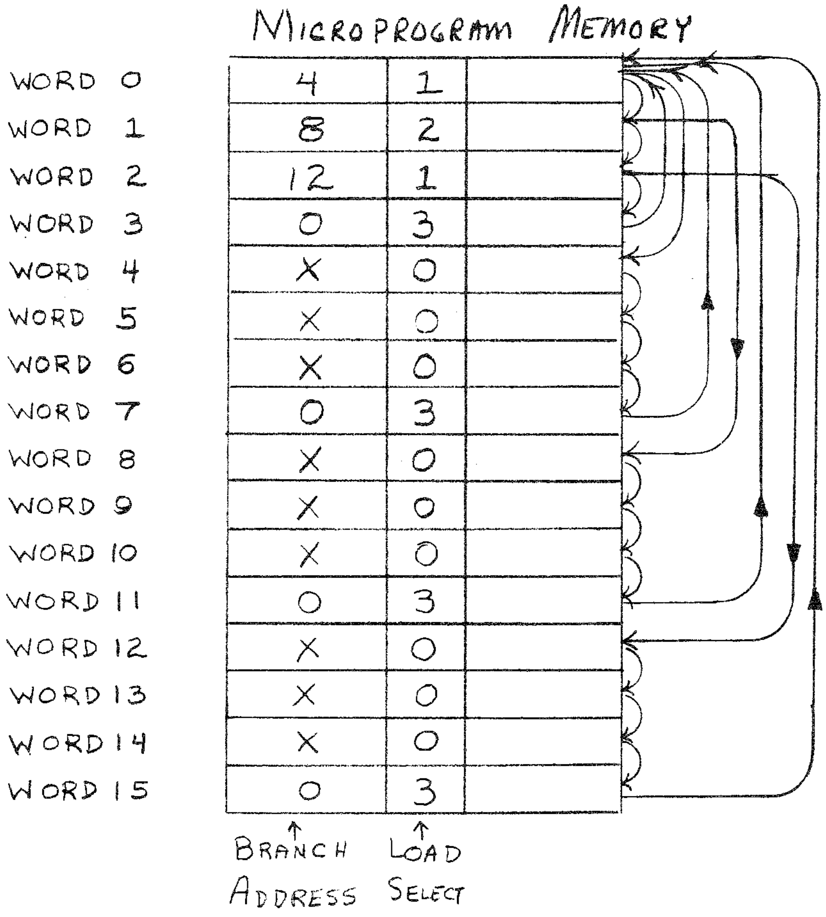


FIGURE 9A: CONDITIONAL BRANCHING Example

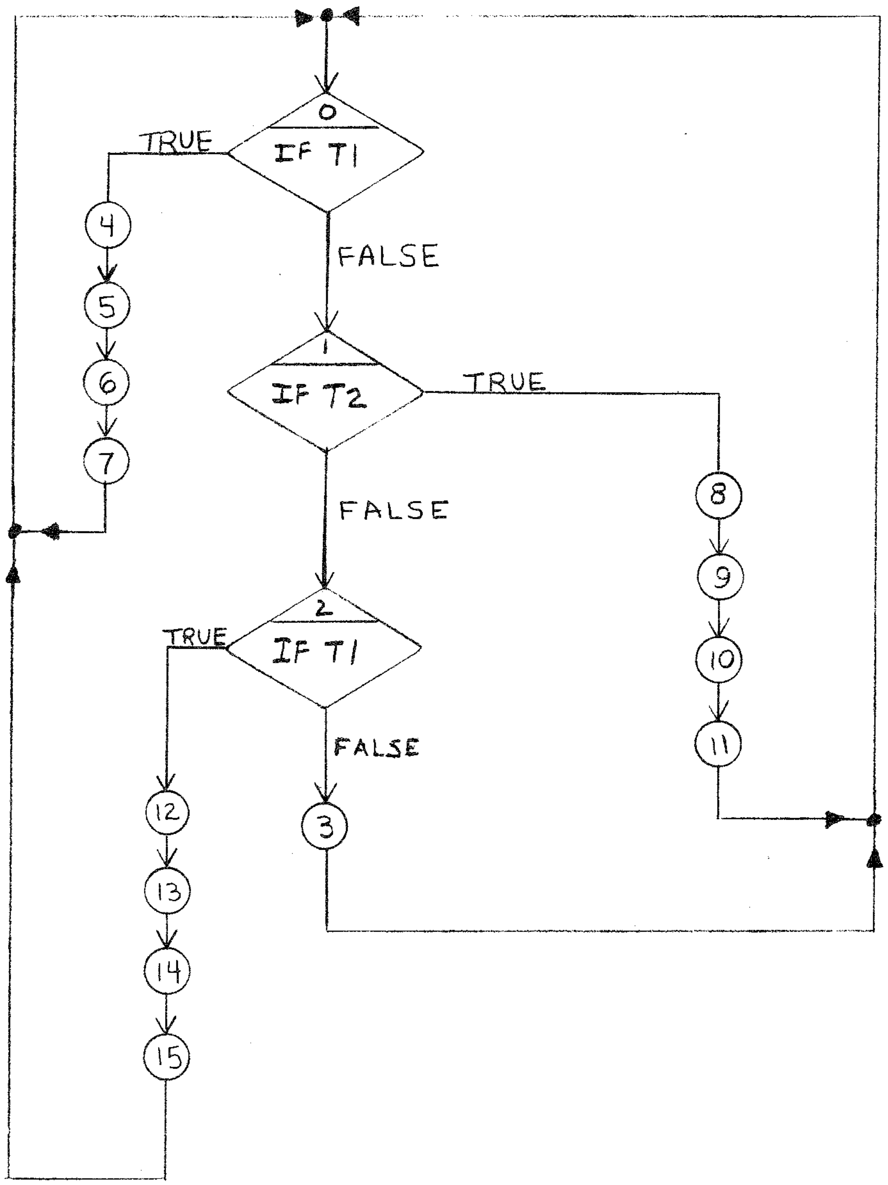


FIGURE 9B: Flow Table for CONDITIONAL BRANCH Example

OVERLAPPING THE MICROPROGRAM INSTRUCTION FETCH

Now that a few basic microprogram address control instructions have been defined, let us examine the instructions used in a microprogram control unit featuring the overlap fetching of the next microinstruction. This technique is also known as "pipelining." The block diagram for such a microprogram control unit is shown in Figure 10. The key difference when compared with previous microprogrammed architectures is the existence of the "pipeline register" at the output of the microprogram memory. By definition, the pipeline register (or microword register) contains the microinstruction currently being executed by the machine. Simultaneously, while this microinstruction is being executed, the address of the next microinstruction is applied to the microprogram memory and the contents of that memory word are being fetched and set-up at the inputs to the pipeline register. This technique of pipelining can be used to improve the performance of the microprogram control unit. This results because the contents of the microprogram memory word required for the next cycle are being fetched on an overlapping basis with the actual execution of the current microprogram word. It should be realized that when the pipeline approach is used, the microprogram fetch, the current microinstruction being executed and the results of the previous microinstruction are available with respect to each other simultaneously. Said another way, the design engineer must be aware of the fact that some registers contain the results of the previous microinstruction executed, some registers contain the current microinstruction being executed, and some registers contain data for the next microinstruction to be executed.

Let us now compare the block diagram of Figure 10 with that shown in Figure 7. The major difference, of course, is the addition of the pipeline register at the output of the microprogram control memory. Also, notice the addition of the address multiplexer at the source of the microprogram memory address. This address multiplexer is used to select the microprogram counter register or the pipeline register as the source of the next address for the microprogram memory. The condition code multiplexer is used to control the address multiplexer in this address selection. By placing an incrementer at the output of the address multiplexer, it is possible to always generate the current microprogramming address "plus one" at the input of the microprogram counter register. In Figure 7, the microprogram address counter was described as a device such as the Am25LS161 counter. In the implementation as shown in Figure 10, the Am25LS161 counter is not appropriate. Instead, an incrementer and register are used to give the equivalent effect of a counter.

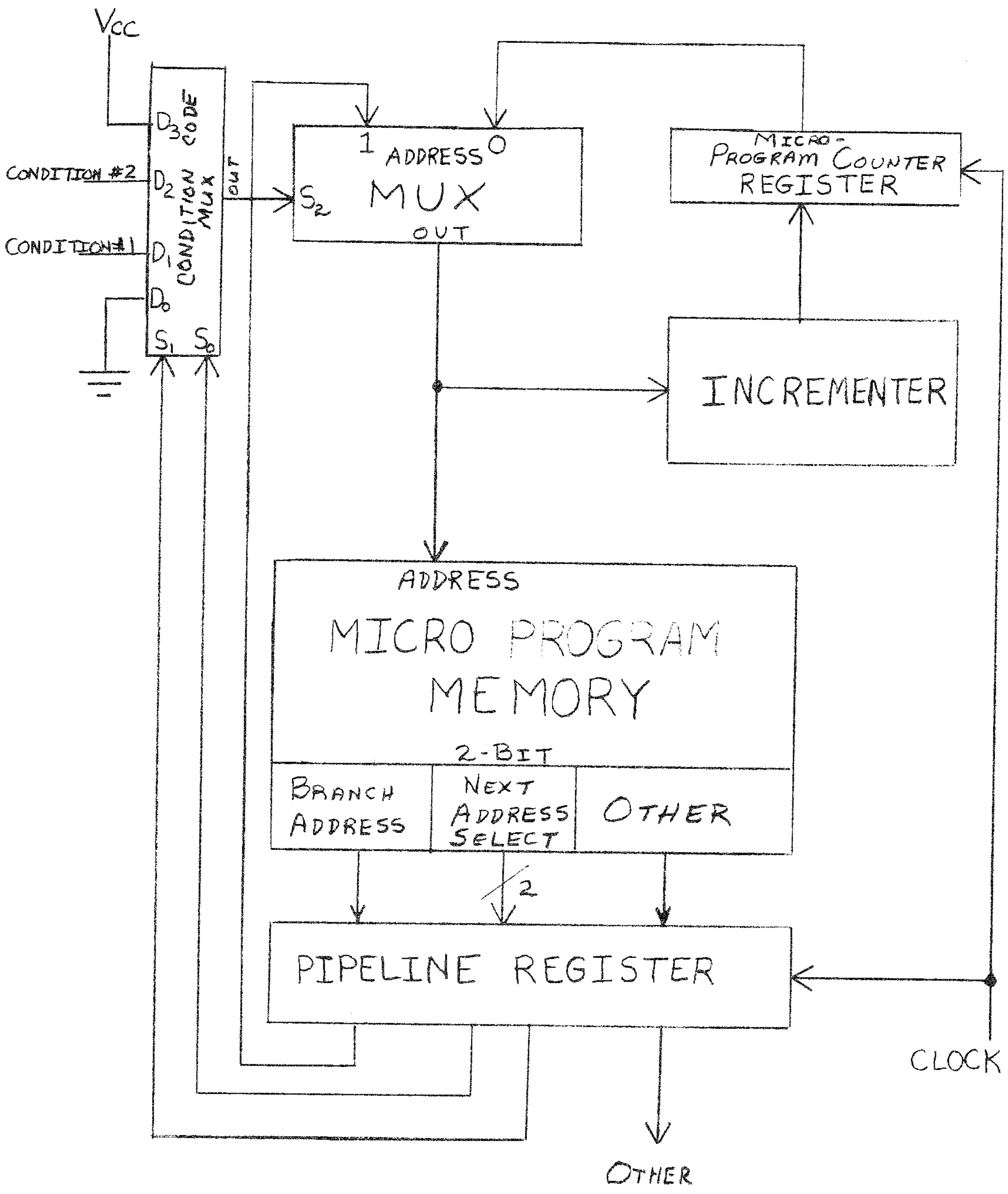


FIGURE 10: OVERLAPPING (or PIPELINING) the Fetch of the Next Microinstruction

The key difference between using a true binary counter and the incrementer register described here is as follows. When the branch address from the pipeline register is selected by the multiplexer, the incrementer will combinatorially prepare that address plus one for entry into the microprogram counter register. This entry will occur on the LOW-to-HIGH transition of the clock. Thus, the microprogram counter register can always be made to contain address plus one, independent of the selection of the next microinstruction address. When the address mutliplexer is switched so that the microprogram counter register is selected as the source of the microprogram memory address, the incrementer will again set-up address plus one for entry into the microprogram counter register. Thus, when the address multiplexer selects the microprogram counter register, the address multiplexer, incrementer and microprogram counter register appear to operate as a normal binary counter.

The condition code multiplexer S_0S_1 operates in exactly the same fashion as described for the condition code multiplexer of Figure 7. That is, binary zero in the pipeline register (the current microinstruction being executed) forces an unconditional selection of the microprogram register via D_0. Binary one or binary two in the next address select control bits of the pipeline register cause a conditional selection at the address mutliplexer via D_1 or D_2. Thus, a CONDITIONAL BRANCH can be executed. Binary three in the next address select portion of the pipeline register causes an UNCONDITIONAL BRANCH instruction to be executed via D_3.

When the overall machine timing is studied, it will be noticed that the key difference between overlap fetching and non-overlap fetching involves the propagation delay of the microprogram memory. In the non-pipelined architecture, the microprogram memory propagation delay must be added to the propagation delay of all the other elements of the machine. In the overlap fetch architecture, the propagation delay associated with the next microprogram memory address fetch is a separate loop independent of the other portion of the machine.

SUBROUTINING IN MICROPROGRAM CONTROL

Thus far, we have examined the CONTINUE instruction as well as the CONDITIONAL and UNCONDITIONAL BRANCH instructions. Just as in the programming of minicomputers and microcomputers, the advantages of SUBROUTINING can be realized in microprogramming. The idea here, of course, is that the same block of microcode can be shared by several instruction sequences of microcode. This results in the overall reduction in the total number of microprogram memory words required by the design. If we are to jump to a subroutine, what is required is the ability to store an address to which the subroutine should return when it has completed its execution. Examining the block diagram of Figure 11, we see the

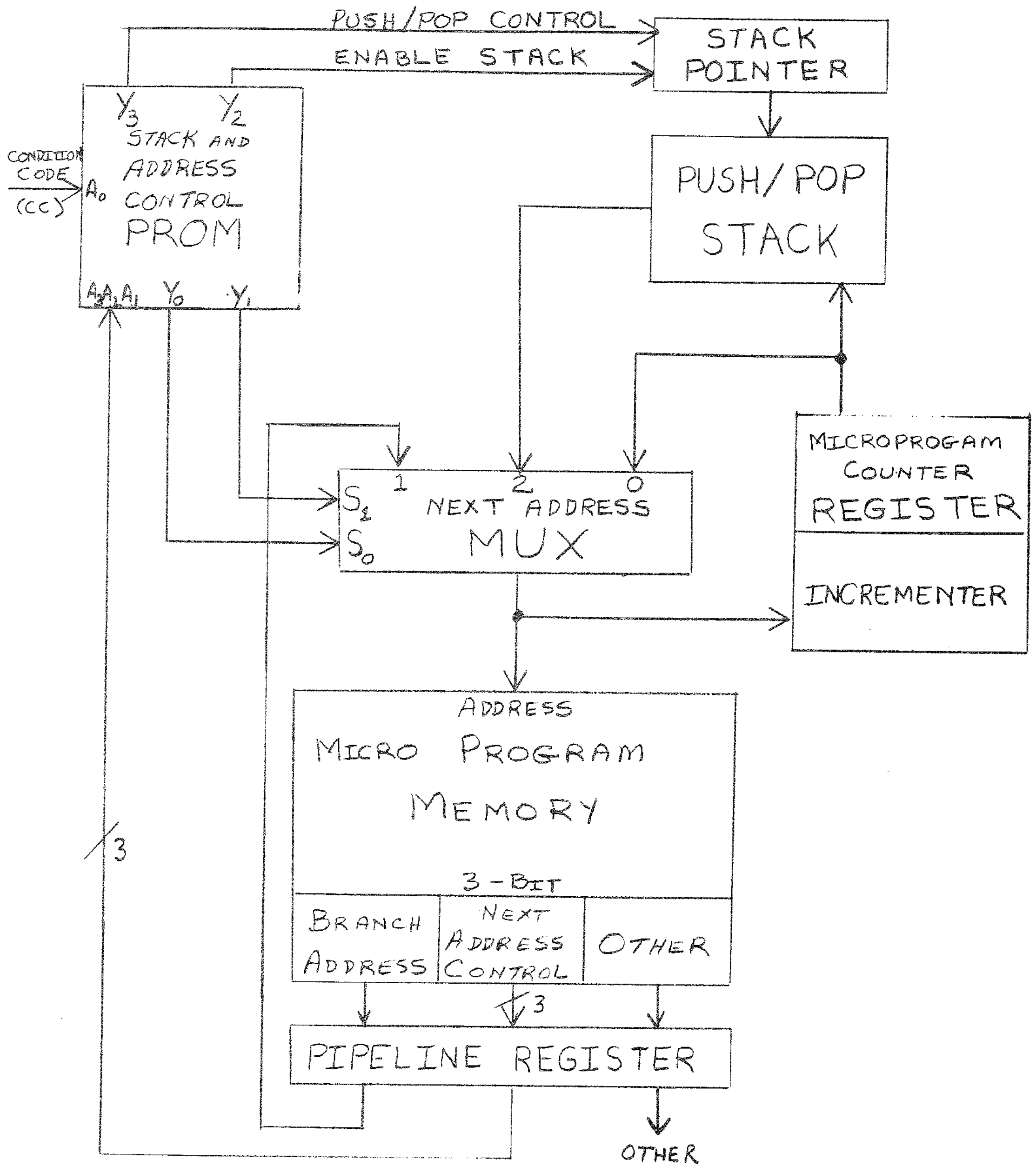


FIGURE 11: Block Diagram for SUBROUTINE Control Using a PUSH/POP Stack

addition of a push/pop stack and its associated stack pointer. The control signals required by the stack are an enable stack signal which will be used to tell the file whenever we wish to perform a push or a pop, and a push/pop control used to control the direction of the stack pointer (push or pop).

In this architecture, the stack pointer always points to the addres of the last microinstruction written into the file. This allows the "next address multiplexer" to read the file at any time via port 2 on the next address MUX. When this selection is performed, the last word written on the stack will be the word applied to the microprogram memory. The condition code multiplexer of the previous example has also been replaced by a stack and address control programmable read only memory (PROM). The next address control field of the microprogram word has been expanded to a three-bit field. This three-bit field in conjunction with one condition code bit form the four-bit address control for the stack and address control PROM. The PROM has four outputs used to control the next address multiplexer and the stack pointer. The net result is that in this example, eight instructions have been defined where three of the instructions are conditional. Table II shows these next address select control functions. They include continue, branch, branch on condition, push, jump-to-subroutine, jump-to-subroutine on condition, return-from-subroutine, and test end of loop. Note that this architecture does not provide the ability to select any particular starting address. Also, either a master reset or a known address must be provided to the system to somehow handle power-up turn-on. A system using the Am2909 can provide this feature. Table III shows a detailed definition of the instruction set defined in Table II.

TYPICAL COMPUTER CONTROL UNIT ARCHITECTURE USING THE AM2909

The microprogram memory control described in Figure 11 is easily implemented using the Am2909 as shown in the block diagram of Figure 12. Note the addition of a fourth input (D) for selecting the starting address of a sequence of microinstructions. The Am2909 thereby provides four different inputs from which the next address can be selected. These are the direct input (D), the register input (R), the program counter (PC), and the file (F). A detailed logic diagram of the Am2909 is shown for clarity in Figure 13.

The architecture of Figure 12 shows a macroinstruction register capable of being loaded with a macroinstruction word from the data bus. The op code portion of the instruction is decoded in a mapping PROM to arrive at a starting address for the microinstruction sequence required to execute the macroinstruction. When the microprogram memory address is to be the first microinstruction of the macroinstruction sequence, the microsequencer control PROM selects the multiplexer D input.

TABLE II

Next Address Select Control Functions

$A_3A_2A_1$	Function
000	Continue
001	Branch
010	Branch on condition
011	Push
100	Jump to subroutine
101	Jump to subroutine on condition
110	Return from subroutine
111	Test end of loop

TABLE III

PROM Control for Table II Function

Function	$A_3A_2A_1$	A_0	Next Address MUX S_1S_0	Enable Stack	Push/ Pop
Continue	000	0 1	0	No	X
Branch	001	0 1	1	No	X
Conditional Branch	010	0 1	0 1	No	X
Push	011	0 1	1	Yes	Push
Jump to Subroutine	100	0 1	1	Yes	Push
Conditional Jump to Subroutine	101	0 1	0 1	No Yes	X Push
Return from Subroutine	110	0 1	2	Yes	Pop
Test end of loop	111	0 1	2 0	No Yes	X Pop

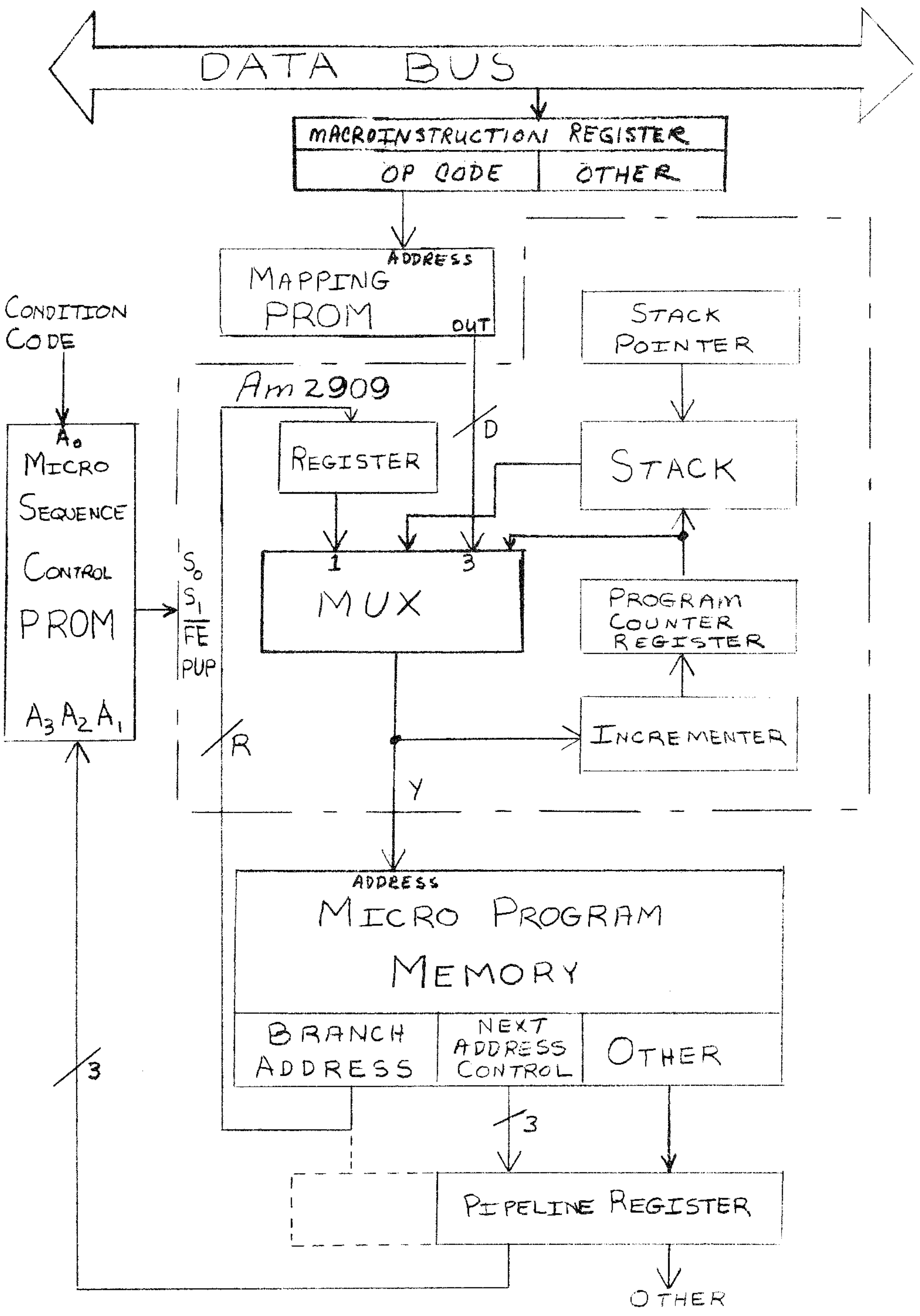
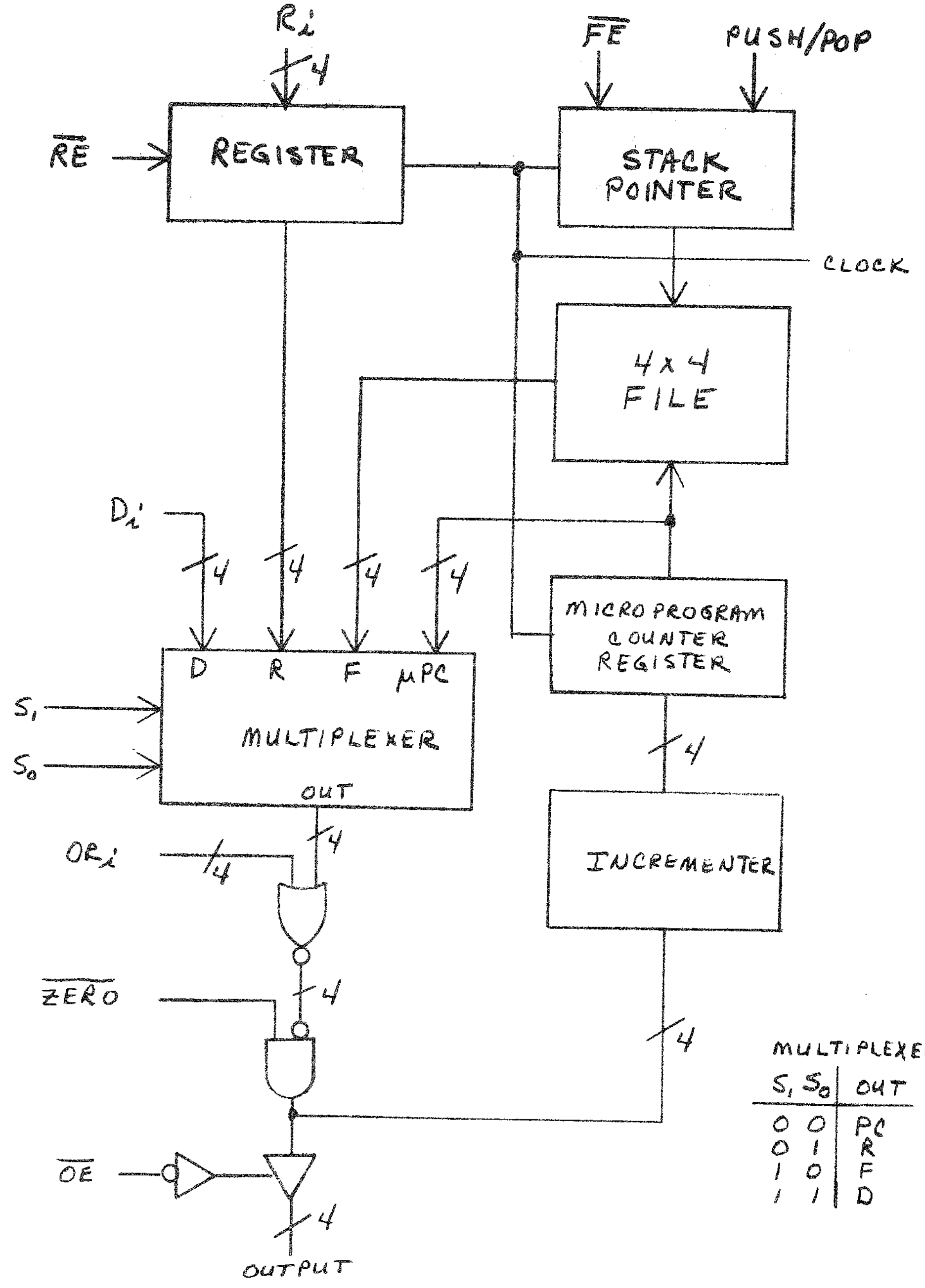


FIGURE 12: A Typical Computer Control Unit Architecture Using Am2909



MULTIPLEXER

S_1 S_0	OUT
0 0	PC
0 1	R
1 0	F
1 1	D

FIGURE 13: Detailed Logic Diagram of the Am2909

The register internal to the Am2909 should be thought of as the pipeline register. When the current microinstruction being executed is selecting the next microinstruction address as a branch function, the branch address will be contained in the register internal to the Am2909. When sequencing through continuous instruction in microprogram memory, the program counter in the Am2909 is used. The 4 x 4 stack in the Am2909 is used for looping and subroutining in microprogram operations.

If a three-bit field for the next address control is used for the architecture shown in Figure 12, the recommended eight functions are shown in Table IV. The eight functions providing good control for selection of the address of the next microinstruction include continue, branch, branch on condition, push, jump-to-subroutine, starting address, return-from-subroutine, and test end of loop. The detailed control required to implement these eight functions is shown in Table V. The actual PROM coding for this example is shown in Table VI. The PROM selected for this implementation is an Am29751 32-word by 8-bit Programmable Read Only Memory.

If more flexibility is required, the architecture shown in Figure 12 can be expanded such that a four-bit next address field format is used to drive the microsequencer control PROM. This provides the ability to have 16 different control functions for selecting the address of the next microinstruction. Table VII shows an example of these 16 functions providing a comprehensive set of instructions for microprogram address control. Table VIII shows the actual PROM coding for this instruction set and Figure 14 shows a detailed functional connection diagram for this microsequencer instruction set.

Am2911 MICROPROGRAM SEQUENCER

The Am2911 Microprogram Sequencer is similar in design to the Am2909 device. The difference between the Am2909 and the Am2911 involves the D inputs, R inputs, and OR inputs. On the Am2911, the OR inputs have been eliminated to save four pins. Also on the Am2911, the R inputs and D inputs have been connected together to save an additional four pins. This allows the 28-pin Am2909 to be packaged in a 20-pin package.

When using the Am2911, the normal technique in a computer control unit is to multiplex the output of the mapping PROM and the output of the pipeline register for the branch address. This is easily accomplished using three-state control at each of these points. The enable signal or the mapping PROM and the branch address field three-state control can be supplied from the Am29751 Control PROM associated with the microprogram sequencer.

TABLE IV

Next Address Select Control Functions

$A_3A_2A_1$	Function
000	Continue
001	Branch
010	Branch on condition
011	Push
100	Jump to subroutine
101	Starting address
110	Return from subroutine
111	Test end of loop

TABLE V

PROM Control for Table IV Function

Function	$A_3A_2A_1$	CC A_0	Next Address MUX S_1S_0	Enable Stack	Push/ Pop
Continue	000	0 1	0	No	X
Branch	001	0 1	1	No	X
Conditional Branch	010	0 1	0 1	No No	X X
Push	011	0 1	0	Yes	Push
Jump to Subroutine	100	0 1	1	Yes	Push
Starting Address	101	0 1	3	No	X
Return	110	0 1	2	Yes	Pop
Test end of loop	111	0 1	2 0	No Yes	X Pop

TABLE VI

Am29751 PROM Coding for Table V Example

Address $A_4A_3A_2A_1A_0$	PROM Output 0_3	0_2	0_1	0_0
0 0 0 0 0	X	1	0	0
0 0 0 0 1	X	1	0	0
0 0 0 1 0	X	1	0	1
0 0 0 1 1	X	1	0	1
0 0 1 0 0	X	1	0	0
0 0 1 0 1	X	1	0	1
0 0 1 1 0	1	0	0	0
0 0 1 1 1	1	0	0	0
0 1 0 0 0	1	0	0	1
0 1 0 0 1	1	0	0	1
0 1 0 1 0	X	1	1	1
0 1 0 1 1	X	1	1	1
0 1 1 0 0	0	0	1	0
0 1 1 0 1	0	0	1	0
0 1 1 1 0	X	0	1	0
0 1 1 1 1	0	1	0	0

0 = LOW 1 = HIGH

X = Don't care

Notes:

1. Am2909 Connections:

PROM 0_0 to S_0 (Pin 16)

PROM 0_1 to S_1 (Pin 17)

PROM 0_2 to $\overline{FE}$ (Pin 25)

PROM 0_3 to PUP (Pin 26)

2. Ground A_4 (Pin 14) of Am29751

TABLE VII

PROM Control for 16 Next Address Functions

Function	$A_3A_2A_1A_0$	CC	Next Address	Stack Enable	Push Pop	Zero	OR
Start Address	0000	0 1	D	No	X	H	L
Continue	0001	0 1	PC	No	X	H	L
Branch	0010	0 1	R	No	X	H	L
Jump zero	0011	0 1	X	No	X	L	X
Push	0100	0 1	PC	Yes	Push	H	L
Pop	0101	0 1	PC	Yes	Pop	H	L
JSB	0110	0 1	R	Yes	Push	H	L
RTS	0111	0 1	F	Yes	Pop	H	L
Test end of loop	1000	0 1	F PC	No Yes	X Pop	H H	L
Conditional Branch	1001	0 1	PC R	No	X	H	L
Conditional JSB	1010	0 1	PC R	No Yes	X Push	H	L
Conditional RTS	1011	0 1	PC F	No Yes	X Pop	H	L
Conditional Jump MAX	1100	0 1	PC X	No	X	H H	L H
Jump MAX	1101	0 1	X	No	X	H	H
Conditional Jump Zero	1110	0 1	PC X	No No	X X	H L	L
	1111	0 1	TO BE DETERMINED BY USER				

TABLE VIII

Am29751 PROM Coding for Table VII Example

Address A_4	A_3	A_2	A_1	A_0	PROM Output O_5	O_4	O_3	O_2	O_1	O_0
0	0	0	0	0	0	1	X	1	1	1
0	0	0	0	1	0	1	X	1	1	1
0	0	0	1	0	0	1	X	1	0	0
0	0	0	1	1	0	1	X	1	0	0
0	0	1	0	0	0	1	X	1	0	1
0	0	1	0	1	0	1	X	1	0	1
0	0	1	1	0	0	0	X	1	X	X
0	0	1	1	1	0	0	X	1	X	X
0	1	0	0	0	0	1	1	0	0	0
0	1	0	0	1	0	1	1	0	0	0
0	1	0	1	0	0	1	0	0	0	0
0	1	0	1	1	0	1	0	0	0	0
0	1	1	0	0	0	1	1	0	0	1
0	1	1	0	1	0	1	1	0	0	1
0	1	1	1	0	0	1	0	0	1	0
0	1	1	1	1	0	1	0	0	1	0
1	0	0	0	0	0	1	X	1	1	0
1	0	0	0	1	0	1	0	0	0	0
1	0	0	1	0	0	1	X	1	0	0
1	0	0	1	1	0	1	X	1	0	1
1	0	1	0	0	0	1	X	1	0	0
1	0	1	0	1	0	1	1	0	0	1
1	0	1	1	0	0	1	X	1	0	0
1	0	1	1	1	0	1	0	0	1	0
1	1	0	0	0	0	1	X	1	0	0
1	1	0	0	1	1	1	X	1	X	X
1	1	0	1	0	1	1	X	1	X	X
1	1	0	1	1	1	1	X	1	X	X
1	1	1	0	0	0	1	X	1	0	0
1	1	1	0	1	0	0	X	1	X	X
1	1	1	1	0	To be determined					
1	1	1	1	1	by user					

0 = LOW   1 = HIGH   X = Don't care

Am2909 Connections:

PROM O_0 to S_0 (Pin 16)
PROM O_1 to S_1 (Pin 17)
PROM O_2 to $\overline{FE}$ (Pin 25)
PROM O_3 to PUP (Pin 26)
PROM O_4 to Zero (Pin 15)
PROM O_5 to OR_0, OR_1, OR_2, OR_3 (Pins 12, 10, 8 and 6)

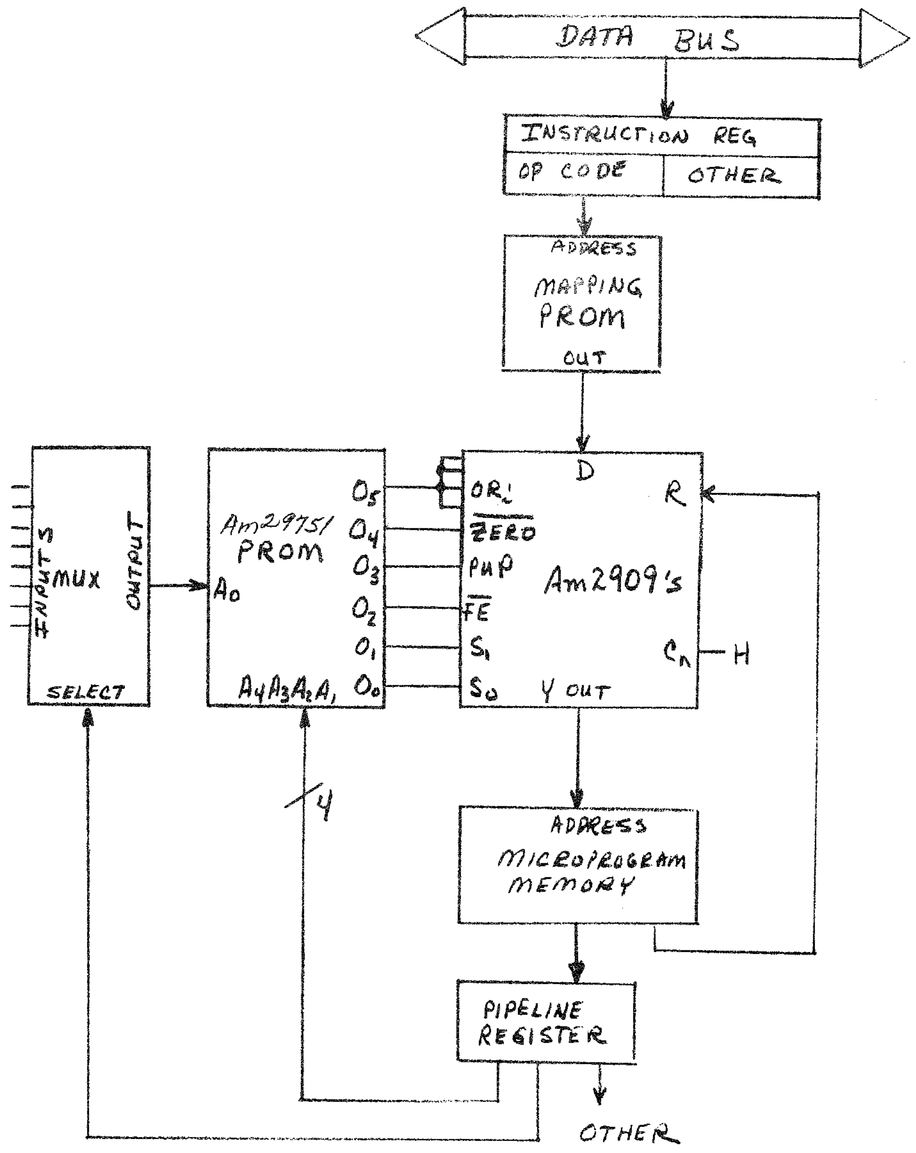


FIGURE 14: Functional Connection Diagram for Typical Computer Control Unit Using Four-bit Next Address Control Field

FORMATTING IN MICROPROGRAM MEMORY

Many engineers designing a microprogrammed machine for the first time find difficulty in determining an approach to the microprogramming problem. At Advanced Micro Devices, we have found the following approach to be the most useful for engineers attempting their first microprogramming design job. This approach involves initially assuming only one format for the microprogram word will be used. That is, the machine architecture should be determined as required for the performance desired and the total word width of the microprogram memory temporarily ignored. This can result in a fairly wide microprogram word at the onset of the design. However, by taking this approach, the design engineer does not become immediately bogged down in making several trade-offs as to the various formats useful for the microprogram words. If the machine architecture is designed to achieve the performance required and the microprogram word is expanded such that bits are available to control the various portions of the machine as required, then the initial design task becomes straight-forward.

After the initial architecture has been laid out, the design should proceed such that small groups of microinstructions are written for the key macroinstructions. Particular emphasis should be placed on the macroinstruction set requiring the highest use in the firmware operation. For example, particular emphasis should be placed on the fetch routine since it is used over and over on every machine instruction cycle. Likewise, if the machine will typically execute many register-to-register arithmetic operations, the microprogram sequence used in these instructions should be reviewed carefully. As small sequences of firmware are written, it will become apparent that perhaps some fields of the microprogram could be shared in the memory. For example, it may become apparent that the interrupt control microprogram bits and the A source operand microprogram bits never occur in the same microinstruction. Thus, perhaps if four bits are used for interrupt control and four bits are used for A source operand select, then these four bits might use the same four-bit field in the memory. This would save a four-bit field throughout the entire microprogram memory addressing space. However, it will be necessary to determine which field is present in the word in the pipeline register. Thus, one bit may be dedicated in the entire microprogram word field to distinguish between format number 1 and format number 2. When this bit is a zero, the machine recognizes format 1 and applies these four bits to the source operand control. When this bit is a logic 1, the machine recognizes that format number 2 is in the format register and applies this field to the interrupt control.

From this discussion, it should be apparent that perhaps four different formats might be utilized by the machine and a two-bit field used to identify which of the formats is in the pipeline register. In some designs, parts of the pipeline register may be repeated more than one time for some control fields. If this is the case, then it becomes necessary to select the appropriate register to be loaded by the microinstruction coming from the memory for this field. Thus, the two bits determining the format may be decoded using a two-line to four-line decoder such as the Am25LS139. The appropriate output from the Am25LS139 can enable a register with a clock enable function such as on the Am25LS07, Am25LS08 or Am25LS377 such that the microprogram memory word is loaded into the appropriate register for use by the machine.

The primary advantage of formatting the microprogram word is to save bits in width for the microprogram memory. Occasionally, as the formatting becomes more complex, it will be found that two fields are required simultaneously. Thus, this can result in an extra microinstruction being required to be able to perform all of the functions required by the microprogram control. Normally, anywhere from two to eight different formats can be found useful in the microprogramming of many machines.

2-4 different [illegible] formats used

For the design engineer doing his first microprogramming job, the emphasis should be on layout of the architecture of the machine to meet the required specification. Should the design accidentally use too many bits in microword width, it would be unfortunate but not catastrophic. The machine would still perform properly and meet specification and, in many cases, only one or two extra integrated circuits would have been used. Based on the cost of Field Programmable Read Only Memories, this cost is not unacceptable. Also, it is possible that such an approach may considerably reduce the overall design time of the machine. As the designer gains experience, he will become more proficient in formatting microprogram words and will soon find he can generate microprogram control architectures with reasonable speed.

SUMMARY

The Am2909 provides a unique solution to the microprogram memory sequence control problem. It is particularly well suited for high-performance computer control units using overlap fetch of the next address in microprogram memory. By using a microsequencer control PROM, the functional control of the next address selection can be uniquely defined. In addition, operations involving the internal stack can be selected as desired. The Am2909 is also a useful device in state machine control design. That is, control of the microprogram memory can be programmed as needed for the design.

The Am2911, being similar in design to the Am2909, provides a similar type of control in the 20-pin, 0.3" centers package. This package provides lower cost because of its reduced size and complexity. In addition, this package requires less printed circuit board area than the 28-pin Am2909 but provides almost the same functional capability.

SECTION III

Operation of the Am2900 Evaluation and Learning Kit

INTRODUCTION

The Am2900 Learning and Evaluation Kit design exemplifies the basic microprogrammed architecture of a typical minicomputer. The kit consists of a microprogram control section and a CPU. The microprogram control section contains most of the elements found in the microprogram control section of a typical computer control unit. All that need be added is an instruction register and op code mapping PROM. The CPU section of the kit contains a condition code register and shift matrix multiplexer linkage.

Toggle switches are used in the kit to control the clock select circuitry and to enter data into the microprogram memory. These toggle switches are also used to control the data display section selection points throughout the learning kit. The kit contains three LED display ports. One four-bit display is used in conjunction with the microprogram memory, the second LED display is used to view the contents of the pipeline register, and the third LED display is used to examine data at various points in the kit.

BLOCK DIAGRAM

A detailed block diagram of the Am2900 kit is shown in Figure 1. Basically, the left-hand side in the block diagram represents the microprogram control section and the right-hand side of the block diagram represents the CPU section. Likewise, various switches and multiplexers are shown in conjunction with the function described above. That is, address select switches, microprogram memory data switches, and RAM and MUX select switches are connected to the microprogram control section of the kit. Three separate LED displays are used to view the microprogram memory outputs, the pipeline register output, and the various data paths.

MICROPROGRAM SEQUENCER

One Am2909 Bipolar Microprogram Sequencer is used for the central control point for the microprogram memory. The output of the Am2909 Microprogram Sequencer represents the address of the next microinstruction to be executed. This four-bit address is applied to the four address lines of the microprogram memory. The microprogram memory consists of eight Am27S03 64-bit RAM's. These 64-bit RAM's are organized as a 16-word by 4-bit memory. Therefore, using eight of the Am27S03 Memories, a microprogram memory storage of 16 words by 32 bits is realized.

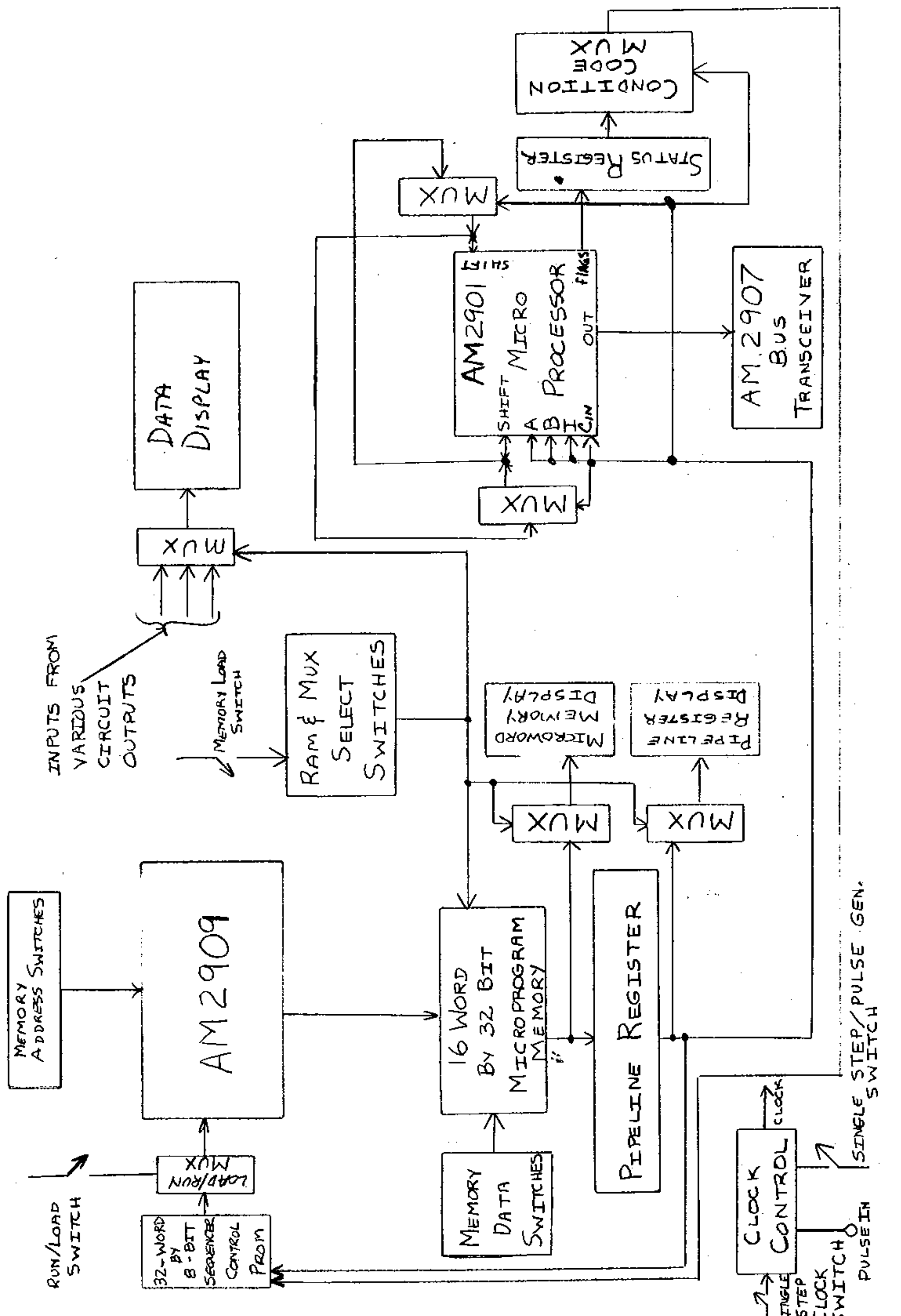


FIGURE 1: Block Diagram of the Am2900 Evaluation and Learning Kit

Four Am25LS151 eight-input Multiplexers are used at the output of the microprogram memory such that the RAM and MUX SELECT switches are used to view the contents of the RAM on the microprogram memory LED display. Thus, the user can examine the contents of each Am27S03 four bits at a time.

Eight Am2918 Four-bit Registers are used at the output of the microprogram memory to form a 32-bit pipeline register. The pipeline register contains the microinstruction currently being executed. This instruction consists of the two parts described previously. That is, the pipeline register provides the various signals to control the Am2901 and the associated shift matrix multiplexer. It also provides the necessary control signals for selecting the address of the next microinstruction. These signals include the control inputs to the Am29751 PROM and a four-bit branch address field.

The field definition for the pipeline register bits is shown in Figure 2. The definition is as follows:

Bit Position	Description
0-3	Am2901 Data Input
4-7	Am2901 B Address
8-11	Am2901 A Address
12-14	ALU Function
15	Am2901 Carry Input
16-18	Am2901 Source Operand Select
19	Am25LS253 Shift Matrix A Input (MUX_0)
20-22	Am2901 Destination Control
23	Am25LS253 Shift Matrix B Input (MUX_1)
24-27	Next Microinstruction Address Select Control
28-31	Microprogram Branch Address

The above definition shows 24 of the bits in the pipeline register are used to control the CPU section of the machine and 8 bits of the pipeline register are used for next microinstruction address control. It should be remembered that when the microprogram memory is being loaded with data, it is being loaded four bits at a time. The heavy lines in Figure 2 show the breakout of the four fields for each of the possible binary positions of the RAM and MUX SELECT switch positions. Thus, the microprogram memory is loaded four bits at a time across the eight individual fields associated with each microprogram memory word.

The next microinstruction address control four-bit field is used to select the source of the next microinstruction. This source may allow for a conditional address or an unconditional address select. The definition of the four-bit field associated with the Am2900 Evaluation and Learning Kit is shown in Figure 3. Here, the four-bit instruction field associated with the PROM is P_3, P_2, P_1, and P_0, where $P_0 = A_1$, $P_1 = A_2$, $P_2 = A_3$, and $P_3 = A_4$. The condition code enable bit controls the A_0 input to the PROM. If both the A_0 and $\overline{A_0}$ inputs contain the same address selection, the next address is unconditional. If the A_0 and $\overline{A_0}$ inputs point to different sources for the next address selection, the source of the next microinstruction address is conditional.

MICROPROGRAM MEMORY FIELD DEFINITION

RAM & MUX SELECT	7				6				5				4				3				2				1				0			
RAM LOCATION	U9				U7				U8				U6				U5				U4				U3				U2			
BIT NUMBER	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BIT DEFINITION	BR_3	BR_2	BR_1	BR_0	P_3	P_2	P_1	P_0	MUX_1	I_8	I_7	I_6	MUX_0	I_2	I_1	I_0	C_n	I_5	I_4	I_3	A_3	A_2	A_1	A_0	B_3	B_2	B_1	B_0	D_3	D_2	D_1	D_0
FIELD DEFINITION	"BRANCH ADDRESS"				NEXT MINSTRUCTION CONTROL				MUX_1	DESTINATION CONTROL			MUX_0	SOURCE SELECT			C_n	ALU			"A"				"B"				"D"			

FIGURE 2: Definition of the Field of the 32-bit Microprogram Memory Word

Am2909 Next Address Control PROM

Binary Code	$P_3P_2P_1P_0$	Function
0	0 0 0 0	Branch Register if F≠0
1	0 0 0 1	Branch Register
2	0 0 1 0	Continue
3	0 0 1 1	Branch Map (D Switches)
4	0 1 0 0	Jump-to-Subroutine if F≠0
5	0 1 0 1	Jump-to-Subroutine
6	0 1 1 0	Return-from-Subroutine
7	0 1 1 1	File Reference
8	1 0 0 0	End Loop & Pop if F=0
9	1 0 0 1	Push (and continue)
10	1 0 1 0	Pop (and continue)
11	1 0 1 1	End Loop & Pop if C_{n+4}
12	1 1 0 0	Branch Register if F=0
13	1 1 0 1	Branch Register if F_3
14	1 1 1 0	Branch Register if OVR
15	1 1 1 1	Branch Register if C_{n+4}

FIGURE 3: Instructions in the Am29751 Next Address Control PROM Supplied in the Am2900 Kit

It is this set of instructions defined in Figure 3 that is contained in the Am29751 PROM. This PROM is installed in the kit assembly in a socket (U27) such that the user can define different microinstruction source select control sets and install other PROM's in this position. The actual coding for the PROM is shown in Figure 4.

CENTRAL PROCESSOR UNIT (CPU)

The heart of the CPU section in the Am2900 Evaluation and Learning Kit is the Am2901 Four-bit Microprocessor Slice. This device contains a 16-word by 4-bit, 2-port RAM, a high-speed ALU, and the associated shifting, decoding, and multiplexing circuitry required by a high-speed cascadable element. A nine-bit microinstruction word is used to control the various functions of the device. The nine-bit microinstruction word is organized into three groups of three bits each and selects the ALU source operands, the ALU function, and the ALU destination register. While the Am2900 Evaluation and Learning Kit contains only one Am2901, the Microprocessor Slice is cascadable with full lookahead or with ripple carry, has three-state outputs, and provides various status flag outputs from the ALU.

SHIFT MATRIX CONTROL

Referring to the block diagram of Figure 1, the Am2901 Microprocessor uses multiplexers at the four bi-directional shift data pins. The most significant bit or least significant bit data transfers occur over these lines when the device is in the shift mode. The Am2900 Evaluation and Learning Kit uses two Am25LS253 Multiplexers to perform this function. The Am25LS253 Multiplexer select input (A and B) are controlled from the pipeline register MUX_0 and MUX_1 control bits. Thus, the multiplexer can perform four different functions. These include enter zeros, single length rotate, double length rotate, and arithmetic shift. These four functions can be performed in either the shift left mode or shift right mode of operation as shown in the coding table of Figure 5.

STATUS REGISTER

The status register in the Am2900 Evaluation and Learning Kit utilizes an Am25LS08 Four-bit Register with clock enable. The status register is used to store the four status flags on the Am2901 Microprocessor. These status flags are the carry output, overflow, zero detect, and sign bit.

The Am25LS08 Register clock enable function is used to hold the data in the status register during conditional tests. Thus, whenever the microprogram control function is a conditional test on one of the bits in the status register, the clock enable input is brought HIGH so the Am25LS08 Register will retain the current status word. This allows sequential testing of all four status bits in the Am2900 kit. Control for the clock enable input to the register comes from the Am29751 Microprogram Sequencer Control PROM.

Next µ Instruction Code	CCE	MUX Select	File	OR	Status Register
0	0	Reg	Inhibit	Low	Inhibit
	1	PC			
1	0	Reg	Inhibit	Low	Enable
	1				
2	0	PC	Inhibit	Low	Enable
	1				
3	0	D	Inhibit	Low	Enable
	1				
4	0	Reg	Enable &Push	Low	Inhibit
	1	PC	Inhibit		
5	0	Reg	Enable &Push	Low	Enable
	1				
6	0	File	Enable & Pop	Low	Enable
	1				
7	0	File	Inhibit	Low	Enable
	1				
8	0	File	Inhibit	Low	Inhibit
	1	PC	Enable & Pop		
9	0	PC	Enable & Push	Low	Enable
	1				
10	0	PC	Enable & Pop	Low	Enable
	1				
11	0	File	Inhibit	Low	Inhibit
	1	PC	Enable & Pop		
12	0	PC	Inhibit	Low	Inhibit
	1	Reg			
13	0	PC	Inhibit	Low	Inhibit
	1	Reg			
14	0	PC	Inhibit	Low	Inhibit
	1	Reg			
15	0	PC	Inhibit	Low	Inhibit
	1	Reg			

Figure 4A

Function Definition of Am29751
Next Microinstruction Control PROM

Next µInstruction Code	PROM Address $A_4A_3A_2A_1A_0$	0_7	0_6	0_5	0_4	0_3	0_2	0_1	0_0
0	0 0 0 0 0	1	0	0	0	1	0	1	X
	0 0 0 0 1	1	0	0	0	0	0	1	X
1	0 0 0 1 0	0	0	0	0	1	0	1	X
	0 0 0 1 1	0	0	0	0	1	0	1	x
2	0 0 1 0 0	0	0	0	0	0	0	1	X
	0 0 1 0 1	0	0	0	0	0	0	1	X
3	0 0 1 1 0	0	0	0	0	1	1	1	X
	0 0 1 1 1	0	0	0	0	1	1	1	x
4	0 1 0 0 0	1	0	0	0	1	0	0	1
	0 1 0 0 1	1	0	0	0	0	0	1	X
5	0 1 0 1 0	0	0	0	0	1	0	0	1
	0 1 0 1 1	0	0	0	0	1	0	0	1
6	0 1 1 0 0	0	0	0	0	0	1	0	0
	0 1 1 0 1	0	0	0	0	0	1	0	0
7	0 1 1 1 0	0	0	0	0	0	1	1	X
	0 1 1 1 1	0	0	0	0	0	1	1	X
8	1 0 0 0 0	1	0	0	0	0	1	1	X
	1 0 0 0 1	1	0	0	0	0	0	0	0
9	1 0 0 1 0	0	0	0	0	0	0	0	1
	1 0 0 1 1	0	0	0	0	0	0	0	1
10	1 0 1 0 0	0	0	0	0	0	0	0	0
	1 0 1 0 1	0	0	0	0	0	0	0	0
11	1 0 1 1 0	1	0	0	0	0	1	1	X
	1 0 1 1 1	1	0	0	0	0	0	0	0
12	1 1 0 0 0	1	0	0	0	0	0	1	X
	1 1 0 0 1	1	0	0	0	1	0	1	X
13	1 1 0 1 0	1	0	0	0	0	0	1	X
	1 1 0 1 1	1	0	0	0	1	0	1	X
14	1 1 1 0 0	1	0	0	0	0	0	1	X
	1 1 1 0 1	1	0	0	0	1	0	1	X
15	1 1 1 1 0	1	0	0	0	0	0	1	X
	1 1 1 1 1	1	0	0	0	1	0	1	X

X = Don't Care

FIGURE 4B: PROM Code for the Next Microinstruction Control PROM in the Am2900 Kit

CODE	MUX_1	MUX_0	UP	DOWN
0	0	0	ZERO	ZERO
1	0	1	ROTATE	ROTATE
2	1	0	DOUBLE LENGTH ROTATE	DOUBLE LENGTH ROTATE
3	1	1	DOUBLE LENGTH ARITHMETIC "0" $\rightarrow Q_0$	DOUBLE LENGTH ARITHMETIC $F_3 \rightarrow RAM_3$

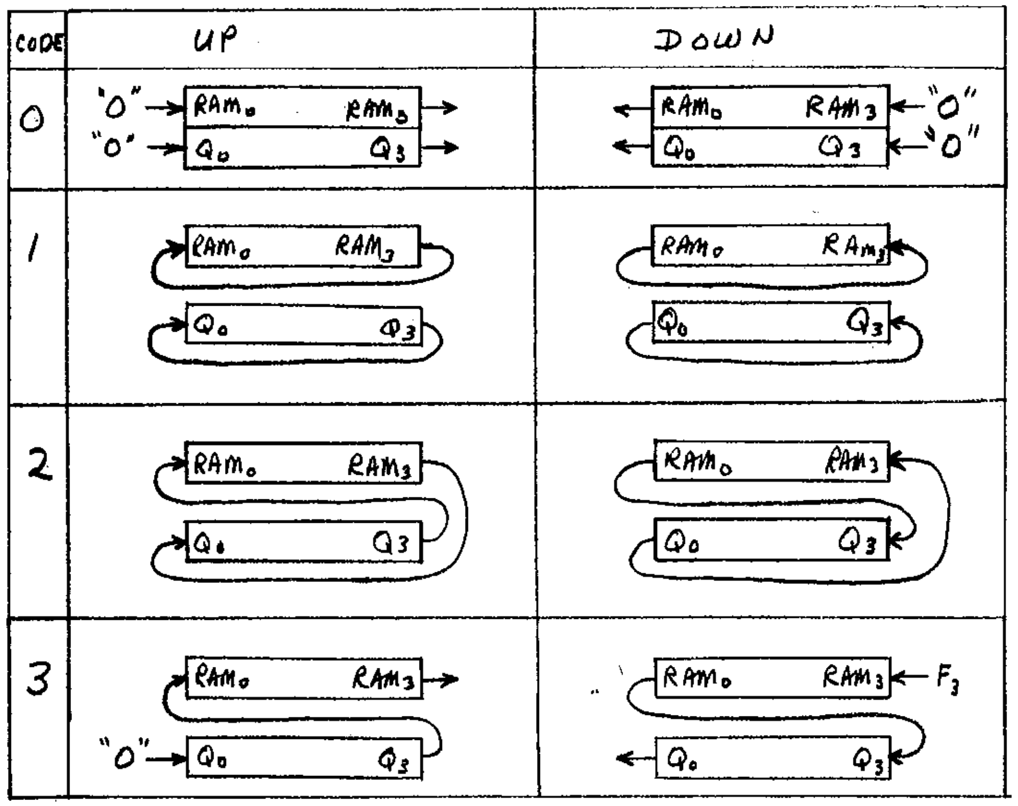


FIGURE 5: Definition of the Shift Matrix Multiplexer Control Functions

The outputs of the Am25LS08 Status Register drive an Am9309 Four-input Multiplexer. This allows one of the four status register bits to be selected at the multiplexer output as the condition code enable (CCE) input. In the Am2900 Kit, the Am9309 Four-input Multiplexer select inputs are controlled from the pipeline register. The actual control signals used are the two least significant bits of the next address control field associated with the Am29751 Microsequencer Control PROM. This allows the condition code multiplexer select inputs to share the same field as the next address control field for the microprogram sequencer.

DETAILED DESCRIPTION OF SWITCH AND DISPLAY FUNCTIONS

RAM and MUX SELECT Switches

Three toggle switches (S1, S2, S3) are used to implement the RAM and MUX SELECT control. They can access eight different fields which we will refer to as binary 0 through binary 7. The RAM and MUX SELECT switches control the three 4-bit LED displays as well as the select of an Am27S03 Microprogram Memory. Figure 2 shows the RAM location, bit definition, and the field definition for the eight RAM and MUX SELECT switch positions associated with the Random Access Memories of the pipeline register. Figure 6 shows the definition of the eight fields associated with the data display LED's as controlled by the RAM and MUX SELECT switches.

MEMORY ADDRESS Switches

The MEMORY ADDRESS switches are used to select one of the sixteen possible addresses of the microprogram memory. These MEMORY ADDRESS switches (S8, S9, S10, and S11) are used to select memory words binary 0 through binary 15. The switches connect to the D inputs of the Am2909 Microprogram Sequencer and are enabled by forcing the Am2909 internal multiplexer to select the D input. This is accomplished using the RUN/LOAD switch to be described later. Note the Am29751 PROM must be installed to inhibit OR_0, OR_1 and OR_2 from overriding MEMORY ADDRESS switches.

MEMORY DATA Switches

The MEMORY DATA switches provide the control of the individual data bits to be written into the microprogram memory. These MEMORY DATA switches (S4, S5, S6, and S7) are connected to the data inputs of the Am27S03 RAM's. Since the Am27S03 RAM's invert the data from data-in to data-out, the MEMORY DATA switches are connected such that the opposite polarity is applied to the RAM's. That is, when the MEMORY DATA switches are in the binary 0 position (pointing toward the bottom of the PC board) they apply a high logic level to the inputs to the Am27S03 RAM's. Thus, when this data is loaded into the RAM's, it will appear at the RAM output as a logic LOW level. This results in correctly entering binary 0 when the MEMORY DATA switches are in the binary 0 position.

The actual data entered into the data fields can be thought of in several different numbering schemes. For example, the A and B address fields for the Am2901 might be thought of in a hexidecimal format. These fields might also be considered to be a binary format. Data entered into the field connected to the data inputs of the Am2901 might be considered to be two's complement data or magnitude only data. Fields associated with the ALU, source operand selection and destination control might be thought of as octal represented fields. Other fields such as carry-in may be thought of as a single bit field having the capability of supplying only a logic zero or a logic one.

The following table is included for users not familiar with the standard binary number representations. This table shows the most commonly used representations for the various number schemes that are useful when using this kit.

RAM & MUX Select Switch	Data Display LED Lamps				Function
	8	4	2	1	
0	Y_3	Y_2	Y_1	Y_0	Am2909 Output
1	μP_3	μP_2	μP_1	μP_0	Am2901 Output
2	C_{n+4}	OVR	F_3	F=0	Am2901 Flags
3	Parity	CCE	$\overline{P}$	$\overline{G}$	Miscellaneous
4	ST_3	ST_2	ST_1	ST_0	Status Register
5	Q_0	Q_3	RAM_0	RAM_3	Am2901 Shift
6	BUS_3	BUS_2	BUS_1	BUS_0	Am2907 Bus
7	R_3	R_2	R_1	R_0	Am2907 Receiver

FIGURE 6: The DATA DISPLAY Field Definition as a Function of the RAM and MUX SELECT Switch Position

BINARY CODE REPRESENTATIONS

Code	Decimal	Octal	Hexidecimal	Two's Complement
0000	0	0	0	0
0001	1	1	1	1
0010	2	2	2	2
0011	3	3	3	3
0100	4	4	4	4
0101	5	5	5	5
0110	6	6	6	6
0111	7	7	7	7
1000	8	10	8	-8
1001	9	11	9	-7
1010	10	12	A	-6
1011	11	13	B	-5
1100	12	14	C	-4
1101	13	15	D	-3
1110	14	16	E	-2
1111	15	17	F	-1

MEMORY LOAD Switch

The MEMORY LOAD switch (S14) is a momentary switch used to load data into the Am27S03 Microprogram Memories. Each time the MEMORY LOAD switch is depressed, data from the MEMORY DATA switches will be loaded into the address selected by the MEMORY ADDRESS switches of the memory field selected the RAM and MUX SELECT switches. For this to occur, the RAM/LOAD SELECT switch must be in the LOAD position.

RUN/LOAD Switch

The RUN/LOAD SELECT switch (S12) is used to control the basic loading of the microprogram in memory. When the RUN/LOAD switch is in the LOAD position, the MEMORY LOAD switch can be used to enter data into the microprogram memory. When the RUN/LOAD switch is in the RUN position, the MEMORY LOAD switch is inhibited and the microprogram sequencer control PROM (U27) is enabled. Thus, the Am2900 Evaluation and Learning Kit will be in the "Operate" mode. The kit will now perform all of its functions using either the SINGLE STEP CLOCK or the CLOCK PULSE GENERATOR.

SINGLE STEP CLOCK Switch

The SINGLE STEP CLOCK switch (S15) is used to generate a clock pulse for the Am2900 Evaluation and Learning Kit. This clock pulse is applied to all clocked devices in the kit. That is, the clock pulse is applied to the Am2901, Am2909, status register, and pipeline register. The SINGLE STEP CLOCK switch is a momentary

switch that generates one clock pulse each time it is depressed. The switch is debounced using an Am9314 Latch. This ensures that one and only one clock pulse will be applied to the system for each depression of the SINGLE STEP CLOCK switch.

It is important to understand that the SINGLE STEP CLOCK switch is enabled when the RUN/LOAD switch is in the LOAD position. That is, when the microprogram memory is being loaded, the contents of the memory can be clocked into the pipeline register and executed in the Am2901 Microprocessor. This is a very useful feature for loading data into the RAM inside the Am2901.

When the SINGLE STEP/PULSE GEN switch (S13) is in the PULSE GEN mode, the SINGLE STEP CLOCK switch is inhibited and does not affect the clock input to the kit.

SINGLE STEP/PULSE GENERATOR SELECT Switch

The SINGLE STEP/PULSE GEN switch is used to select the source of the clock pulse for the Am2900 Evaluation and Learning Kit. The clock input can either be the SINGLE STEP CLOCK switch (S15) or can come from an external pulse generator. If an external pulse generator is used, it is attached to the PULSE GEN terminals at the R9 position. The pulse generator coax cable shield should be attached to the terminal closest to the bottom of the board and the pulse generator center wire should be attached to the terminal closest to the top of the board. The terminal closest to the bottom of the board is connected to ground in the Am2900 Evaluation and Learning Kit.

CLOCK TEST Turret Terminal

A turret terminal labeled CLOCK TEST is soldered into the center left-hand side of the printed circuit board. This terminal connects directly to the clock line of the Am2900 Evaluation and Learning Kit. This test point is supplied as a convenient test output for connection of an oscilloscope when using the kit with a pulse generator. This test point represents the actual clock signal applied to the Am2901 Microprocessor and Am2909 Microprogram Sequencer as well as the remainder of the kit registers.

SYNC TEST Turret Terminal

The SYNC TEST test point is connected directly to the carry output of the incrementer in the Am2909 Microprogram Sequencer. This test point is intended to be used as a convenient sync point in evaluating microprogram sequences. The limitation of using this point, of course, is that the microprogram must be written so that binary 15 of the microprogram memory addresses is used only once. The result is that a sync pulse will be generated each time the microprogram sequencer address outputs is binary 15. From this discussion, it can be seen that if the user wishes to provide a sync pulse, he can branch in microprogram control from the current address to address 15 and then branch back. This will generate a SYNC pulse at the desired time.

Thus, if the Am2900 Evaluation and Learning Kit is programmed properly, the SYNC TEST turret test pint can be used to provide synchronization to an oscilloscope.

V_{CC} and GROUND Turret Inputs

The V_{CC} and GND inputs are used to apply power to the Am2900 Evaluation and Learning Kit. The V_{CC} terminal should be connected to the +5 volt connection of a 5 volt power supply capable of delivering 2 amps of current. The GND terminal should be connected to the Ground terminal of the +5 volt power supply. Needless to say, if these terminals are accidentally reversed, many of the integrated circuits in the kit will probably be destroyed. Therefore, the user should take particular care in connecting the power supply to the kit.

ADDRESS SYNC TURRET TERMINAL

A turret terminal labeled ADDRESS SYNC is soldered into the lower left-hand corner of the printed circuit board. This terminal connects directly to the A = B output of an Am9324 Comparator. The purpose of this test point is to provide a synchronization or reference signal for the microprogram address field when using the kit with conjunction with an oscilloscope. Four bits of one comparator input field are connected to the MEMORY ADDRESS switches. The other four-bit field of the comparator is connected to the Am2909 Y output field. In this manner, the MEMORY ADDRESS switches can be used to select the microprogram address at which a sync pulse or reference pulse is provided. This provides a handy signal for use as a reference trace on the oscilloscope.

PROGRAMMING GUIDE

A handy programming guide defining the functions of the various control fields in the 32-bit microprogram word is given in Figure 7. This figure defines the five different control fields of the microprogram word. These fields include the ALU, the ALU source operand, the ALU destination, the shift array multiplexer, and the next microinstruction control fields. These five fields make up 15 bits of the microprogram memory word.

Seventeen bits of the microprogram memory word are used to provide undecoded control. These include the four-bit data field, the four-bit A address field, the four-bit B address field, the four-bit branch address field, and the carry input. In these five cases, the binary value in the field represents the binary value required for control.

The user will find the programming guide of Figure 7 a most useful tool in programming the kit. Therefore, it is recommended that the user become very familiar with this guide and all the various functions it represents.

ALTERNATE NEXT INSTRUCTION PROM

The Am2900 Evaluation and Learning Kit is designed such that a next address PROM can be used in conjunction with the OR inputs. While this technique does not demonstrate all the capability of conditional OR branching, it does allow the kit user to experiment with the OR inputs. Figure 8 shows the next address select control functions and Figure 9 shows the PROM coding for an Am27LS09 PROM with OR branching. When using such an alternate PROM with the Am2900 Evaluation and Learning Kit, the user should remember the initialization problem. That is, when power is applied to the Am2900 Kit, the status register, pipeline register and microprogram memory can turn on in either the logic 1 or logic 0 state. Thus, some of the Am2909 outputs may be forced HIGH due to the OR inputs. When the RUN/LOAD select switch is placed in the load position, it is possible for the address select switches to be overridden by the OR inputs. Therefore, when power is initially applied to the Am2900 kit, an initialization procedure is required. The recommended procedure is to place the memory address select switches to the binary 15 position and use this word in microprogram memory to clear the Am2909 next address control function such that the Am27LS09 PROM cannot conditionally force any of the OR inputs HIGH. This word should be loaded into the pipeline register using the SINGLE STEP CLOCK momentary switch. Now, the microprogram memory can be loaded in the normal fashion. In fact, an initialization procedure may be required each time a new program is loaded. If data is to be entered in the Am2901 to initialize the RAM inside the four-bit microprocessor slice, microprogram memory word binary 15 should be used for this function. Then, the user does not have to worry about the conditions of the various flags in the status register.

Am2909 NEXT ADDRESS CONTROL

CODE	FUNCTION
0	BRANCH
1	CONTINUE
2	PUSH
3	POP
4	JUMP - SUBROUTINE
5	RETURN - SUBROUTINE
6	FILE REFERENCE
7	JUMP D
8	END LOOP, F=0
9	END LOOP, F_3
10	END LOOP, OVR
11	END LOOP, C_{n+4}
12	BRANCH, F=0
13	BRANCH, " F_3
14	BRANCH, OVR
15	BRANCH, C_{n+4}

NOTE: CODE IS DECIMAL EQUIVALENT OF BINARY

ALU FUNCTION SELECT

CODE	FUNCTION
0	R plus S
1	S minus R
2	R minus S
3	R OR S
4	R AND S
5	$\bar{R}$ AND S
6	R EX-OR S
7	R EX-NOR S

SOURCE OPERAND SELECT

CODE	R	S
0	A	Q
1	A	B
2	O	Q
3	O	B
4	O	A
5	D	A
6	D	Q
7	D	O

DESTINATION SELECT

CODE	LOAD	Y OUT-PUT
0	F → Q	F
1	NONE	F
2	F → RAM	A
3	F → RAM	F
4	F/2 → RAM; Q/2 → Q	F
5	F/2 → RAM	F
6	2F → RAM; 2Q → Q	F
7	2F → RAM	F

F = ALU OUTPUT

MUX_0, MUX_1 CONTROL

CODE	UP	DOWN
0	ZERO	ZERO
1	ROTATE	ROTATE
2	DOUBLE LENGTH ROTATE	DOUBLE LENGTH ROTATE
3	ARITHMETIC (DOUBLE LENGTH ZERO → RAM_0	ARITHMEDIC DOUBLE LENGTH F_3 → RAM_3

FIGURE 7: Programming Guide for the Am2900 Evaluation and Learning Kit

ALTERNATE NEXT ADDRESS PROM

Binary Code	$P_3P_2P_1P_0$	Function
0	0000	Continue
1	0001	PUSH (and continue)
2	0010	POP (and continue)
3	0011	Jump Register if C_{n+4}
4	0100	Jump-to-Subroutine
5	0101	Return-from-Subroutine
6	0110	Jump Register
7	0111	End Loop and POP if C_{n+4}
8	1000	Jump Register if F=0
9	1001	Jump Register if F_3
10	1010	File Reference
11	1011	Jump D
12	1100	Force OR_3 if F=0
13	1101	Force OR_2 if F_3
14	1110	Force OR_1 if OVR
15	1111	Force OR_0 if C_{n+4}

FIGURE 8: Alternate Next Address PROM Instructions

Next Microinstruction Code	PROM Address	O_7	O_6	O_5	O_4	O_3	O_2	O_1	O_0
0	00000	0	0	0	0	0	0	1	X
	00001	0	0	0	0	0	0	1	X
1	00010	0	0	0	0	0	0	0	1
	00011	0	0	0	0	0	0	0	1
2	00100	0	0	0	0	0	0	0	0
	00101	0	0	0	0	0	0	0	0
3	00110	1	0	0	0	0	0	1	X
	00111	1	0	0	0	1	0	1	X
4	01000	0	0	0	0	1	0	0	1
	01001	0	0	0	0	1	0	0	1
5	01010	0	0	0	0	0	1	0	0
	01011	0	0	0	0	0	1	0	0
6	01100	0	0	0	0	1	0	1	X
	01101	0	0	0	0	1	0	1	X
7	01110	1	0	0	0	1	0	1	X
	01111	1	0	0	0	0	0	0	0
8	10000	1	0	0	0	0	0	1	X
	10001	1	0	0	0	1	0	1	X
9	10010	1	0	0	0	0	0	1	X
	10011	1	0	0	0	1	0	1	X
10	10100	0	0	0	0	0	1	1	X
	10101	0	0	0	0	0	1	1	X
11	10110	0	0	0	0	1	1	1	X
	10111	0	0	0	0	1	1	1	X
12	11000	1	0	0	0	0	0	1	X
	11001	1	1	0	0	0	0	1	X
13	11010	1	0	0	0	0	0	1	X
	11011	1	0	0	0	0	0	1	X
14	11100	1	0	0	0	0	0	1	X
	11101	1	0	1	0	0	0	1	X
15	11110	1	0	0	0	0	0	1	X
	11111	1	0	0	1	0	0	1	X

FIGURE 9: Alternate Next Address PROM Coding

SECTION IV

Assembly of the Am2900 Evaluation and Learning Kit

INSPECTING THE Am2900 KIT

Carefully unwrap all the components in the Am2900 package being careful not to damage the leads of any components. Examine all the components and the printed circuit board for damage. Count all of the parts. If any components are missing or damaged, return the entire kit to the place of purchase for exchange. After all the components have been identified and determined to be in the kit, the assembly process should begin.

The assembly instructions in this kit are intended to allow the builder the capability of easily assembling and testing the kit with a minimum of electronic test equipment. Should trouble-shooting be required, a volt/ohmmeter will be a handy test aid. All components are intended to be mounted flat on the printed circuit board.

EXAMINING THE PRINTED CIRCUIT BOARD

The printed circuit board supplied with the Am2900 Evaluation and Learning Kit is a high quality, two-sided printed circuit board with plated through holes. The board contains markings showing all the component locations and nomenclature. Before assembling the components, the printed circuit board should be examined very carefully in order to assure quality workmanship throughtout. That is, the board should be examined for any apparent briding between traces, improper plating or lack of holes in pad locations. Advanced Micro Devices has made every effort to ensure the printed circuit board is of the highest quality to provide long life to this kit.

TESTING

Throughout the assembly of the Am2900 Evaluation and Learning Kit, various tests will be performed using the LED display as an indicator. In order to accomplish this testing method, it will be necessary to have two jumper wires of approximately 12 inches in length to be used as intermediate connection aids. It is recommended that a No. 22 or No. 24 solid insulated wire be used with approximately 1/8 inch of the sleeving stripped from each end. Throughout the testing, these wires will be inserted in various holes in the printed circuit board to provide a connection path to the LED. In all cases, the test wire should NOT be soldered to the PC board. It should only be inserted into the hole and held in place by hand.

If a defective component is found, fill out the warranty exercise card and return the component and card to Advanced Micro Devices for replacement. Please be <u>sure</u> the component is defective before returning it.

It is very important not to touch any pin or PC track except that specified as the final location while using the test wire. In other words, do not count pins around the IC's location by touching the wire to the printed circuit board while counting. The reason for this, of course, is that the wire could cause accidental shorts between two undesired points.

GENERAL TESTING

It is possible to perform general testing on any integrated circuit printed circuit board that is operating in a single step clock mode by using a voltmeter. Normally, such test procedures involve using the voltmeter to evalute logic HIGH's and logic LOW's. By definition, a logic HIGH is any voltage above 2.4V and a logic LOW is any voltage below 0.4V.

There are times when an output can be in the three-state mode; that is, the high impedance state. When using a voltmeter to measure this voltage, the impedance of the voltmeter may affect the reading. When a very high impedance voltmeter is used, the voltage reading may be about 1.5 volts. In other cases, however, an output in the high impedance state may read a voltage above 2.4 volts and the user may erroneously assume a logic HIGH state. In order to be sure of the voltage of three-state outputs, it is recommended that a voltage reading between the output and ground be taken and a voltage reading between the output and the V_{CC} be taken. If the sum of the voltage readings just taken is equal to the voltage between V_{CC} and ground, then there is no measurement problem and the output voltage is known. Remember, a HIGH and a high impedance can measure the same voltage. If the sum of the voltages between the output and ground and the output and V_{CC} is considerably different than the normal V_{CC} to ground voltage, then the voltmeter impedance is affecting the reading and most likely the output is in the high impedance state. Such a problem should not be encountered with this kit; however, the user should be aware of good trouble-shooting practices for three-state outputs.

SOLDERING IRON

The soldering iron used in assembling the Am2900 Evaluation and Learning Kit should be a low wattage, pencil-type soldering iron. The wattage of the soldering iron should be approximately 30-40 watts and a sharp, chisel-type point performs best. If an excessively hot soldering iron is used in the assembly of this kit, the user may accidentally lift the circuit traces from the printed circuit board. This will result in permanent damage to the printed circuit board.

TURRET TERMINALS

The Am2900 series kit is supplied with seven turret terminals. Two of these terminals are used to connect power and ground to the PC board. Two of the terminals are used to connect a coaxial cable to the board for use with a pulse generator. Three of the terminals are used for test points on the board when evaluating the Am2900 series components with an oscilloscope.

Assemble the seven turret terminals to the printed circuit board by installing them in the front of the board. If a swaging tool is available, it should be used to mechanically secure the terminals to the printed circuit board. If no such tool is available, the terminals may be soldered directly to the printed circuit board without swaging. In either case, solder the terminals to the PC board.

Install the seven terminals in the following positions:

Assembled	Location
☐	V_{CC}
☐	GND
☐	Address SYNC
☐	Sync Test
☐	Clock Test
☐	Pulse Gen (close to R9)
☐	

At this point, two wires should be attached to the V_{CC} and ground terminals to check for power supply shorts on the printed circuit board. An ohmmeter can be connected between the V_{CC} and ground wires and infinite resistance should be measured. If no ohmmeter is available, a current limited power supply can be connected and the current meter reading should be 0 amperes.

If a short circuit is found between these input terminals, the board should be examined very carefully for shorts. If none can be found but very low ohmic resistance is measured, do not continue assembling the kit. Instead, contact the local AMD sales representative or distributor.

RESISTORS

During this portion of the assembly, all resistors used in the Am2900 kit will be installed. Identify all resistors by their color coded bands and wattage. Properly bend the leads on each resistor such that they can be easily inserted in the holes on the printed circuit board. Install each resistor, solder both ends in place and then clip the leads on the back of the PC board. The resistors should lay flat on the printed circuit board and sharp bends at the resistor ends should be avoided. Install the resistors in the order listed below.

Assembled	Location	Value	Wattage
☐	R1	2200Ω	1/4 W
☐	R2	2200Ω	1/4 W
☐	R3	2200Ω	1/4 W
☐	R4	2200Ω	1/4 W
☐	R5	2200Ω	1/4 W
☐	R6	2200Ω	1/4 W
☐	R7	2200Ω	1/4 W
☐	R8	2200Ω	1/4 W
☐	R9	51Ω	1/2 W
☐	R10	1000Ω	1/4 W
☐	R11	51Ω	1 W
☐	R12	51Ω	1 W
☐	R13	51Ω	1 W
☐	R14	51Ω	1 W
☐	R15	390Ω	1/4 W
☐	R16	390Ω	1/4 W
☐	R17	390Ω	1/4 W
☐	R18	390Ω	1/4 W
☐	R19	390Ω	1/4 W
☐	R20	390Ω	1/4 W
☐	R21	390Ω	1/4 W
☐	R22	390Ω	1/4 W
☐	R23	390Ω	1/4 W
☐	R24	390Ω	1/4 W
☐	R25	390Ω	1/4 W
☐	R26	390Ω	1/4 W

After all the resistors are installed and soldered in place, carefully examine the printed circuit board to be sure no shorts have been accidentally introduced. Again, the ohmmeter reading between the V_{CC} and ground terminal should be infinity. If a power supply is connected to the V_{CC} and ground terminals, the reading should be 0 amperes.

CAPACITORS

Two types of capacitors are supplied with the Am2900 Kit. There are 14 0.022μF high-frequency decoupling capacitors and 3 polarized 60μF decoupling capacitors.

CAUTION: Capacitors C7, C13, and C14 are polarized. Be sure the positive (+) end is placed in the positive (+) position on the PC board.

The leads on the three polarized capacitors should be formed so the devices will easily lay on the printed circuit board. The high-frequency decoupling capacitors will install directly into the printed circuit board. Install the capacitors and solder into place one at a time in the following order:

Assembled	Location	Value	Polarized
☐	C1	0.022μF	No
☐	C2	0.022μF	No
☐	C3	0.022μF	No
☐	C4	0.022μF	No
☐	C5	0.022μF	No
☐	C6	0.022μF	No
☐	C7	60.0μF	Yes
☐	C8	0.022μF	No
☐	C9	0.022μF	No
☐	C10	0.022μF	No
☐	C11	0.022μF	No
☐	C12	0.022μF	No
☐	C13	60.0μF	Yes
☐	C14	60.0μF	Yes
☐	C15	0.022μF	No
☐	C16	0.022μF	No
☐	C17	0.022μF	No

After installing all the capacitors, carefully check the printed circuit boards for any shorts that may have been introduced during this installation. Also, carefully check the polarity of the three polarized capacitors with respect to the marking on the printed circuit board. If the polarity of these capacitors does not match that shown on the printed circuit board, remove these capacitors and install them in the proper polarity connection. At this point, an ohmmeter may again be used to check for shorts. It should be remembered that the decoupling capacitors now have to be charged to the voltage used in the ohmmeter to supply the measuring current. Thus, the ohmmeter needle will jump to a resistance value and decay towards infinity while the capacitors are charging. At this point, some very high resistance value may be measured as the leakage value of the capacitors. If the power supply test is used, only an extremely negligible current should be read on the ampere meter. If a high current value is read or a very low continuous resistance is measured using the ohmmeter, a short may have been introduced to the PC board during this operation. If none can be found, then one of the capacitors may be defective. If all else fails, remove the capacitors one at a time until the short circuit disappears. Then, reinstall the remaining capacitors and repeat the test.

SWITCHES

Carefully identify the 15 switches supplied in the Am2900 kit. The 13 toggle switches may be installed in any fashion and their position is not critical. The two momentary switches must be installed in the proper fashion if the polarity of the momentary contact is to be in the proper direction.

CAUTION: The momentary switches require insertion in the proper polarity if the Am2900 kit is to operate satisfactorily. Read the momentary switch installation in detail before assembling switches S14 and S15.

Install the 13 toggle switches first. Each switch should be fitted in position and soldered in place. Care should be taken when the switches are being installed such that they are not damaged during the assembly. This could result if the printed circuit board is turned upside down to solder the switches in place and too much pressure is applied to the top of the board.

Assembled	Location	Type
☐	S1	Toggle
☐	S2	Toggle
☐	S3	Toggle
☐	S4	Toggle
☐	S5	Toggle
☐	S6	Toggle
☐	S7	Toggle
☐	S8	Toggle
☐	S9	Toggle
☐	S10	Toggle
☐	S11	Toggle
☐	S12	Toggle
☐	S13	Toggle

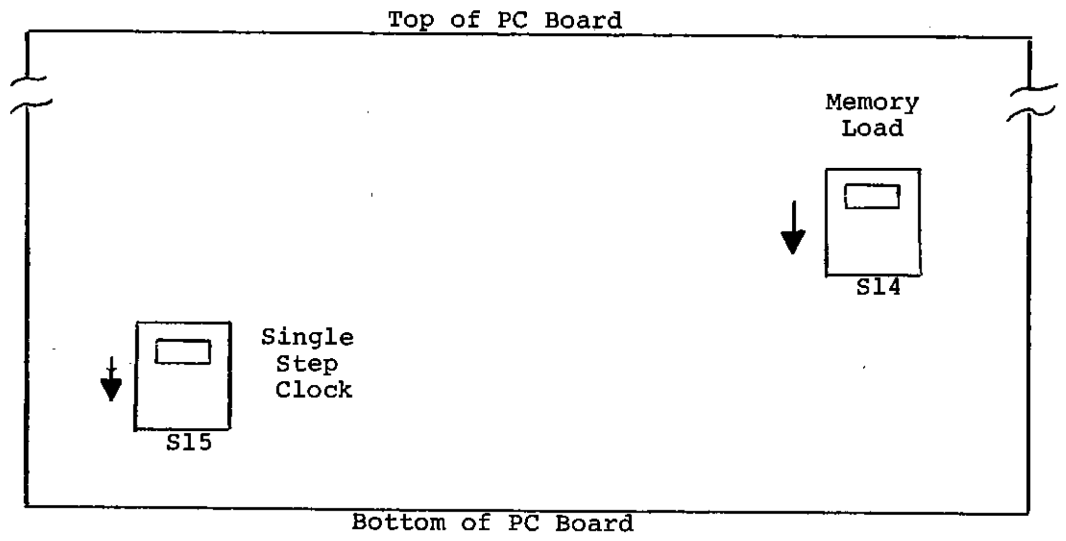
NOTE: The plastic covers should be removed from the toggle switch handles and discarded.

MOMENTARY SWITCHES

CAUTION: Important - Read First

Carefully test the action of the momentary switches to determine the spring loaded direction. Both switches must be installed such that the switch handle normally points to the top of the PC board and is pulled towards the bottom of the PC board in the momentary position. Install the switches as follows.

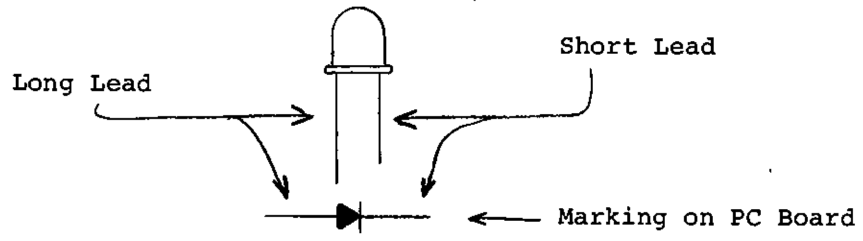
Assembled	Location	Type
☐	S14	Momentary
☐	S15	Momentary



The switches S14 and S15 should normally point toward the top of the PC board and should have the momentary motion toward the bottom of the PC board.

LIGHT EMITTING DIODES (LED's)

The Am2900 Evaluation Kit contains 12 light emitting diodes (LED's) to display data and various other signals. The LED's must be installed in the printed circuit board in the proper polarity. The correct orientation for the LED is shown in the figure below. Note that the polarity of the LED is determined by the lead length of the device.



CAUTION: The Light Emitting Diodes should be installed in the printed circuit board with approximately 1/16 inch clearance between the diode base and the printed circuit board. A wooden toothpick is supplied with the Am2900 series kit to be used as an installation tool. The LED should be installed in the printed circuit board and the wooden toothpick placed between the LED base and the printed circuit board as a spacer. The LED should be pressed firmly against the toothpick, positioned properly, the leads bent on the back of the board such that the LED will be held in a straight, vertical position. Both leads of the LED should then be soldered in place and the toothpick removed.

After all 12 LED's have been installed, they should be bent towards the bottom of the board to about a 30° angle from vertical. This will allow the LED's to point directly at the user of the Am2900 kit. Thus, maximum brilliance of the LED's will be seen by the user.

Install the LED's as follows.

Assembled	Location
☐	Data - 1
☐	Data - 2
☐	Data - 4
☐	Data - 8
☐	Memory Address - 1
☐	Memory Address - 2
☐	Memory Address - 4
☐	Memory Address - 8

☐	Pipeline Register - 1
☐	Pipeline Register - 2
☐	Pipeline Register - 4
☐	Pipeline Register - 8

The LED's may be tested in the following manner. Connect the V_{CC} terminal of the PC board to the positive terminal of the power supply. Touch a wire connected to the ground terminal of the power supply to the points indicated below and the appropriate LED should illuminate.

LED	IC Location-Pin Number
Data Display 8	U39-6
Data Display 4	U38-6
Data Display 2	U37-6
Data Display 1	U36-6
Pipeline Register 8	U26-12
Pipeline Register 4	U26-9
Pipeline Register 2	U26-7
Pipeline Register 1	U26-4
Microword Memory 8	U25-6
Microword Memory 4	U24-6
Microword Memory 2	U23-6
Microword Memory 1	U22-6

RUBBER FEET

The Am2900 Evaluation and Learning Kit is supplied with six rubber feet, six #4 machine screws, and six #4 nuts. Attach the rubber feet to the back of the printed circuit board using the machine screws and nuts. The machine screw head should be inside the rubber foot with the nut secured to the top of the printed circuit board. By attaching the rubber feet at this point, a convenient support base is supplied for the bottom of the printed circuit board.

Assembled	Description
☐	Six rubber feet

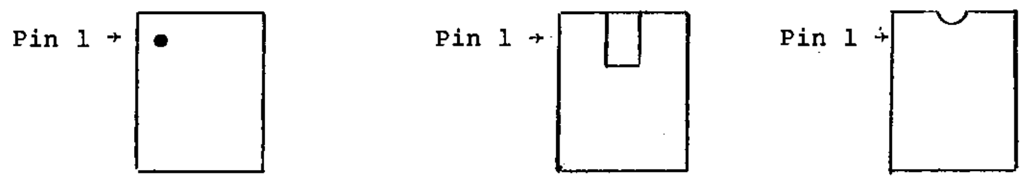
If the reader is not familiar with IC's, he should review the remainder of the assembly procedure first.

INSTALLING THE INTEGRATED CIRCUITS

Carefully identify each integrated circuit so that there will be no confusion with installing the devices in their proper location. The assembly of the Am2900 kit is defined such that a minimum of trouble-shooting will be required if accidental solder bridges should result during assembly. In addition, this procedure should allow easy identification of a defective component should this be required.

The assembly procedure for the integrated circuits in the Am2900 kit will be based on identifying various operational blocks of the kit. That is, the multiplexer for the data and microword memory selector IC's will be installed first. Then the Am2909 will be installed and the multiplexer used to test the outputs of the 2909. Next, the multiplexer for looking at the RAM memory output will be installed and tested. Following this, the eight 16-word by 4-bit Random Access Memories will be loaded onto the PC board and functionally tested. This process of identifying particular blocks of the circuitry and installing the integrated circuits into the printed circuit board will be followed and each portion tested until the entire kit is assembled.

Each integrated circuit in the Am2900 kit must be installed in the proper orientation. The PC board is marked with a (●) representing the pin 1 position for each IC. The IC's denote pin 1 in one of the following ways.



It is very important not to touch any pin or PC track except that specified as the final location while using the test wire. In other words, do not count pins around the IC's location by touching the wire to the printed circuit board while counting. The reason for this, of course, is that the wire could cause accidental shorts between two undesired points.

<u>Note</u> there are two '157 multiplexers in the Am2900 Kit. One is a SN74<u>S</u>157 and the second is a Am25<u>LS</u>157. Be sure to install them in the proper location.

PROM SOCKET

One 16-pin socket has been supplied with the Am2900 kit. It is to be installed in the U27 position and will hold the Am29751 32-word by 8-bit PROM. This device controls the next address function select of the Am2909 bipolar microprocessor sequencer. A socket has been supplied for this device (U27) so that the user can modify the instruction set at a later time if he desires. All that is required is to purchase a new Am29751 PROM and program the device such that it executes the new control functions designed by the user.

Install the 16-pin socket in position U27.

Assembled	Device	Location
☐	16-pin socket	U27

Do not install the Am29751 PROM yet.

DATA DISPLAY MULTIPLEXER

Select four of the Am25LS151 devices and install them in circuit positions U36, U37, U38 and U39.

Assembled	Location	Device Installed
☐	U36	Am25LS151
☐	U37	Am25LS151
☐	U38	Am25LS151
☐	U39	Am25LS151

Operation of the multiplexer may be tested as follows. Connect the PC board to the +5 volt power supply. Put the RAM and MUX SELECT control switches into the logic zero position with the bat handles pointing towards the bottom of the board. Turn ON the power. All four LED's associated with the DATA DISPLAY should be ON. Connect one end of a test probe or wire to the ground terminal of the power supply. By using the other end, proper operation of each data input to the multiplexer can be tested by grounding the appropriate input pin. The testing is accomplished by following the table shown below. With the RAM and MUX SELECT switches in the 000 position (binary 0), each D0 input of U36, U37, U38 and U39 are selected. By sequentially grounding each of these inputs (pin 4), the appropriate weight LED's 1, 2, 4 and 8 can be turned OFF. The table below shows the correct RAM and MUX SELECT position associated with each input for a complete test at this point.

RAM and MUX Select Switches	MUX Input	Data Display-1	Data Display-2	Data Display-4	Data Display-8
(Binary)0	D0	U36-4	U37-4	U38-4	U39-4
1	D1	U36-3	U37-3	U38-3	U39-3
2	D2	U36-2	U37-2	U38-2	U39-2
3	D3	U36-1	U37-1	U38-1	U39-1
4	D4	U36-15	U37-15	U38-15	U39-15
5	D5	U36-14	U37-14	U38-14	U39-14
6	D6	U36-13	U37-13	U38-13	U39-13
7	D7	U36-12	U37-12	U38-12	U39-12

MICROWORD MEMORY DISPLAY MULTIPLEXER

The MICROWORD MEMORY display is used to view the output of the microprogram memory four bits at a time. Select four Am25LS151 multiplexers and install them in U22, U23, U24 and U25.

Assembled	Location	Device Installed
☐	U22	Am25LS151
☐	U23	Am25LS151
☐	U24	Am25LS151
☐	U25	Am25LS151

The microword memory display multiplexer may be tested in exactly the same fashion as was the data display multiplexer.

By sequentially grounding each of the inputs described in the table below, the microword memory multiplexer can be tested. The table below shows the correct RAM and MUX SELECT switch positions associated with each input if the complete test is desired at this point.

RAM and MUX Select Switches	MUX Input	Microword Memory-1	Microword Memory-2	Microword Memory-4	Microword Memory-8
0	D0	U22-4	U23-4	U24-4	U25-4
1	D1	U22-3	U23-3	U24-3	U25-3
2	D2	U22-2	U23-2	U24-2	U25-2
3	D3	U22-1	U23-1	U24-1	U25-1
4	D4	U22-15	U23-15	U24-15	U25-15
5	D5	U22-14	U23-14	U24-14	U25-14
6	D6	U22-13	U23-13	U24-13	U25-13
7	D7	U22-12	U23-12	U24-12	U25-12

When the above points are gounded, the appropriate MICROWORD MEMORY LED will turn OFF.

RUN/LOAD SELECT

Select an Am9314 integrated circuit and install it in the U11 position. This I. C. is used as a switch debouncer.

Assembled	Location	Device Installed
☐	U11	Am9314

In order to test the LOAD/RUN switch operation, insert the test jumper wire in U11, pin 13 and U26, pin 4. With the RUN/LOAD switch in the LOAD position, the pipeline register-1 LED will be off. When the RUN/LOAD switch is changed to the RUN position, the pipeline register-1 LED will be on. Remove the test jumper wire.

CLOCK CIRCUITRY

The clock circuitry associated with the Am2900 series Evaluation and Learning Kit is used to select either the SINGLE STEP CLOCK or the PULSE GENERATOR clock. Select an Am9309 multiplexer and an SN74S157 multiplexer.

CAUTION: The Am2900 Evaluation and Learning Kit includes two types of '157's. One device is an Am25LS157 while the second device is an SN74S157. The device to be installed in U28 is the SN74S157.

Intall these two circuits as follows:

Assembled	Location	Device Installed
☐	U29	Am9309
☐	U28	SN74S157

The clock circuitry can be tested in the following manner. Connect the test jumper wire between the CLOCK TEST turret terminal and U26, pin 4. Place the RUN/LOAD select switch in the LOAD position. Place the SINGLE STEP CLOCK/PULSE GEN select switch in the SINGLE STEP CLOCK position. Connect the power supply to the printed circuit board and apply power. The PIPELINE REGISTER-1 LED should be OFF. Depress the SINGLE STEP CLOCK momentary switch and the PIPELINE REGISTER-1 LED lamp should turn ON.

Change the RUN/LOAD switch to the RUN position. The PIPELINE REGISTER-1 LED lamp should be OFF. Again, depress the SINGLE STEP CLOCK momentary switch and the PIPELINE REGISTER-1 LED lamp should turn ON. Next, switch the SINGLE STEP/PULSE GENERATOR switch to the PULSE GEN position. The PIPELINE REGISTER-1 LED lamp should be OFF. Using a second test jumper wire, momentarily connect U29, pin 3 to V_{CC}. The PIPELINE REGISTER-1 LED display should turn ON. Care should be taken during this testing such that no pins are accidentally shorted. Remove both test jumper wires. Connect one end of a test jumper wire to U21, pin 1 and connect the other end of this jumper wire to U26, pin 4. Place the RUN/LOAD toggle switch in the LOAD position. The PIPELINE REGISTER-1 LED display should be OFF. Now switch the RUN/LOAD toggle switch to the RUN position. The PIPELINE REGISTER-1 LED lamp should be ON. Remove the test jumper wire.

MICROPROGRAM SEQUENCER

The microprogram sequencer controls the 16-word by 32-bit writeable microprogram control memory. Select the Am25LS157 multiplexer and install it in U21. Select the Am2909 Microprogram Sequencer and install it in the U10 position. Select the Am29751 PROM and install it in the PROM socket, U27.

Assembled	Location	Device Installed
☐	U21	Am25LS157
☐	U10	Am2909
☐	U27	Am29751

The Am2909 MEMORY ADDRESS SELECTION switches can be tested as follows. Apply power to the printed circuit board and place the RUN/LOAD select switch in the LOAD position. Place the RAM and MUX SELECT switches in the binary 0 position. Now exercise the MEMORY ADDRESS switches while viewing the data display LED's. When the MEMORY ADDRESS switches 1, 2, 4, and 8 are LOW or in the logic 0 position, the data display LED's should be OFF. MEMORY ADDRESS switches 1, 2, 4, and 8 should control data display LED's 1, 2, 4, and 8, respectively. That is, with MEMORY ADDRESS-1 switch (S8) in the logic 0 position, DATA DISPLAY-1 LED should be OFF. When this switch is changed to the logic 1 position, the DATA DISPLAY-1 LED should turn ON. The same should be true for the remaining three switches with respect to the DATA DISPLAY LED's.

PIPELINE DISPLAY BUFFER

An Am25LS158 multiplexer is used as a four-bit buffer to drive the PIPELINE REGISTER LED lamps. Select the Am25LS158 and install it in the U26 position.

Assembled	Location	Device Installed
☐	U26	Am25LS158

Operation of the pipeline display buffer can be tested as follows. Connect the PC board to the +5V power supply. When power is applied, all four LED's of the pipeline register should be ON. Connect one end of the test wire to the power supply ground. With the other end, momentarily ground U26, pin 2 and the PIPELINE REGISTER-1 LED lamp should turn OFF. When U26, pin 5 is grounded, the PIPELINE REGISTER-2 LED should turn OFF. When U26, pin 11 is grounded, the PIPELINE REGISTER-4 LED should turn OFF. When U26, pin 14 is grounded, the PIPELINE REGISTER-8 LED should turn OFF. Remove the test jumper wire.

MICROPROGRAM MEMORY

Now we will install the 16-word by 32-bit microprogram memory. It consists of eight Am27S03 64-bit Random Access Memories. Select one Am25LS138 and eight Am27S03 devices. The Am25LS138 is installed in the U1 position and the Am27S03's are installed in the U2-U9 positions.

Assembled	Location	Device Installed
☐	U1	Am25LS138
☐	U2	Am27S03
☐	U3	Am27S03
☐	U4	Am27S03
☐	U5	Am27S03
☐	U6	Am27S03
☐	U7	Am27S03
☐	U8	Am27S03
☐	U9	Am27S03

The microprogram memory may be tested as follows. Apply power to the printed circuit board. Place the RUN/LOAD switch in the LOAD position. The RAM and MUX SELECT switch is used to select memories U2-U9. The MEMORY ADDRESS switches are used to select one of the 16 addresses in the memory. The MEMORY DATA switches are used to select the actual data to be loaded into the memory. The MEMORY LOAD momentary switch is used to enter data into the Random Access Memory. With power applied, the following procedure is recommended. Set the RAM and MUX SELECT switches to binary zero, the MEMORY DATA switches to binary zero and the MEMORY ADDRESS switches to zero. Depress the MEMORY LOAD momentary switch. The four LED's (8, 4, 2, 1) of the MICROWORD MEMORY display should all be OFF. Advance the four MEMORY DATA switches to logic one. Again, depress the MEMORY LOAD switch. The four MICROWORD MEMORY LED"s should now be ON. This sequence of loading zeroes and loading ones into each four-bit memory word may be continued through the remaining 15 addresses as selected by the four MEMORY ADDRESS switches. After completing all 16 words for this memory, the RAM and MUX SELECT switch should be advanced to binary one. This procedure should be repeated for this position. Next, the RAM and MUX SELECT switch should be advanced to binary two and the procedure repeated and so forth until the RAM and MUX SELECT switches have been advanced through binary 7 and the procedure repeated for each position.

PIPELINE REGISTER

The pipeline register is used to hold the microinstruction currently being executed. Select eight Am2918's and one Am25LS138. Install the eight Am2918's in U13-U20, respectively and the Am25LS138 in U12.

Assembled	Location	Device Installed
☐	U13	Am2918
☐	U14	Am2918
☐	U15	Am2918
☐	U16	Am2918
☐	U17	Am2918
☐	U18	Am2918
☐	U19	Am2918
☐	U20	Am2918
☐	U12	Am25LS138

The microword or pipeline register can be tested as follows. Place the RUN/LOAD switch in the LOAD position, the SINGLE STEP/ PULSE GENERATOR switch in the SINGLE STEP position. The procedure used will be similar to the procedure used to check the Random Access Memory previously. Set the RAM and MUX SELECT switch to binary zero, the MEMORY DATA select switch to binary zero, and the MEMORY ADDRESS switch to binary zero. Depress the MEMORY LOAD switch. The MICROWORD MEMORY LED's should be OFF. Next, depress the SINGLE STEP CLOCK momentary switch. The PIPELINE REGISTER LED display should now also be OFF. Then set the MEMORY DATA select switches to binary 15 and depress the MEMORY LOAD switch. Next, depress the SINGLE STEP CLOCK switch and now the PIPELINE REGISTER LED's should read binary 15. This procedure should be repeated for RAM and MUX SELECT switch positions binary one through binary seven. This will test all the Am2918 registers as well as the Am25LS138.

BUS TRANSCEIVER

The Am2907 is a large scale integration (LSI) bus transceiver. It is included in the Am2900 kit for evaluation purposes. That is, this device does not connect directly to the input of the Am2901 bipolar microprocessor. The Am2907 is one of the first integrated circuits built by Advanced Micro Devices in the new space-saving, 20-pin package. Select the Am2907 and install it in the U35 position.

Assembled	Location	Device Installed
☐	U35	Am2907

The Am2907 operation can be tested in the following manner. Connect the power supply to the printed circuit board and apply power. Put the CLOCK SELECT switch in the SINGLE STEP position and the RUN/LOAD switch to the LOAD position. Set the RAM & MUX SELECT switches to the binary 6 position. Operate the SINGLE STEP CLOCK momentary switch one time and the four DATA DISPLAY LED's should be OFF. Change the RAM & MUX SELECT switches to the binary 7 position and the four DATA DISPLAY LED's should all be ON. Next, change the RAM & MUX SELECT switches to the binary 1 position and the DATA DISPLAY LED's should be ON. Using the test wire, momentarily ground U34, pin 39. The DATA DISPLAY-8 LED should turn OFF. When the ground is removed from U34, pin 39, the DATA DISPLAY-8 LED should turn ON. Using the test wire, momentarily ground U34, pin 38. The DATA DISPLAY-4 LED should turn OFF. If the ground is removed, the LED will turn ON. Using the test wire, momentarily ground U34, pin 37. The DATA DISPLAY-4 LED should turn OFF. Remove the ground and the LED should turn ON. Using the test wire, momentarily ground U34, pin 36. The DATA DISPLAY-1 LED should turn OFF. When the ground is removed, the LED should turn ON.

Change the RAM & MUX SELECT switches to the binary 6 position. The four DATA DISPLAY LED's should be OFF. If they are not, momentarily depress the SINGLE STEP CLOCK switch. Next, apply a ground to U34, pin 39, and depress the SINGLE STEP CLOCK momentary switch. The DATA DISPLAY-8 LED should turn ON. Next, apply a ground to U34, pin 38, using the test wire and depress the SINGLE STEP CLOCK momentary switch. The DATA DISPLAY-4 LED should turn ON and the DATA DISPLAY-8 LED should turn OFF. Next, apply a ground to U34, pin 37, and depress the SINGLE STEP CLOCK momentary switch. The DATA DISPLAY-2 LED should turn ON and the DATA DISPLAY-4 LED should turn OFF. Finally, apply a ground to U34, pin 36, and depress the SINGLE STEP CLOCK momentary switch. The DATA DISPLAY-1 LED should turn ON and the DATA DISPLAY-2 LED should turn OFF. With the ground removed, depress the SINGLE STEP CLOCK momentary switch and the DATA DISPLAY-1 LED should turn OFF.

Switch the RAM & MUX SELECT switches to the binary 7 position. Apply a ground to U34, pin 39, and depress the SINGLE STEP CLOCK switch. The DATA DISPLAY-8 LED should turn OFF. Now apply the ground to U34, pin 38, and depress the SINGLE STEP CLOCK switch. The DATA DISPLAY-4 LED should turn OFF and the DATA DISPLAY-8 LED should turn ON. Now apply the ground to U34, pin 37, and depress the SINGLE STEP CLOCK switch. The DATA DISPLAY-2 LED should turn ON and the DATA DISPLAY-4 LED should turn OFF. Now apply the ground to U34, pin 36, and depress the SINGLE STEP CLOCK switch. The DATA DISPLAY-1 LED should turn OFF and the DATA DISPLAY-2 LED should turn ON. Remember, it is always important while using the test wire not to touch any pin or PC track except that specified

as the final location. In other words, do not count pins around the IC's location by touching the wire to the printed circuit board. If all of the above tests have given the correct LED readout, the Am2907 operates properly.

A STATUS REGISTER

Select the Am25LS08 and install it in the U31 position. The Am25LS08 is used as the status register for the Am2901 microprocessor. This status register holds the carry output, overflow, sign bit, and zero detect flags. Install the Am25LS08 in the U31 position.

Assembled	Location	Device Installed
☐	U31	Am25LS08

The Am25LS08 can be tested as follows. Place the RAM and MUX SELECT switches to the binary 6 position and the MEMORY DATA switches to the binary 0 position. Depress the MEMORY LOAD momentary switch and then the SINGLE STEP CLOCK momentary switch. This is required to properly load the pipeline register so that the Am27LS09 PROM points to memory word 1 and provides a LOW signal to the enable input of the Am27LS08 in the U31 position. Next, place the RAM and MUX SELECT switches in the binary 4 position. All four DATA DISPLAY LED's should be ON. Using the test wire, ground U34, pin 11 and depress the SINGLE STEP CLOCK momentary switch. The DATA DISPLAY-1 LED should turn OFF. Next, apply the ground to U34, pin 31 and depress the SINGLE STEP CLOCK momentary switch. The DATA DISPLAY-2 LED should turn OFF and the DATA DISPLAY-1 LED should turn ON. Next, apply the ground to U34, pin 34 and depress the SINGLE STEP CLOCK momentary switch. The DATA DISPLAY-4 LED should turn OFF and the DATA DISPLAY-2 LED should turn ON. Next, apply the ground to U34, pin 33 and depress the SINGLE STEP CLOCK switch. The DATA DISPLAY-8 LED should turn OFF and the DATA DISPLAY-4 LED should turn ON. Now remove the test wire and again depress the SINGLE STEP CLOCK momentary switch. The DATA DISPLAY-8 LED should turn ON.

ADDRESS SYNC COMPARATOR

Select the Am9324 Comparator and install it in the U40 position. The ADDRESS SYNC comparator is used to generate a synchronization pulse whenever the Am2909 Y address outputs match the address selected by the MEMORY ADDRESS switches of the kit.

Assembled	Location	Device Installed
☐	U40	Am9324

COMPLETING THE KIT

In order to complete the kit, the following integrated circuits are installed.

Assembled	Location	Device Installed
☐	U32	Am25LS253
☐	U33	Am25LS253
☐	U30	Am9309
☐	U34	Am2901

Testing these last devices in the kit assembly is more difficult and requires use of the exercises described in Section V of this manual. Please complete the testing as required by turning to Section V and beginning the exercises as described. The Am2901 can be tested by following some of the initial loading procedures and reading that the correct data is loaded into the internal memory in the Am2901. Likewise, the Am25LS253's are tested by performing some of the simple rotate functions described in Section V.

SUMMARY

The Am2900 Evaluation and Learning Kit is now fully assembled. If the assembly procedure described in Section IV has been followed, the user should recognize that the major portion of the kit has been tested. As with any such device, exhaustive testing has not been performed. However, it has been our experience at Advanced Micro Devices that if the kit is assembled and tested in the fashion described in Section IV, no functional problems will be encountered. Thus, as the user begins the exercises in Section V or especially as the user begins exercises of his own generation, he should be especially wary of any apparent malfunctions. These malfunctions are most likely operator error rather than faulty devices in the kit.

As the user gains more experience with the Am2900 Kit, he will find that he can immediately identify the reason that the expected answer does not appear as he anticipated. However, during the initial stages of becoming familiar with the kit, the user will often find the display does not give the answer he expected. It has been our experience at Advanced Micro Devices that almost always the ultimate solution to these problems is "operator error". Thus, when the expected answer does not appear at the LED display, the user should re-evaluate the anticipated result in order to determine the error in his microprogramming (that's where the learning comes in -- even the most competent microprogrammer will blow it now and then!).

SECTION V

PARTS LIST

INTEGRATED CIRCUITS

Location	Device	Description
U1	Am25LS138	One-of-eight decoder/demultiplexer
U2	Am27S03	16-word by 4-bit RAM
U3	Am27S03	16-word by 4-bit RAM
U4	Am27S03	16-word by 4-bit RAM
U5	Am27S03	16-word by 4-bit RAM
U6	Am27S03	16-word by 4-bit RAM
U7	Am27S03	16-word by 4-bit RAM
U8	Am27S03	16-word by 4-bit RAM
U9	Am27S03	16-word by 4-bit RAM
U10	Am2909	Bipolar microprogram sequencer
U11	Am9314	Four-Bit latch
U12	Am25LS138	One-of-eight decoder/demultiplexer
U13	Am2918	Four-bit register
U14	Am2918	Four-bit register
U15	Am2918	Four-bit register
U16	Am2918	Four-bit register
U17	Am2918	Four-bit register
U18	Am2918	Four-bit register
U19	Am2918	Four-bit register
U20	Am2918	Four-bit register
U21	Am25LS157	Quad two-input multiplexer; non-inverting
U22	Am25LS151	Eight-input multiplexer
U23	Am25LS151	Eight-input multiplexer
U24	Am25LS151	Eight-input multiplexer
U25	Am25LS151	Eight-input multiplexer
U26	Am25LS158	Quad two-input multiplexer;inverting
U27	Am29751	32-word by 8-bit PROM; three state
U28	Am74S157	Quad two-input multiplexer; non-inverting
U29	Am9309	Dual four-input multiplexer
U30	Am9309	Dual four-input multiplexer
U31	Am25LS08	Four-bit register with common clock enable
U32	Am25LS253	Three-state dual four-input multiplexer
U33	Am25LS253	Three-state dual four-input multiplexer

Location	Device	Description
U34	Am2901	Four-bit bipolar microprocessor slice
U35	Am2907	Quad open collector bus transceiver with three-state receiver and parity outputs
U36	Am25LS151	Eight-input multiplexer
U37	Am25LS151	Eight-input multiplexer
U38	Am25LS151	Eight-input multiplexer
U39	Am25LS151	Eight-input multiplexer
U40	Am9324	Five-bit Comparator

NOTE: Package type designations (DM, DC, PC) at the end of the part number have been intentionally omitted.

RESISTORS

Location	Value	Wattage	Part Number
R1	2200Ω	1/4 Watt	RCR07G222JM
R2	2200Ω	1/4 Watt	RCR07G222JM
R3	2200Ω	1/4 Watt	RCR07G222JM
R4	2200Ω	1/4 Watt	RCR07G222JM
R5	2200Ω	1/4 Watt	RCR07G222JM
R6	2200Ω	1/4 Watt	RCR07G222JM
R7	2200Ω	1/4 Watt	RCR07G222JM
R8	2200Ω	1/4 Watt	RCR07G222JM
R9	51Ω	1/2 Watt	RCR20G510JM
R10	1000Ω	1/4 Watt	RCR07G102JM
R11	51Ω	1 Watt	RCR32G510JM
R12	51Ω	1 Watt	RCR32G510JM
R13	51Ω	1 Watt	RCR32G510JM
R14	51Ω	1 Watt	RCR32G510JM
R15	330Ω	1/4 Watt	RCR07G331JM
R16	330Ω	1/4 Watt	RCR07G331JM
R17	330Ω	1/4 Watt	RCR07G331JM
R18	330Ω	1/4 Watt	RCR07G331JM
R19	330Ω	1/4 Watt	RCR07G331JM
R20	330Ω	1/4 Watt	RCR07G331JM
R21	330Ω	1/4 Watt	RCR07G331JM
R22	330Ω	1/4 Watt	RCR07G331JM
R23	330Ω	1/4 Watt	RCR07G331JM
R24	330Ω	1/4 Watt	RCR07G331JM
R25	330Ω	1/4 Watt	RCR07G331JM
R26	330Ω	1/4 Watt	RCR07G331JM

CAPACITORS

Location	Value	Polarized	Part Number
C1	0.022µf	No	5021EM50RD223M
C2	0.022µf	No	5021EM50RD223M
C3	0.022µf	No	5021EM50RD223M
C4	0.022µf	No	5021EM50RD223M
C5	0.022µf	No	5021EM50RD223M
C6	0.022µf	No	5021EM50RD223M
C7	60µf	Yes	500D606G102CB7
C8	0.022µf	No	5021EM50RD223M
C9	0.022µf	No	5021EM50RD223M
C10	0.022µf	No	5021EM50RD223M
C11	0.022µf	No	5021EM50RD223M
C12	0.022µf	No	5021EM50RD223M
C13	60µf	Yes	500D606G102CB7
C14	60µf	Yes	500D606G102CB7
C15	0.022µf	No	5021EM50RD223M
C16	0.022µf	No	5021EM50RD223M
C17	0.022µf	No	5021EM50RD223M

SWITCHES

Location	Type	Part Number
S1	DPDT Toggle	CTS-022
S2	DPDT Toggle	CTS-022
S3	DPDT Toggle	CTS-022
S4	DPDT Toggle	CTS-022
S5	DPDT Toggle	CTS-022
S6	DPDT Toggle	CTS-022
S7	DPDT Toggle	CTS-022
S8	DPDT Toggle	CTS-022
S9	DPDT Toggle	CTS-022
S10	DPDT Toggle	CTS-022
S11	DPDT Toggle	CTS-022
S12	DPDT Toggle	CTS-022
S13	DPDT Toggle	CTS-022
S14	SPDT Momentary	MDL-106F-80
S15	SPDT Momentary	MDL-106F-80

LIGHT EMITTING DIODES (LED's)

Location	Part Number
Data Display 8	HP5082-4655
Data Display 4	HP5082-4655
Data Display 2	HP5082-4655
Data Display 1	HP5082-4655
Pipeline Register 8	HP5082-4655
Pipeline Register 4	HP5082-4655
Pipeline Register 2	HP5082-4655
Pipeline Register 1	HP5082-4655
Microword Memory 8	HP5082-4655
Microword Memory 4	HP5082-4655
Microword Memory 2	HP5082-4655
Microword Memory 1	HP5082-4655

MECHANICAL COMPONENTS

Quantity	Part Number	Description
1	Am2900EKL	Printed Circuit Board
6	HHS 2181	Rubber Feet
6	HHS 1373	#4 Machine Screw
6	HHS 1168	#4 Nuts
7	1457-2	Turrett Terminals
1	-	Wooden Toothpick (tool)
1	AMP 583529-1	16-pin Socket

SECTION VI

PROGRAMMING EXERCISES

INTRODUCTION

The Am2900 Evaluation and Learning Kit is intended to teach the basics of microprogramming to the hardware design engineer. The exercises are geared to the objectives of demonstrating the uses of the Am2901 and Am2909 and to involve the user in the microprogramming of these devices. In addition, it is intended to allow evaluation of the Am2901 and Am2909 in an application environment. As such, the learning exercises are divided into two sections. These include static exercises and dynamic exercises. The static exercises are performed using the toggle switches and momentary switches available on the printed circuit board. The user clocks the system by hand observing the various states of the machine at each point. The dynamic tests are set up using the switches on the printed circuit board but are evaluated by using an external pulse generator and oscilloscope. This allows the Am2901 and Am2909 to be tested under operating conditions. In this fashion, the student can learn microprogramming techniques in the SINGLE STEP CLOCK mode and can evaluate the components in the PULSE GENERATOR mode, bearing in mind that the kit has not been designed as a small computer. However, once the user understands the Am2901, Am2909 and the principles of microprogramming, then the design of a computer becomes obvious.

The static test exercises are divided into three basic groups. The first group is intended to familiarize the engineer with the Am2901 capabilities. The second group is intended to familiarize the engineer with the Am2909 functions. The third group of exercises that can be generated appear almost infinite. The exercises presented here are only representative and the student is encouraged to write numerous additional exercises to evaluate any specific parameter or feature in which he is interested.

In the dynamic testing mode, several exercises are presented such that the student can evaluate typical switching characteristics. Many of the specified parameters of the Am2901, Am2907, Am2909 and Am2918 can indeed be measured using the Am2900 Kit. It is not possible to measure some of the parameters due to the complex nature of the testing required.

USING THE EXERCISES

The microprogramming exercises are contained on the Am2900 Kit microprogramming worksheets following this discussion. These worksheets show the specific "logic one" and "logic zero" codes that must be entered into the microprogram memory in order to execute the various microinstructions so as to perform a function (HIGH = logic one). The results of the execution of these microinstructions is described in the accompanying text.

The Am2900 Kit programming worksheet is divided into two main parts. The top half of the worksheet shows a map of the entire 16-word by 32-bit microprogram memory. The logic ones and logic zeros associated with the fields of each microprogram memory word can be entered into the worksheet. Any blank space on the worksheet indicates a "don't care" condition. The bottom half of the worksheet shows a functional block diagram representation of the Am2900 Kit. This block diagram is intended to allow the user to sketch the path of a microinstruction so he may more easily understand the actual event being executed. The use of these two sections of the worksheet is demonstrated in the exercises.

PROGRAMMING THE MICROPROGRAM MEMORY

The recommended procedure for using the exercises as described with the Am2900 Kit is to load all the required data into the microprogram memory. For example, all six words of Exercise 1 should be loaded into the microprogram memory before executing these instructions. The procedure to be used in loading the microprogram memory is as follows:

1. Connect the 5V power supply to the Am2900 Kit and apply power.
2. Set the RUN/LOAD SELECT switch (S12) to the LOAD position.
3. Set the MEMORY ADDRESS switches (S8-S11) to the decimal zero position.
4. Set the RAM & MUX SELECT switches (S1-S3) to the decimal zero position.
5. Set the MEMORY DATA switches (S4-S7) to the required bit position.
6. Depress the MEMORY LOAD switch (S14) to enter the data.
7. View the MICROWORD MEMORY LED's to be sure the correct word has been written into the microprogram memory.
8. Repeat steps 3-7 for each decimal position of the RAM & MUX SELECT switches until all eight fields (decimal 0-7) have been loaded.
9. Repeat steps 3-8 for each microprogram memory address word until the entire 16 words (word 0-15) of the microprogram memory have been loaded.

Needless to say, any blank entries ("don't care" conditions) may be omitted. When this procedure is followed, all possible 128 four-bit fields in the microprogram memory are easily loaded.

The above description has sequentially loaded the 8 four-bit fields of the microprogram word, one word at a time. The user may wish to load all 16 words of each field sequentially. That is, the RAM & MUX SELECT switches are set to decimal 0 and each MEMORY ADDRESS is loaded with the appropriate word for the entire 16 words of the memory. Advanced Micro Devices' Applications Engineers have found both techniques to be convenient depending on the particular exercise.

Once the microprogram memory has been loaded, the exercise should be executed as described in the accompanying text with the programming worksheet.

CONNECTING THE PULSE GENERATOR

The Am2900 Evaluation and Learning Kit is designed such that it can be easily driven from a pulse generator. Two turret terminals are provided near resistor R9 so that a coaxial cable can be soldered to the board. One of the turret terminals is marked "GND" and connects directly to the ground bus of the system. A 51 ohm resistor is connected directly between the two turret terminals. This provides a termination for the coaxial cable at the U29 IC so that the cable is properly matched.

The printed circuit board has been designed such that it will accept a printed circuit type BNC connector. If the user desires, he may purchase a King's KC-79-153 connector for installation directly beneath the pulse generator turret terminals. This will provide a second method for connecting the pulse generator to the Am2900 Evaluation and Learning Kit. In some laboratory environments, this may be more convenient than directly connecting a coaxial cable to the printed circuit board.

Programming Exercise #1 - Loading the Am2901 Memory

Many of the operations performed with the Am2901 Microprocessor will involve the use of the 16-word, two-address RAM. These 16 RAM locations may be employed at the option of the user as program counters, accumulators, index registers, scratch pad memory, stack pointers, non-user registers, and so forth. Application of the RAM can be determined by the system architecture as well as the microprograms. The purpose of this exercise is to demonstrate one technique for loading the 16 registers from the incoming data input. The exercise is written so as to load a selected word in the RAM memory and then advance to the next line of microprogram code. At this word, the contents of the Am2901 RAM is read out so that it can be examined at the output of the Am2901.

In this exercise, microprogram word zero contains a microinstruction that will load Register "0" of the Am2901 with the value binary two. The microinstruction at word one in the microprogram memory reads the contents of register zero to the Am2901 output via the B data output of the 16 x 4 RAM. Microinstructions 2 and 3 perform the same function on RAM Register "1" while microinstructions 4 and 5 perform the same function on RAM Register "9". Obviously, the user can load and examine all 16 registers in the Am2901 RAM using a similar pair of microinstructions. Incidentally, microinstructions 0 and 1 could be used to perform this function on all 16 registers in the Am2901 by simply re-loading the contents of the microprogram memory "D" and "B" fields and then executing these two instructions; then reloading and executing again.

The actual execution of this exercise is as described below. Leave the RUN/LOAD switch in the LOAD position and set the MEMORY ADDRESS switches to decimal 0. First, depress the SINGLE STEP CLOCK momentary switch so as to enter the contents of the microprogram memory into the pipeline register. Next, advance the MEMORY ADDRESS switches to the decimal 1 position and again depress the SINGLE STEP CLOCK momentary switch. This has the effect of executing the contents of the pipeline register which is currently word zero and entering the contents of microprogram memory word 1 into the pipeline register. The result is that binary two is loaded into Register 0 and the pipeline register contains an instruction that is currently reading the contents of Register 0 to the Am2901 outputs. The user can view the Am2901 outputs on the DATA DISPLAY LED's by setting the RAM & MUX SELECT switches to the decimal 1 position. When this function is performed, the DATA DISPLAY reads binary two (0010).

The next pair of microinstructions which load Register 1 can be executed in a similar fashion. That is, set the MEMORY ADDRESS select switches to the decimal 2 position and depress the SINGLE STEP CLOCK momentary switch. This loads the contents of microprogram memory word 2 into the pipeline register. Next, advance the MEMORY ADDRESS switches to the decimal 3 position and again depress the SINGLE STEP CLOCK momentary switch. This executes the contents of the pipeline register and enters the "read" microinstruction into

the pipeline register. Thus, the user now views the word that has been loaded into Register 1 of the Am2901 RAM. Microinstruction words 4 and 5 are executed in a similar fashion.

The purpose of this exercise is to demonstrate the simple loading of data into the Am2901 memory. The loading of the Q register can be accomplished in exactly the same fashion with only the destination select field of the microinstruction control word being changed. In this example, each "load" instruction has been followed by a "read" instruction so the contents of the Am2901 memory can be examined. In normal operation, only the load operation is performed and the machine moves on to the next task at hand.

Am2900 KIT PROGRAMMING WORK SHEET

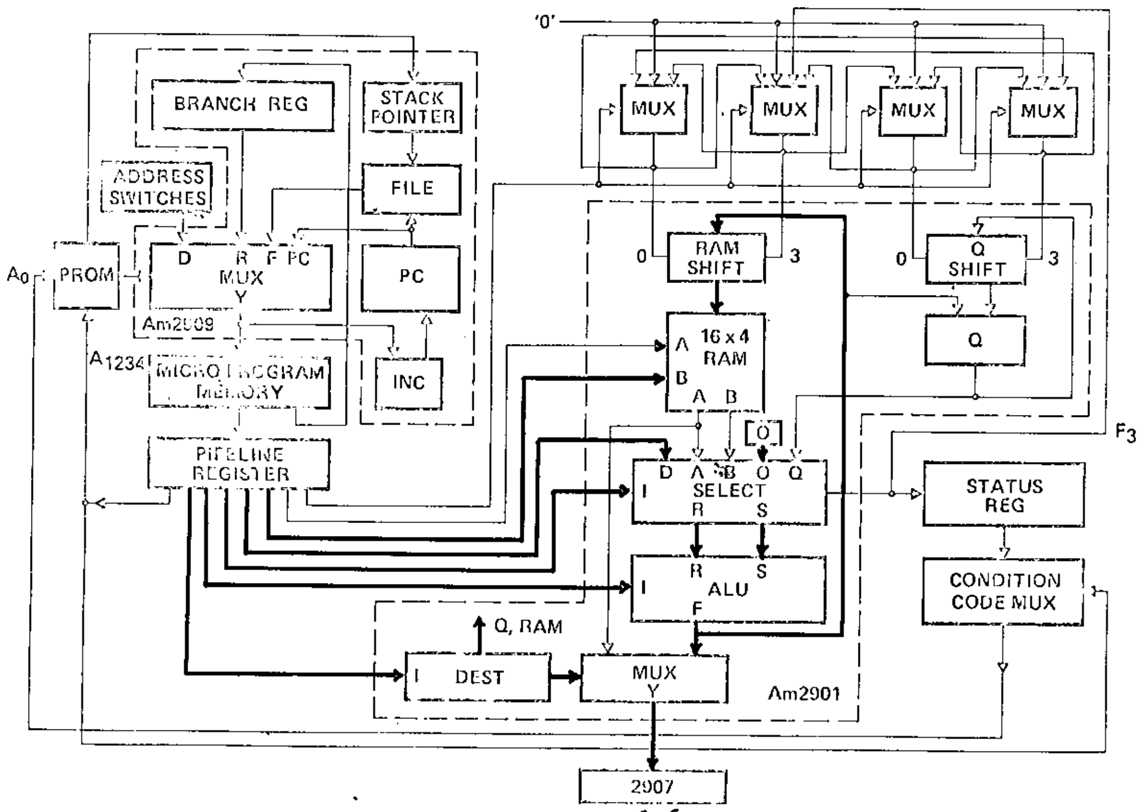
EXERCISE NUMBER: 1 NAME: LOADING THE AM2901 RAM

MICROPROGRAM MEMORY

RAM & MUX	7	6		5		4		3	2	1	0	NOTES	
	BRANCH ADDRESS	NEXT INSTRUCTION CONTROL	MUX1	DEST. SELECT	MUX0	SOURCE SELECT	Cn	ALU	'A'	'B'	'D'	NEXT ADDRESS CONTROL	DATA CONTROL
MEMORY ADDRESS	BR3 BR2 BR1 BR0	P3 P2 P1 P0		I8 I7 I6		I2 I1 I0		I5 I4 I3	A3 A2 A1 A0	B3 B2 B1 B0	D3 D2 D1 D0		
0				011		111		011		0000	0010		LOAD R0
1				001		011		011		0000			READ R0
2				011		111		011		0001	0100		LOAD R1
3				001		011		011		0001			READ R1
4				011		111		011		1001	0101		LOAD R9
5				001		011		011		1001			READ R9
6													
7													
8													
9													
10													
11													
12													
13													
14													
15													

BLANK - DON'T CARE

INSTRUCTION LOAD R_i



Programming Exercise #2 - Rotate Functions in the Am2901

The Am2900 Evaluation Kit is designed to provide four different shift matrix multiplexer control functions as shown on page 3-9 of the operational description. These four functions are enter zeros, rotate, double length rotate, and double length arithmetic rotate. This shift function can be performed in either the shift-up mode or shift-down mode. The purpose of this exercise is to demonstrate the use of the single length rotate up and down, as well as the double length rotate up and down.

First, the microprogram memory should be loaded with the various data as shown in the accompanying worksheet. Microprogram word 0 is used to load a binary one in the second bit position of Register 0 in the Am2901 RAM. The microinstruction at word 1 performs a rotate up function on this bit. Microprogram memory word 2 performs the rotate down function; word 3 is the double length rotate up function and word 4 is the double length rotate down function. Word 5 is shown to demonstrate the "no op" function.

Exercise 2 should be performed in the following manner. Keep the RUN/LOAD switch in the LOAD position. Set the MEMORY ADDRESS switches to the decimal 0 position and depress the SINGLE STEP CLOCK momentary switch. This will enter the contents of microprogram memory word 0 into the pipeline register. Next, advance the MEMORY ADDRESS switches to the decimal 1 positon. Depress the SINGLE STEP CLOCK momentary switch. If the RAM & MUX SELECT switches are set to the decimal 1 position, the output of the Am2901 Microprocessor will be viewed on the DATA DISPLAY LED's. The current contents of the DATA DISPLAY should be binary two (0010). Now, if the MEMORY ADDRESS switches are left at the decimal 1 positon and the SINGLE STEP CLOCK momentary switch is repeatedly depressed, the bit as displayed on the DATA DISPLAY will be rotated in the upward position. That is, the bit will move in the 2, 4, 8, 1, 2, etc. pattern (0010, 0100, 1000, 0001, 0010, etc.). As many times as the SINGLE STEP CLOCK momentary switch is depressed, the single length rotate up function will be executed.

Next, set the MEMORY ADDRESS switches to the decimal 0 position and depress the SINGLE STEP CLOCK momentary switch. This will again load the contents of microprogram memory word 0 into the pipeline register. Then, set the MEMORY ADDRESS switches to the decimal 2 position and depress the SINGLE STEP CLOCK momentary switch. This will execute the load two microinstruction and enter microprogram memory word 2 into the pipeline register. Now, with the RAM & MUX SELECT switches in the decimal 2 position, the DATA DISPLAY LED's will again show binary two (0010). Keeping the MEMORY ADDRESS switches in the decimal 2 position, each depression of the SINGLE STEP CLOCK momentary switch will result in the execution of a rotate down microinstruction. The pattern on the DATA DISPLAY will follow a 2, 1, 8, 4, 2, etc. sequence as the SINGLE STEP CLOCK momentary switch is depressed.

The double length rotate up and double length rotate down instructions are executed in a similar fashion. First, execute the word at decimal 0 position, then execute the word at the decimal 3 position and then execute the word at the decimal 4 or decimal 5 position. The word at the decimal 3 position simply performs a clear Q function so that the contents of the Q register intially is all zeros. The double length rotate up pattern will now be 2, 4, 8, 0, 0, 0, 0, 1, 2, 4, 8, 0, etc. Likewide, the double length rotate down pattern will be 2, 1, 0, 0, 0, 0, 8, 4, 2, 1, 0, etc.

The "no-operation" function at microprogram memory word 6, if selected, simply demonstrates that the Am2901 can be clocked with no write operation being performed. During any of the above exercises, the MEMORY ADDRESS switches can be changed to the decimal 6 position and the SINGLE STEP CLOCK momemtary switch depressed. The result is that the no-operation function will be entered into the pipeline register and then executed. It should be remembered that when the MEMORY ADDRESS switches are changed to the decimal 6 position, the pipeline register will currently contain a rotate instruction. This instruction will be executed as the contents of the microprogram memory (no-operation) are loaded into the pipeline register. Thus, one additional execution of the rotate function takes place before the no operation instruction is executed.

The purpose of this exercise is to demonstrate how an external multiplexer can be used to perform single length and double length rotates in either direction. The user is encouraged to write instructions that perform shifting of zeros into the word and double length arithmetic shifts. From this example, it should be apparent to the user that a multiplexer scheme can be designed to include the carry flip-flop such that single length and double length rotate with or without carry could be implemented. Likewise, shifting ones or shifting zeros into the Am2901 would be possible.

Am2900 KIT PROGRAMMING WORK SHEET

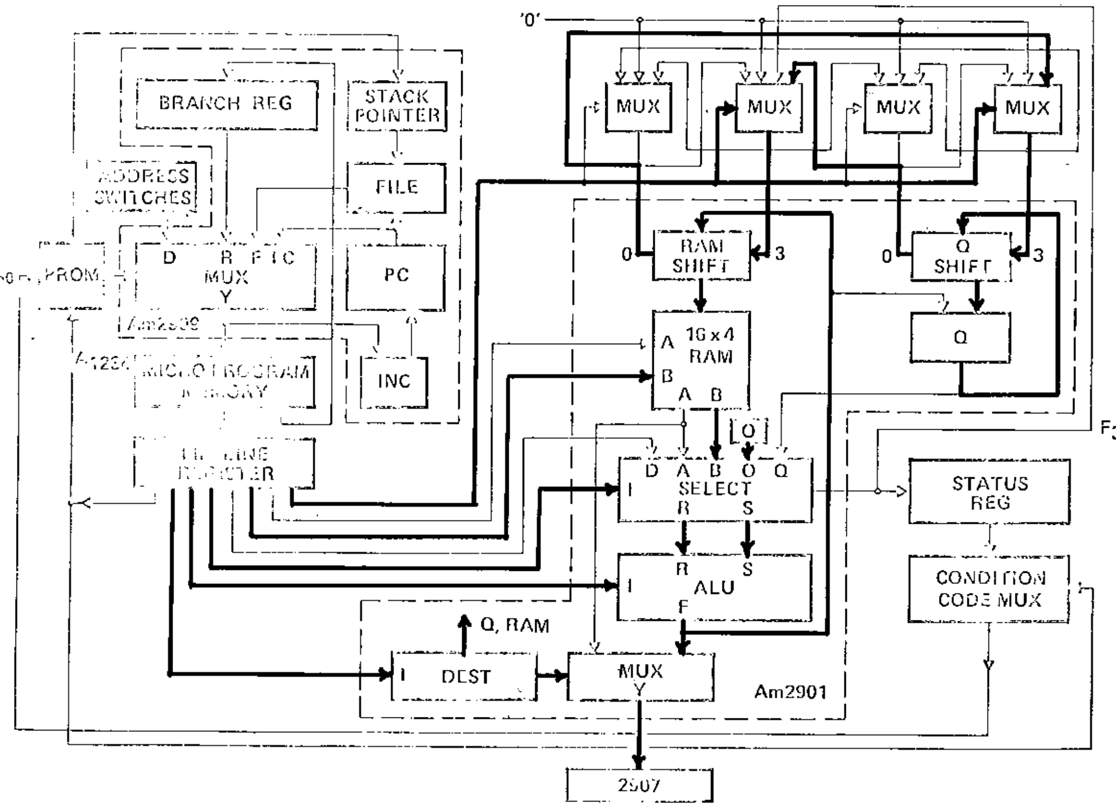
EXERCISE NUMBER: 2 NAME: ROTATE EXERCISES

MICROPROGRAM MEMORY

RAM & MUX / MEMORY ADDRESS	7 BRANCH ADDRESS BR_3 BR_2 BR_1 BR_0	6 NEXT INSTRUCTION CONTROL P_3 P_2 P_1 P_0	5 MUX_1	5 DEST SELECT I_8 I_7 I_6	4 MUX_0	4 SOURCE SELECT I_2 I_1 I_0	3 C_n	3 ALU I_5 I_4 I_3	2 'A' A_3 A_2 A_1 A_0	1 'B' B_3 B_2 B_1 B_0	0 'D' D_3 D_2 D_1 D_0	NOTES NEXT ADDRESS CONTROL	NOTES DATA CONTROL
0				011		111		011		0000	0010		LOAD R_0
1			0	111	1	011		011		0000			Rot Up
2			0	101	1	011		011		0000			Rot DN
3				000		010		100					CLR Q
4			1	110	0	011		011		0000			DBL Rot Up
5			1	100	0	011		011		0000			DBL Rot DN
6				001		011		011					No Op
7													
8													
9													
10													
11													
12													
13													
14													
15													

BLANK = DON'T CARE

INSTRUCTION Double Length Rotate Down



Programing Exercise #3 - Am2901 Arithmetic Operation

The purpose of this exercise is to demonstrate a small set of the arithmetic capability of the Am2901 Microprocessor. Four different instruction types are demonstrated on the worksheet for this exercise. The first three types are demonstrated in microprogram memory words 0, 1, and 2 while the fourth type is demonstrated using three microinstructions at memory locations 7, 8, and 9. Again, all data shown in the worksheet should be entered into the microprogram memory.

This exercise is executed in the following manner. Set the RUN/LOAD switch to the LOAD position and set the MEMORY ADDRESS switches to the decimal 0 position. If the RAM & MUX SELECT switches are set to the decimal 1 position, the output of the Am2901 Microprocessor can be viewed on the DATA DISPLAY LED's. Each time the SINGLE STEP CLOCK momentary switch is depressed, the current DATA DISPLAY will be incremented by one.

This is accomplished by using the B and O source operands, adding in the ALU with the carry-in set to a one, and writing the results into the Register 0 of the RAM.

If the MEMORY ADDRESS switches are placed in the decimal 1 position, each time the SINGLE STEP CLOCK momentary switch is deprssed, the contents of Register 0 will be decremented. Thus, the DATA DISPLAY will show this decrement function on the contents of Register 0.

If the MEMORY ADDRESS switches are placed at the decimal 2 position, each depression of the SINGLE STEP CLOCK momentary switch will result in the contents of Register 0 being increased by three. Needless to say, an overflow will occur every six clock cycles and the contents of the register will "end around" from a positive value to a negative value. This microinstruction demonstrates adding a data bus input value on the D inputs to the contents of a register.

Microinstruction words 7 and 8 perform the required set-up to demonstrate a register-to-register add operation at microinstruction word 9. This sequence of microinstructions is executed in the following manner. Set the MEMORY ADDRESS switches to the decimal 7 position and depress the SINGLE STEP CLOCK momentary switch. Next, set the MEMORY ADDRESS switches to the decimal 8 position and again depress the SINGLE STEP CLOCK momentary switch. Now, set the MEMORY ADDRESS switches to the decimal 9 position and again depress the SINGLE STEP CLOCK momentary switch. Thus far in the procedure, we have executed microinstruction 8 which sets register 1 to the value five. If we leave the MEMORY ADDRESS switches in the decimal 9 position and begin depressing the SINGLE STEP CLOCK momentary switch, the contents of Register 0 will be increased by the value five. This will be equivalent to a sequence of 0, 5, 10, 15, 4, 9, 14, 3, 8, etc., in a

"magnitude only" number system. This output is viewed on the DATA DISPLAY LED's when the RAM & MUX SELECT switches are in the decimal 1 position. This microinstruction represents a register-to-register arithmetic addition within the Am2901 Microprocessor.

Am2900 KIT PROGRAMMING WORK SHEET

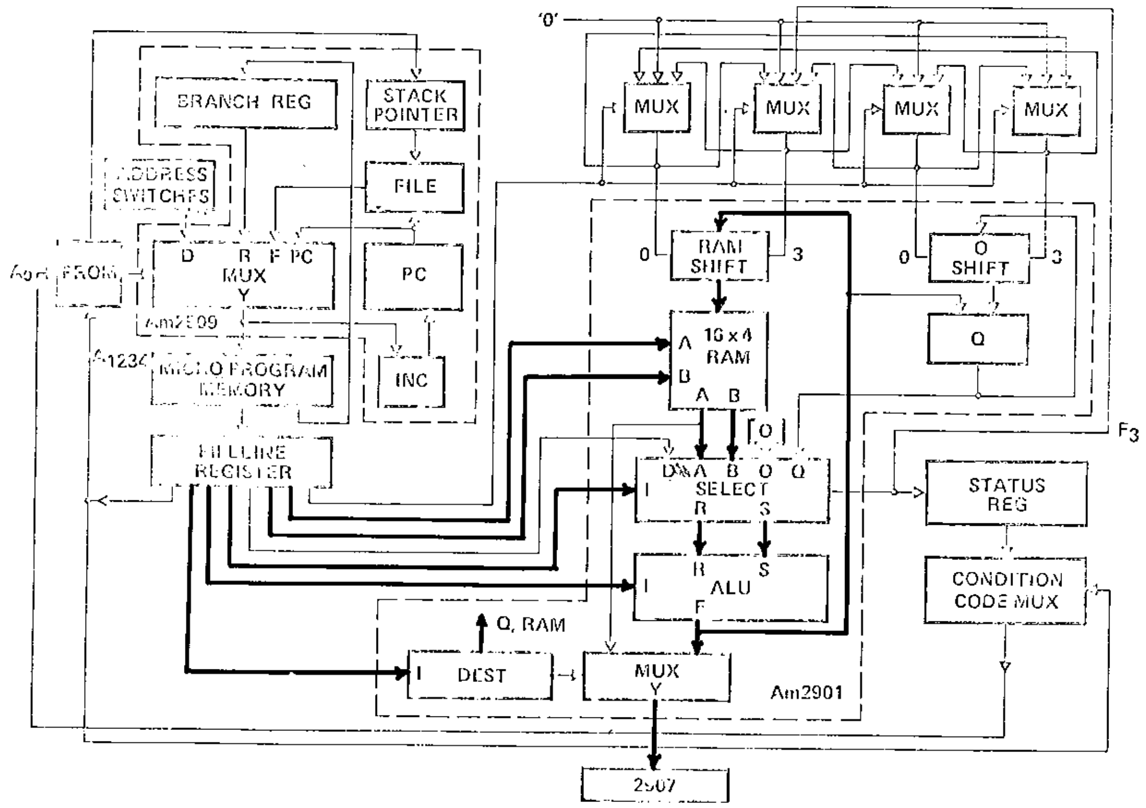
EXERCISE NUMBER: 3 NAME: AM2901 ARITHMETIC OPERATIONS

MICROPROGRAM MEMORY

RAM & MUX	7	6	5	5	4	4	3	3	2	1	0	NOTES	NOTES
	BRANCH ADDRESS	NEXT INSTRUCTION CONTROL	MUX1	DEST. SELECT	MUX0	SOURCE SELECT	C_n	ALU	'A'	'B'	'D'	NEXT ADDRESS CONTROL	DATA CONTROL
MEMORY ADDRESS	BR_3 BR_2 BR_1 BR_0	P_3 P_2 P_1 P_0		I_8 I_7 I_6		I_2 I_1 I_0		I_5 I_4 I_3	A_3 A_2 A_1 A_0	B_3 B_2 B_1 B_0	D_3 D_2 D_1 D_0		
0				011		011	1	000		0000			INC R_0
1				011		011	0	001		0000			DEC R_0
2				011		101	0	000	0000	0000	0011		R_0+3
3													
4													
5													
6													
7				011		011		100		0000			CLR R_0
8				011		111		011		0001	0101		$R_1 = 5$
9				011		001	0	000	0001	0000			$R_0 \leftarrow R_0 + R_1$
10													
11													
12													
13													
14													
15													

BLANK - DON'T CARE

INSTRUCTION $R_0 \leftarrow R_0 + R_1$



Programming Exercise #4 - Continue and Branch Next Instructions in the Am2909 Microprogram Sequencer

The next few exercises will examine various functions for the control of the next microinstruction address. Field 6 of the microprogram memory word contains a number of functions that are executed by the next address control PROM associated with the Am2909 Microprogram Sequencer. These functions are shown on page 3-5 of the discussion on the operation of the Am2900 Kit. Exercises 1, 2, and 3 demonstrated a few of the functions of the Am2901 Microprocessor. All of these microinstructions were executed with the RUN/LOAD switch in the LOAD position. In order to execute the microinstructions associated with the next address control of the Am2909 Microprogram Sequencer, it will be necessary to switch the RUN/LOAD switch to the RUN position. However, in order to load the microprogram memory, the RUN/LOAD switch must be in the LOAD position.

First, the data on the programming table of the exercise 4 worksheet should be loaded into the microprogram memory. This exercise demonstrates the CONTINUE (or EXECUTE) operation as well as the BRANCH (or JUMP) operation in next microinstruction select control. After the microprogram memory has been loaded, the MEMORY ADDRESS switches should be placed in the decimal 0 position and the SINGLE STEP CLOCK momentary switch depressed one time. This will load the contents of memory address zero into the pipeline register. This is necessary to initialize the pipeline register so that the instruction sequence in the Exercise 4 worksheet can be executed. If this step is not performed, the current contents of the pipeline register are not known and the next microprogram memory address will most likely not be as determined from the worksheet. Now, the RUN/LOAD switch should be changed to the RUN position. If the RAM & MUX SELECT switches are placed in the decimal 0 position, the output of the Am2909 Microprogram Sequencer can be viewed on the DATA DISPLAY LED's. This display represents the address of the next microinstruction. At this point in the execution of the exercise, the DATA DISPLAY should show decimal nine (1001). As the SINGLE STEP CLOCK momentary switch is depressed, the sequence on the DATA DISPLAY should be 9, 6, 10, 11, 12, 13, 14, 15, 3, 0, 9, 6, etc. Microinstruction at address 0, 3, 6, 9, and 15 are BRANCH instructions while microinstructions at addresses 10, 11, 12, 13, and 14 are CONTINUE instructions.

The purpose of this exercise, of course, is to demonstrate the simple CONTINUE and BRANCH instructions used throughout microprogram memory next address control.

Am2900 KIT PROGRAMMING WORK SHEET

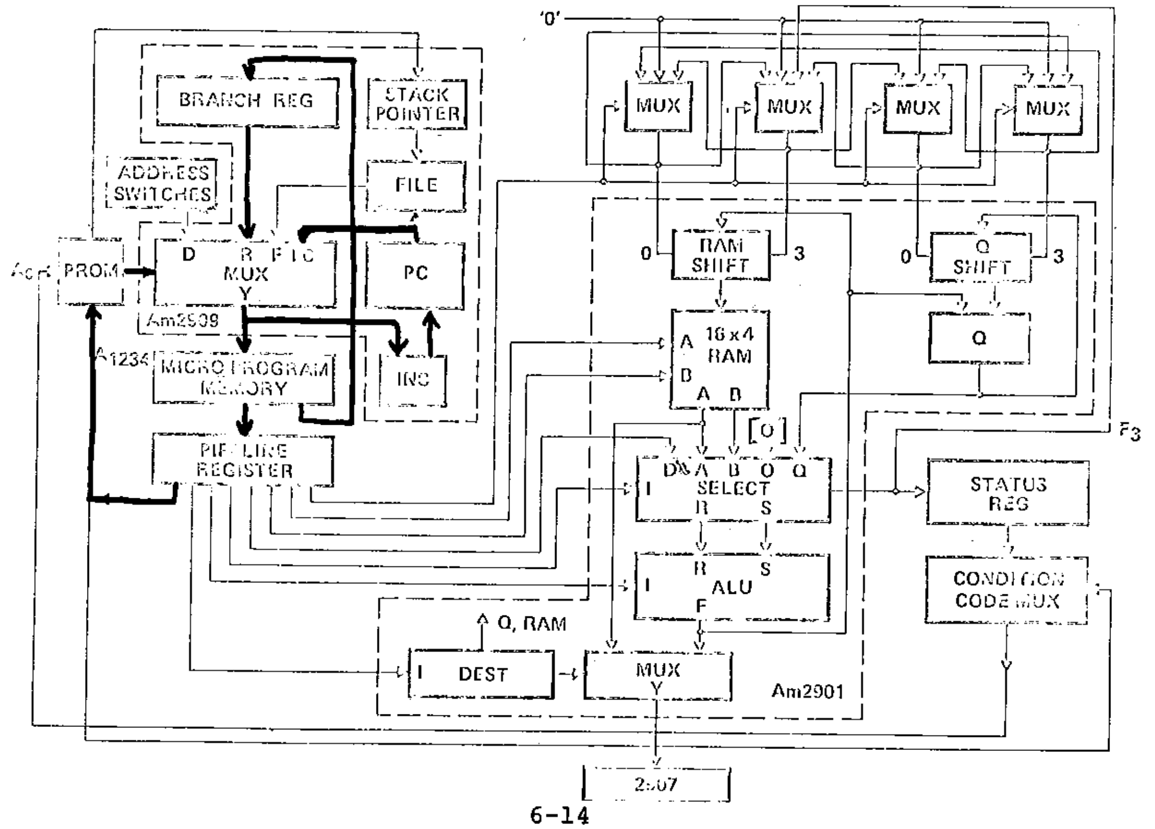
EXERCISE NUMBER: 4 NAME: CONTINUE AND BRANCH

MICROPROGRAM MEMORY

RAM & MUX / MEMORY ADDRESS	7 BRANCH ADDRESS $BR_3\,BR_2\,BR_1\,BR_0$	6 NEXT INSTRUCTION CONTROL $P_3\,P_2\,P_1\,P_0$	MUX_1	5 DEST. SELECT $I_8\,I_7\,I_6$	MUX_0	4 SOURCE SELECT $I_2\,I_1\,I_0$	C_n	3 ALU $I_5\,I_4\,I_3$	2 'A' $A_3\,A_2\,A_1\,A_0$	1 'B' $B_3\,B_2\,B_1\,B_0$	0 'D' $D_3\,D_2\,D_1\,D_0$	NOTES: NEXT ADDRESS CONTROL	NOTES: DATA CONTROL
0	1001	0001										BR 9	
1													
2													
3	0000	0001										BR 0	
4													
5													
6	1010	0001										BR 10	
7													
8													
9	0110	0001										BR 6	
10		0010										CONT	
11		0010										CONT	
12		0010										CONT	
13		0010										CONT	
14		0010										CONT	
15	0011	0001										BR3	

BLANK = DON'T CARE

INSTRUCTION SEQUENCE; 0, 9, 6, 10, 11, 12, 13, 14, 15 3, 0, ETC



Programming Exercise #5 - Looping in Microprogram Memory

The purpose of this exercise is simply to demonstrate the technique of looping within the microprogram memory. This particular demonstration does not make provision for braching out of the loop in any fashion. While this is not the normal case, it does provide the student with a view of what is involved in getting into and executing a loop. The loop is normally terminated by using one of two types of instructions. The first type of instruction would be test-end-of-loop and either repeat the loop if the condition is not true or continue out of the loop if the condition is true. When the "continue out of the loop" microinstruction is performed, a POP is executed to keep the stack maintained properly. The second technique for escaping a loop is to perform a conditional branch microinstruction somewhere within one of the loop microinstructions and when the test condition finally becomes true, a branch from the loop is made. Again, once out of the loop the first microinstruction should be a POP to perform the file maintenance.

Once the data on the worksheet associated with Exercise 5 has been loaded in the microprogram memory, the MEMORY ADDRESS select switches should be placed at decimal 0 and the SINGLE STEP CLOCK momentary switch depressed. Now, the RUN/LOAD select switch should be changed to the RUN position. If the RAM & MUX SELECT switches are placed in the decimal 0 position, the DATA DISPLAY LED's will view the Am2909 Microprogram Sequencer next address output. At this point, the DATA DISPLAY will show decimal 1. As the SINGLE STEP CLOCK momentary switch is depressed, the DATA DISPLAY LED's sequence will follow the pattern of 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 3, 4, 5, etc. Thus, the microinstruction 3 and proceeding through microinstruction 12, at which point a loop back to microinstruction 3 is performed. This loop is made possible by using the file reference next address microinstruction. A PUSH was performed at microword 2 so that the word currently on the stack is address 3. Thus, each time the file reference instruction is executed at microword 12, the next address from the Am2909 Microprogram Sequencer is microinstruction address 3.

Am2900 KIT PROGRAMMING WORK SHEET

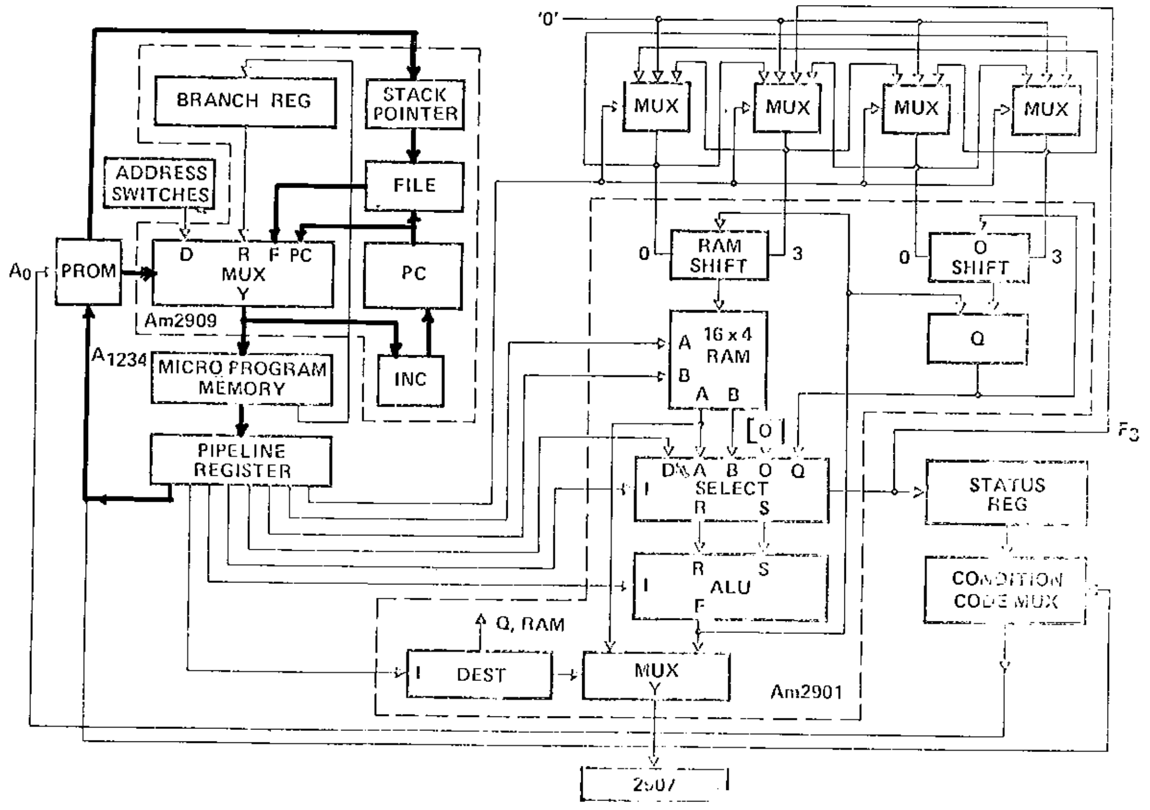
EXERCISE NUMBER: 5 NAME: LOOPING IN MICRO PROGRAM MEMORY

MICROPROGRAM MEMORY

RAM & MUX	7				6				5				4				3				2				1				0				NOTES	
	BRANCH ADDRESS				NEXT μINSTRUCTION CONTROL				MUX_1	DEST. SELECT			MUX_0	SOURCE SELECT			C_n	ALU			'A'				'B'				'D'				NEXT ADDRESS CONTROL	DATA CONTROL
MEMORY ADDRESS	BR_3	BR_2	BR_1	BR_0	P_3	P_2	P_1	P_0		I_8	I_7	I_6		I_2	I_1	I_0		I_5	I_4	I_3	A_3	A_2	A_1	A_0	B_3	B_2	B_1	B_0	D_3	D_2	D_1	D_0		
0					0	0	1	0																									CONT	
1					0	0	1	0																									CONT	
2					1	0	0	1																									PUSH	
3					0	0	1	0																									CONT	
4					0	0	1	0																									CONT	
5					0	0	1	0																									CONT	
6					0	0	1	0																									CONT	
7					0	0	1	0																									CONT	
8					0	0	1	0																									CONT	
9					0	0	1	0																									CONT	
10					0	0	1	0																									CONT	
11					0	0	1	0																									CONT	
12					0	1	1	1																									FILE REF	
13																																		
14																																		
15																																		

BLANK = DON'T CARE

INSTRUCTION SEQUENCE; 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 3, 4, ETC



Programming Exercise #6 - Executing a Subroutine in Microprogram Control

The purpose of this exercise is to demonstrate the technique of subroutining in microprogram memory. Microinstruction 3 executes a jump-to-subroutine at microinstruction 12. The subroutine at microinstruction 12 is three microinstructions in length covering the microprogram memory space between microinstruction 12 and microinstruction 14. The basic microinstruction sequence between microword 0 and microword 6 is simply a continue sequence between 0 and 3 as well as from 4 to 6. At microword 6, a branch to word 0 is performed.

This exercise is executed by first loading the microprogram memory with the data on the worksheet. Once the microprogram memory has been loaded, the MEMORY ADDRESS select switches should be placed in the decimal 0 position and the SINGLE STEP CLOCK momentary switch should be depressed. Now, the RUN/LOAD select switch should be placed to the RUN position and the RAM & MUX SELECT switches placed to the decimal 0 position. This alllows the DATA DISPLAY LED's to view the output of the Am2909 Microprogram Sequencer. The current display should be decimal value 1. As the SINGLE STEP CLOCK momentary switch is deprssed, the DATA DISPLAY should execute the sequence 1, 2, 3, 12, 13, 14, 4, 5, 6, 0, 1, 2, 3, 12, etc.

Am2900 KIT PROGRAMMING WORK SHEET

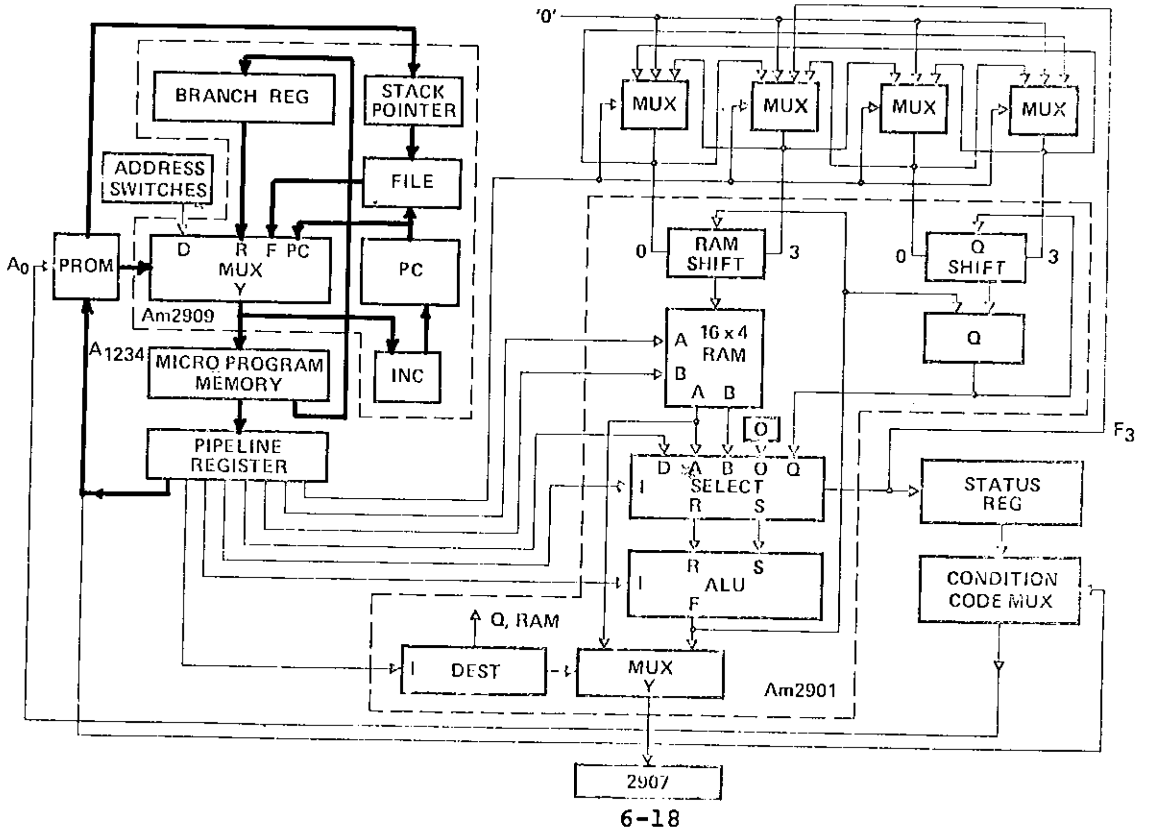
EXERCISE NUMBER: 6 NAME: JUMP TO ONE SUBROUTINE

MICROPROGRAM MEMORY

RAM & MUX / MEMORY ADDRESS	7 BRANCH ADDRESS $BR_3\ BR_2\ BR_1\ BR_0$	6 NEXT μINSTRUCTION CONTROL $P_3\ P_2\ P_1\ P_0$	MUX_1	5 DEST. SELECT $I_8\ I_7\ I_6$	MUX_0	4 SOURCE SELECT $I_2\ I_1\ I_0$	C_n	3 ALU $I_5\ I_4\ I_3$	2 'A' $A_3\ A_2\ A_1\ A_0$	1 'B' $B_3\ B_2\ B_1\ B_0$	0 'D' $D_3\ D_2\ D_1\ D_0$	NOTES NEXT ADDRESS CONTROL	NOTES DATA CONTROL
0		0010										CONT	
1		0010										CONT	
2		0010										CONT	
3	1100	0101										JSB 12	
4		0010										CONT	
5		0010										CONT	
6	0000	0001										BR 0	
7													
8													
9													
10													
11													
12		0010										CONT	
13		0010										CONT	
14		0110										RTS	
15													

BLANK DON'T CARE

INSTRUCTION ____________



Programming Exercise #7 - Nesting Subroutines

This exercise demonstrates the technique of nesting subroutines in microprogram memory. The subroutines are nested four levels deep in this example. The main microprogram resides at microinstructions 13, 14, and 15. The microinstruction at word 14 is a jump-to-subroutine at address 0. This is shown on the flow diagram below. Once at subroutine 0, we see the first instruction is a jump-to-subroutine at microinstruction 12. The subroutine at microinstruction 12 is a single microinstruction subroutine resulting in a return-from-subroutine. The microinstruction at address 1 is another jump-to-subroutine at microprogram address 6. The subroutine beginning at microprogram address 6 again executes a jump-to-subroutine at address 12. Again, the single microinstruction subroutine is executed and the microprogram control returns to address 7 where another jump-to-subroutine is found. Here, the jump-to-subroutine takes the next microinstruction from word 3. The microinstruction at word 3 again executes a jump-to-subroutine at word 12. The single microinstruction subroutine at location 12 is now executed and the microprogram control returns to address 4. The microprogram instruction at word 4 is another jump-to-subroutine at microprogram memory word 9. Microprogram memory word 9 is a continue instruction and microprogram memory word 10 is a return-from-subroutine instruction. The flow is such that subroutine 9 returns to subroutine 3 which returns to subroutine 6 which returns to subroutine 0 which returns to the main program.

From this example, several observations can be made about subroutining in microprogram control. First, we have nested subroutines up to four levels deep using the Am2909 stack. Second, the stack has maintained the correct return linkage throughout the various subroutine calls and returns. Third, the subroutine at microinstruction 12 has been used at more than one level of subroutining.

The purpose of this exercise has been to demonstrate nesting of four levels of subroutining. Note that each level of subroutining required one return address location in the PUSH/POP stack. Recalling the discussion on looping in Exercise 5, a PUSH onto the stack was performed to supply the reference address for the loop. Thus, it should be understood that up to four levels of loops and subroutines can be intermixed in any fashion. One word of the stack is used for each reference address required. For example, a subroutine might contain a loop which contains another subroutine which contains another loop. This would utilize the four levels of stack depth within the Am2909 Microprogram Sequencer.

Am2900 KIT PROGRAMMING WORK SHEET

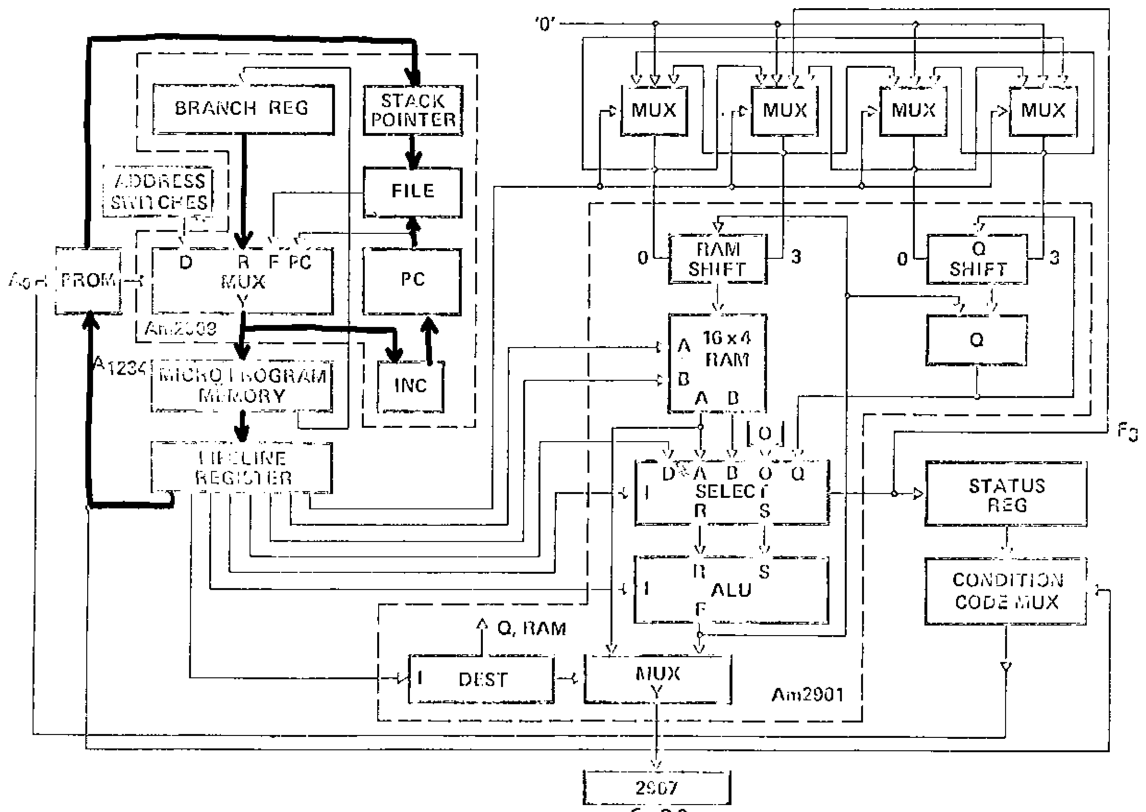
EXERCISE NUMBER: 7 NAME: NESTING SUBROUTINES

MICROPROGRAM MEMORY

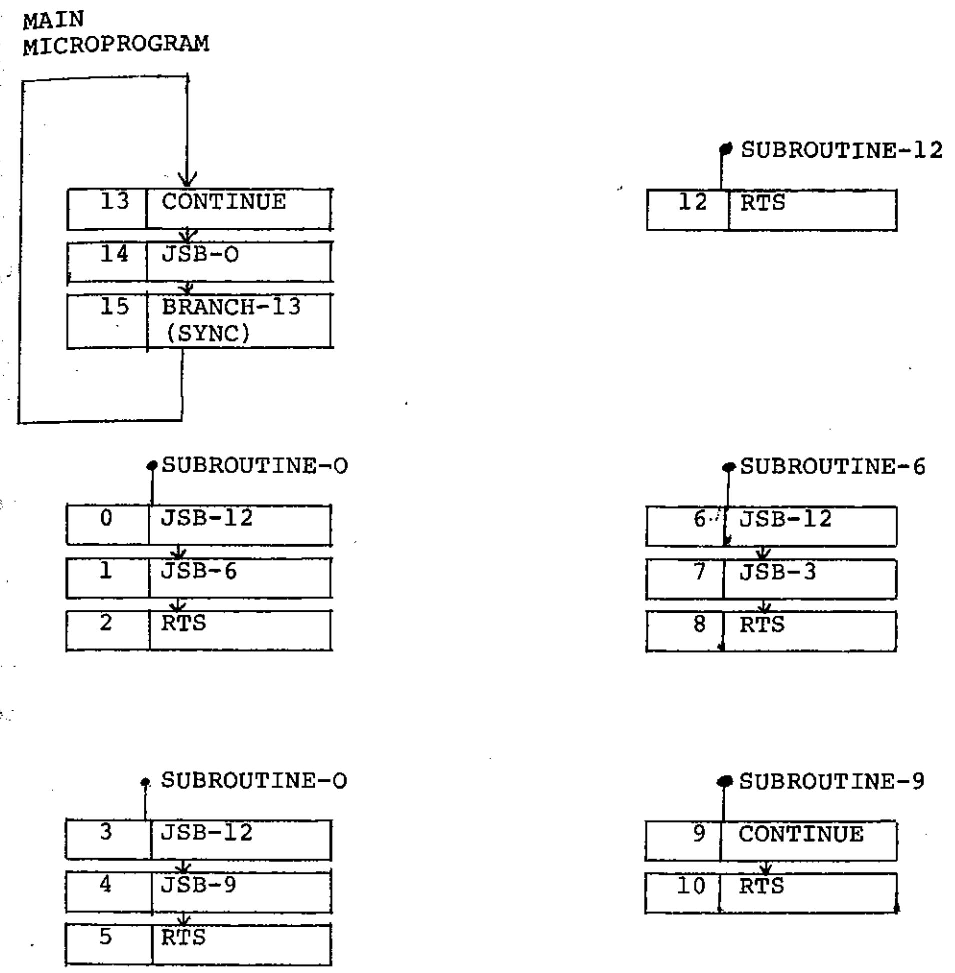
RAM & MUX	7	6	5		4		3		2	1	0	NOTES	
	BRANCH ADDRESS	NEXT µINSTRUCTION CONTROL	MUX1	DEST. SELECT	MUX0	SOURCE SELECT	Cn	ALU	'A'	'B'	'D'	NEXT ADDRESS CONTROL	DATA CONTROL
MEMORY ADDRESS	BR3 BR2 BR1 BR0	P3 P2 P1 P0		I8 I7 I6		I2 I1 I0		I5 I4 I3	A3 A2 A1 A0	B3 B2 B1 B0	D3 D2 D1 D0		
0	1100	0111										JSB 12	
1	0110	0111										JSB 6	
2		0101										RTS	
3	1100	0111										JSB12	
4	1001	0111										JSB9	
5		0101										RTS	
6	1100	0111										JSB 12	
7	0011	0111										JSB 3	
8		0101										RTS	
9		0001										CONT	
10		0101										RTS	
11													
12		0101										RTS	
13		0010										CONT	
14	0000	0111										JSB 0	
15	1101	0000										BR 13	

BLANK = DON'T CARE

INSTRUCTION JSB 12



MAIN MICROPROGRAM



Actual Address flow is:

13, 14, 0, 12, 1, 6, 12, 7, 3, 12, 9, 10, 5, 8, 2, 15, 13, etc.

Programming Exercise #8 - Combining the Am2901 and Am2909 to Perform Conditional Branching

This is the first exercise that combines both the Am2901 and Am2909 functions. It is used to demonstrate the technique of conditional branching in microprogram control. The microinstruction at word 1 contains a test on the carry flag whereby a conditional branch occurs if the carry output is logic 1.

This exercise should be executed in the following manner. After the data on the worksheet has been loaded into the microprogram memory, the MEMORY ADDRESS select switches should be placed at decimal 0 and the SINGLE STEP CLOCK momentary switch depressed. Now, the RUN/LOAD select switch should be placed in the RUN position. If the RAM & MUX select switches are placed in the decimal 0 position, the Am2909 Microprogram Sequencer next microinstruction address output is viewed. The DATA DISPLAY LED's will currently contain decimal value 1. As the SINGLE STEP CLOCK momentary switch is depressed, the data display pattern will follow a 1, 2, 0, 1, 2, 0 pattern until the carry output is a "logic 1". At this point, the next address will branch from microprogram memory word 1 to microprogram memory word 15 and then to microprogram memory word 0. This 1, 2, 0 loop will be repeated 15 times until the next carry flag occurs on the 16th interation and a branch to microword 15 again occurs.

The way this exercise operates is to take the contents of register 0 and increment the current value on microword 0. The instruction at microword 1 is used to test the carry flag stored in the status register. If the carry flag is logic 1, a branch to microword 15 is performed. At microword 15, a continue instruction is executed which results in an end around microword 0.

While this exercise is contained in the static testing mode description, it can also be used in the dynamic testing mode. Microinstruction 15 will occur once every 48 clock cycles. Because the design of the Am2900 Kit uses the Am2909 Sequencer with the carry input tied HIGH, each time address 15 occurs at the Am2909 outputs, the Am2909 carry output will be HIGH. It is this Am2909 carry output that is connected directly to the SYNC TEST turret terminal on the kit printed circuit board. Thus, in this example, the SYNC TEST turret terminal provides a convenient point to synchronize the oscilloscope to see the entire 48 word microprogram sequence. Likewise, the ADDRESS SYNC turret terminal can be used to view particular microinstruction address execution on an oscilloscope. That is, by setting the MEMORY ADDRESS select switches to a particular decimal value, a HIGH output on the ADDRESS SYNC turret terminal results each time the Am2909 Microprogram Sequencer address matches that of the MEMORY ADDRESS select switches. This allows one trace of a multiple trace oscilloscope to be connected to the ADDRESS SYNC turret terminal and the execution of a particular address studied. This will be discussed in more detail in the dynamic operational description of the programming exercises.

Am2900 KIT PROGRAMMING WORK SHEET

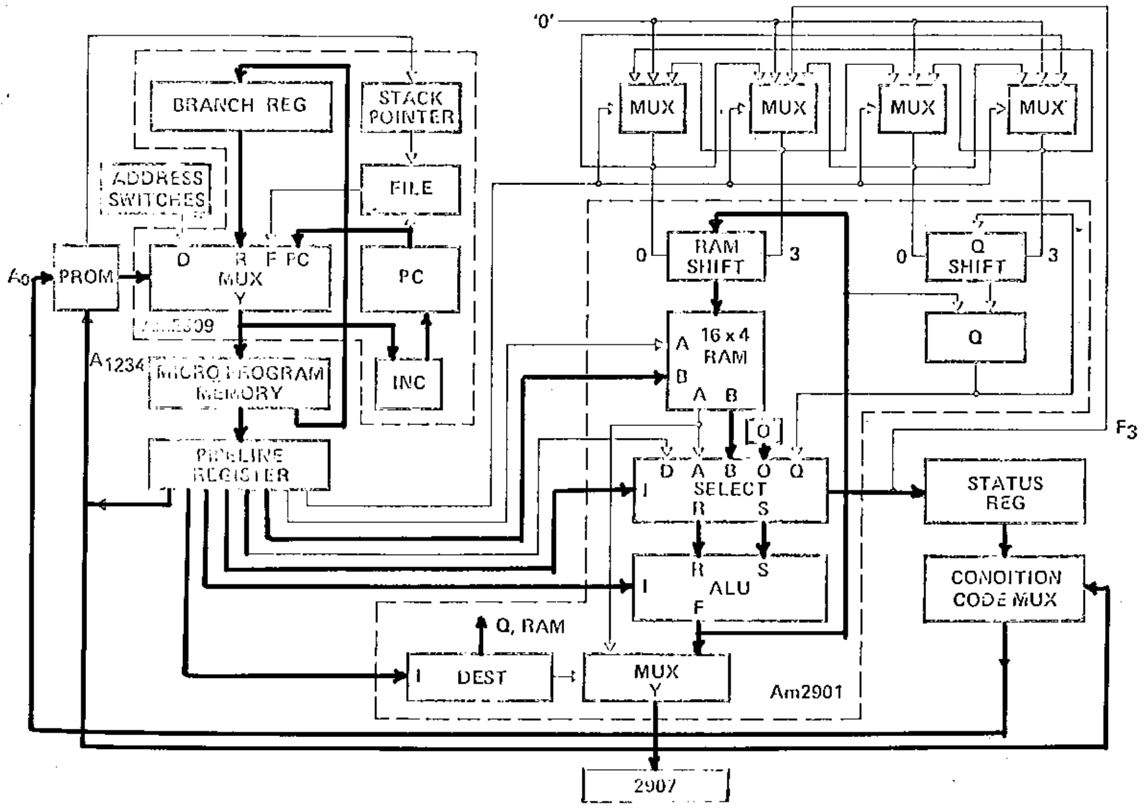
EXERCISE NUMBER: 8 NAME: INCREMENT AND TEST R₀

MICROPROGRAM MEMORY

RAM & MUX / MEMORY ADDRESS	7	6	5		5	4	4	3	3	2	1	0	NOTES	
	BRANCH ADDRESS	NEXT INSTRUCTION CONTROL	POL	DEST SELECT	MUX₀	SOURCE SELECT	C_n	ALU	'A'	'B'	'D'	NEXT ADDRESS CONTROL	DATA CONTROL
	BR_3 BR_2 BR_1 BR_0	P_3 P_2 P_1 P_0		I_8 I_7 I_6		I_2 I_1 I_0		I_5 I_4 I_3	A_3 A_2 A_1 A_0	B_3 B_2 B_1 B_0	D_3 D_2 D_1 D_0		
0		0010		011		011	1	000		0000		CONT	INC R₀
1	1111	1111		001								COND BR	BR15 Cary
2	0000	0001		001								BR 0	No OP
3													
4													
5													
6													
7													
8													
9													
10													
11													
12													
13													
14													
15		0010		001								CONT	No OP

BLANK = DON'T CARE

INSTRUCTION



Programming Exercise #9 - Measuring the B Address Access Time

The purpose of this exercise is to demonstrate the technique used to measure the access time of the Am2901 RAM with respect to the B address. First, microprogram memory word 0, 1, 14, and 15 should be loaded as shown on the worksheet. Now, the CLOCK SELECT switch should be in the SINGLE STEP position and the RUN/LOAD select switch should be in the LOAD position. The MEMORY ADDRESS select switches should be in the decimal 0 position and the SINGLE STEP CLOCK momentary switch depressed. This initializes the pipeline register to a starting address. Next, change the RUN/LOAD select switch to the RUN position. Now, depress the SINGLE STEP CLOCK momemtary switch two times so as to execute the microinstructions at word 0 and word 1. This performs the loading of register 0 with all zeros and register 15 with all ones.

Now, if a pulse generator has been connected to the PULSE GENERATOR inputs, when the CLOCK SELECT switch is changed to the PULSE GENERATOR position, the Am2900 Evaluation Kit will be operating in the dynamic mode. The SYNC TEST turret terminal can be used as a convenient oscilloscope sync point. The microprogram control will now be executing the microinstructions at word 14 and word 15.

The two microinstructions at word 14 and word 15 cause the B address to be changed from all zeros to all ones and then back to all zeros. The Am2901 RAM data output for these two registers is also all zeros and all ones. Thus, if the Am2901 Y outputs are probed with an oscilloscope and the Am2901 B inputs are also examined with an oscilloscope, the differential time between the B address change and the Y data output is measured - the B access time. The path selected in this example is to use the ALU "OR" function with the B and O source operands. In this example, the RAM is being re-written on each microinstruction; however, the no operation destination control instruction can also be used.

From this example, the user should recognize that the B address fields and the data fields could be selected in any fashion so that various patterns can be examined. Likewise, the entire Am2901 RAM might be initialized to all ones, all zeros, or a checkerboard pattern if the user desires. It should also be recognized that similar techniques can be used to measure various propagation delays such as the D inputs to the Y outputs, the A address inputs to the Y address outputs, carry-in to carry-out and so forth. In fact, almost all of the combinatorial propagation delays shown in Table II of the Am2901 data sheet can be measured in this manner. The key item to be remembered in performing any of these measurements is that only the variable to be measured changes between microcycles. That is, all other inputs to the Am2901 should be held constant except for the path being measured. For example, when measuring the carry input to carry output propagation delay, only the carry input should be changed during the microinstruction time of interest.

Am2900 KIT PROGRAMMING WORK SHEET

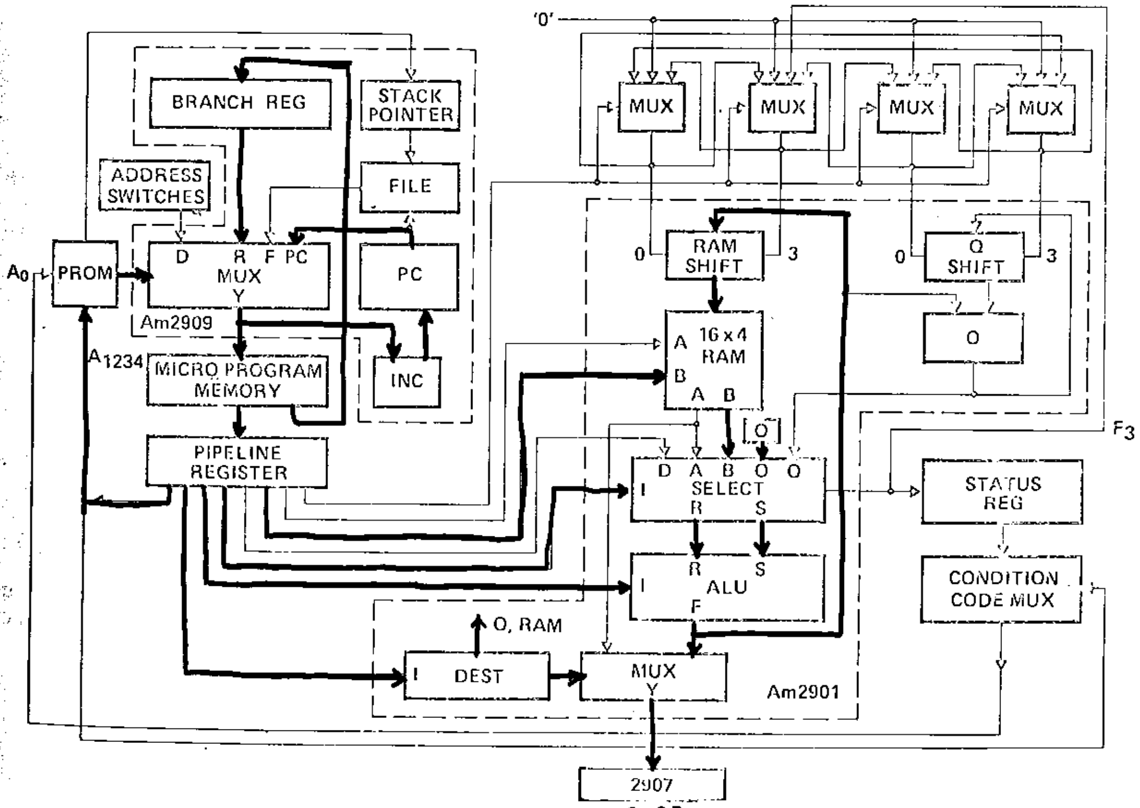
EXERCISE NUMBER: 9 NAME: EXAMPLE OF "B" ACCESS TIME

MICROPROGRAM MEMORY

RAM & MUX	7	6		5		4		3	2	1	0	NOTES	
	BRANCH ADDRESS	NEXT INSTRUCTION CONTROL	MUX1	DEST. SELECT	MUX0	SOURCE SELECT	C_n	ALU	'A'	'B'	'D'	NEXT ADDRESS CONTROL	DATA CONTROL
MEMORY ADDRESS	BR_3 BR_2 BR_1 BR_0	P_3 P_2 P_1 P_0		I_8 I_7 I_6		I_2 I_1 I_0		I_5 I_4 I_3	A_3 A_2 A_1 A_0	B_3 B_2 B_1 B_0	D_3 D_2 D_1 D_0		
0		0010		011		111		011		0000	0000	CONT	$R_0 = 0$
1	1110	0001		011		111		011		1111	1111	BR 14	$R_{15} = 15$
2													
3													
4													
5													
6													
7													
8													
9													
10													
11													
12													
13													
14		0010		011		011		011		0000		CONT	READ R_0
15	1110	0001		011		011		011		1111		BR 14	READ R_{15}

BLANK = DON'T CARE

INSTRUCTION B Access (14, 15)

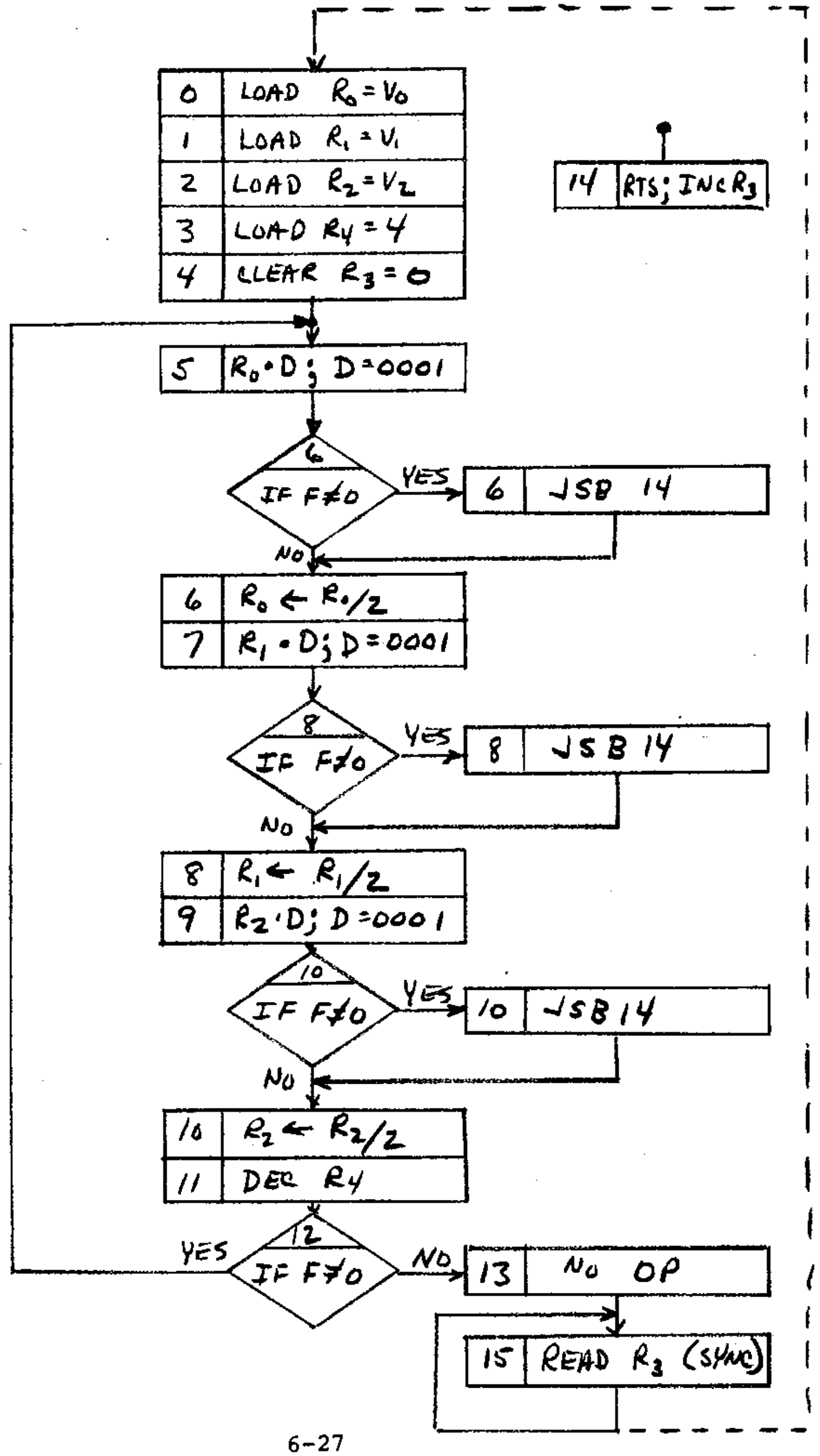


Programming Exercise #10 - Counting the Total Number of Ones in Three Register Words

This exercise demonstrates the use of the conditional jump-to-subroutine function and uses most of the paths in the Am2900 Evaluation and Learning Kit. V_0, V_1, and V_2 are three data values that are written into the data field of microprogram memory words 0, 1, and 2. The user should make up these three values as desired. The microprogram sequence from microword 8 through microword 15 executes a series of microinstructions that will determine the total number of ones in the three data fields, V_0, V_1, and V_2. These data values are loaded into Register 0, Register 1, and Register 2 of the Am2901 RAM. This microprogram uses Register 3 to hold the running partial summation as the number of ones in each word are counted. Register 4 in the Am2901 RAM is used as a working register to count the number of cycles in the algorithm. The contents of Register 0, Register 1, and Register 2 are not retained in the Am2901 memory; but they are destroyed during the execution of the algorithm. The flow diagram shown below is a summary of the algorithm as executed. The data value applied on the D input during the AND operation with each Register 0, 1, and 2 is used as a mask word.

As shown in the programming worksheet, if the program is executed it will finally reach memory word 15 and branch on itself thereafter. Memory word 15 is used to read register 3 such that the DATA DISPLAY can be used to read the total sum of ones in the V_0, V_1, and V_2 data fields. If field 6 of microprogram memory word 15 is changed from decimal 1 to decimal 2, instead of branching on itself at microprogram memory word 15, the microprogram will now continue from address 15 to address 0 and repeat the sequence. This allows the total sequence to be executed in the dynamic mode such that all the various instructions can be viewed using an oscilloscope. If the values V_0, V_1, and V_2 are to be changed, the kit should be switched from the dynamic mode to the static mode and these three data fields reloaded with the new values. Then, the kit can be switched back to the dynamic mode and the new sequence evaluated using an oscilloscope.

The oscilloscope can be synchronized to the total microprogram sequence by using the SYNC TEST turret terminal as a master sync point. This SYNC TEST point will provide one pulse each time the total sequence is executed. The ADDRESS SYNC turret terminal can be used as one trace on the oscilloscope to gain an instruction execution reference. For example, if the MEMORY ADDRESS switches are placed in the decimal 14 position, a sync pulse will be generated each time the algorithm jumps to the subroutine at microword 14 to increment Register 3. Likewise, if the MEMORY ADDRESS select switches are placed in the decimal 11 position, the end point of the four major cycles (decrement, register 4) can be referenced at the ADDRESS SYNC turret terminal.



Test the total number of 1's in R_0, R_1 and R_2. Use R_3 to hold the number of 1's and R_4 as a cycle counter. R_0, R_1 and R_2 are not retained.

Am2900 KIT PROGRAMMING WORK SHEET

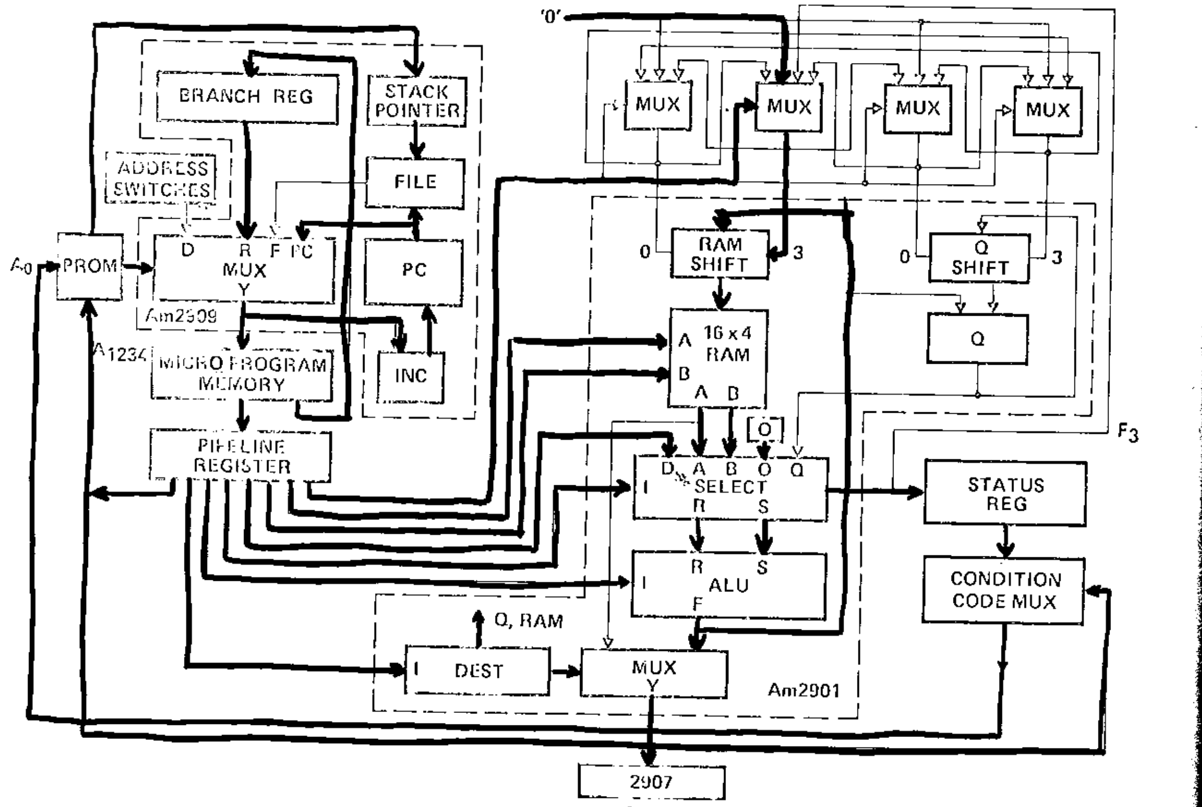
EXERCISE NUMBER: 10 NAME: NUMBER OF ONE'S IN V_0, V_1 & V_2

MICROPROGRAM MEMORY

RAM & MUX / MEMORY ADDRESS	7 BRANCH ADDRESS $BR_3\,BR_2\,BR_1\,BR_0$	6 NEXT INSTRUCTION CONTROL $P_3\,P_2\,P_1\,P_0$	MUX_1	5 DEST. SELECT $I_8\,I_7\,I_6$	MUX_0	4 SOURCE SELECT $I_2\,I_1\,I_0$	C_n	3 ALU $I_5\,I_4\,I_3$	2 'A' $A_3\,A_2\,A_1\,A_0$	1 'B' $B_3\,B_2\,B_1\,B_0$	0 'D' $D_3\,D_2\,D_1\,D_0$	NOTES: NEXT ADDRESS CONTROL	NOTES: DATA CONTROL
0		0010		011		111		011		0000	V_0	CONT	LD V_0
1		0010		011		111		011		0001	V_1	CONT	LD V_1
2		0010		011		111		011		0010	V_2	CONT	LD V_2
3		0010		011		111		011		0100	0100	CONT	$R_4 = 4$
4		0010		011		011		100		0011		CONT	$R_3 = 0$
5		0010		001		101		100	0000	0000	0001	CONT	$R_0 \cdot$ MASK
6	1110	0100		101		011		011		0000		JSB14 F/0	$R_0 \leftarrow R_0/2$
7		0010		001		101		100	0001	0001	0001	CONT	$R_1 \cdot$ MASK
8	1110	0100		101		011		011		0001		JSB14 F/0	$R_1 \leftarrow R_1/2$
9		0010		001		101		100	0010	0010	0001	CONT	$R_2 \cdot$ MASK
10	1110	0100		101		011		011		0010		JSB14 F/0	$R_2 \leftarrow R_2/2$
11		0010		011		011	0	001		0100		CONT	DEC R_4
12	0101	0000		001								BR5 F/0	NO OP
13	1111	0001		001								BR 15	NO OP
14		0110		011		011	1	000		0011		RTS	INC R_3
15	1111	0001		001		011		011		0011		BR15	READ R_3

BLANK = DON'T CARE

INSTRUCTION JSB 14 IF F≠0 (# 6, 8, 10) & CONT (5, 7, 9)

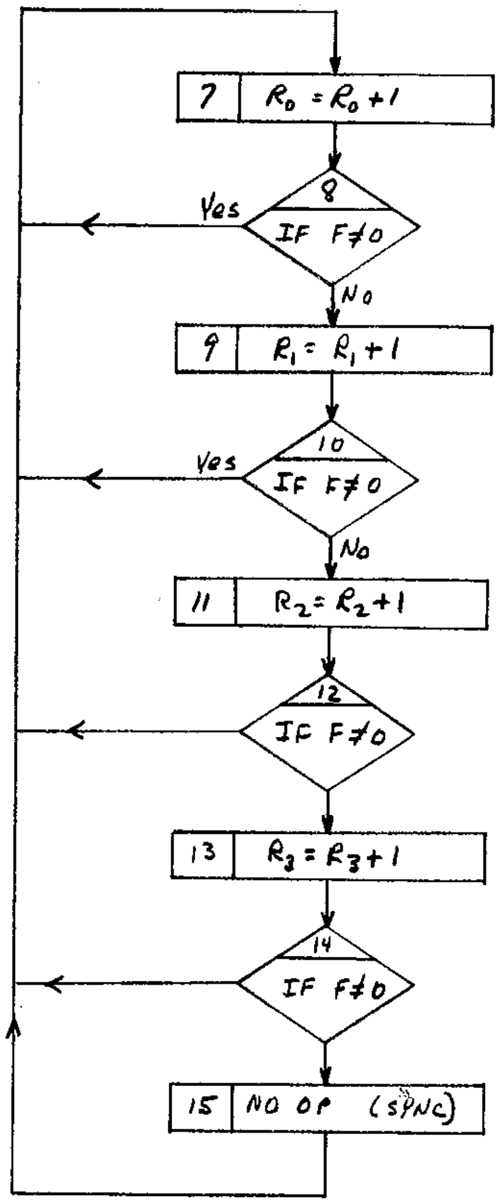


PROGRAMMING EXERCISE #11 - 16-BIT PROGRAMMED COUNTER

This exercise demonstrates a technique for using four internal Am2901 RAM registers to emulate a 16-bit counter. In this example, Register 0 represents the least significant four bits. Then, Register 1 is used for the second four-bit field; Register 2 is used for the third four-bit field and Register 3 is used for the most significant four-bit field. The microprogramming sequence as demonstrated in this example could be used as a subroutine that is called each time an event occurs. A conditional Return-from-Subroutine could be used rather than conditional branch to word 7.

When the program reaches memory word 15, all four internal registers are at 0. At this point, 2^{16} calls of this subroutine would be required before microprogram state 15 will again be reached. The exercise is intended to be used in the dynamic mode; however, the user may wish to preload Register 0 through Register 3 with binary 15 (1111) so that the total branch path can be demonstrated in the static mode.

The flow diagram shown on the following page is a summary of the operation of this algorithm as programmed on the worksheet. By this point, the user should understand the technique required to initialize the sequence using the Am2900 Evaluation and Learning Kit.



EXERCISE # 11

Am2900 KIT PROGRAMMING WORK SHEET

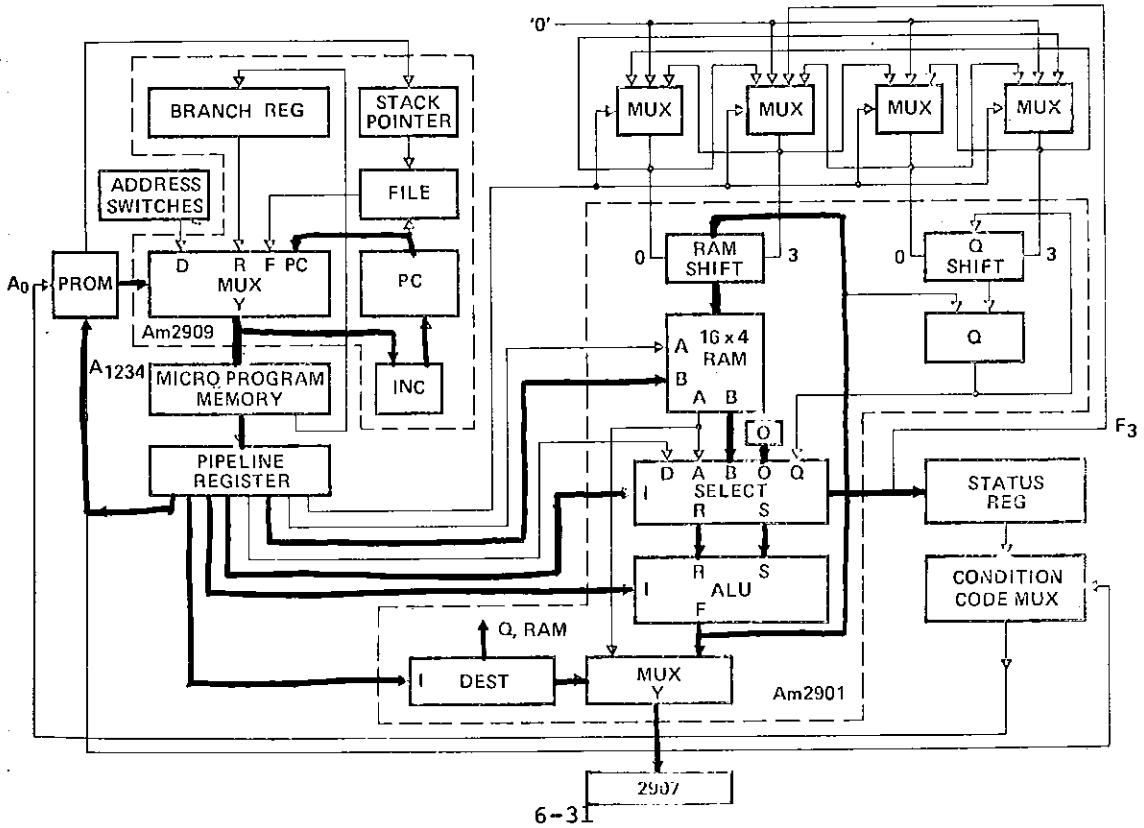
EXERCISE NUMBER: 11 NAME: 16-BIT COUNTER

MICROPROGRAM MEMORY

RAM & MUX	7	6	5	5	4	4	3	3	2	1	0	NOTES	NOTES
	BRANCH ADDRESS	NEXT µINSTRUCTION CONTROL	MUX_1	DEST. SELECT	MUX_0	SOURCE SELECT	C_n	ALU	'A'	'B'	'D'	NEXT ADDRESS CONTROL	DATA CONTROL
MEMORY ADDRESS	BR_3 BR_2 BR_1 BR_0	P_3 P_2 P_1 P_0		I_8 I_7 I_6		I_2 I_1 I_0		I_5 I_4 I_3	A_3 A_2 A_1 A_0	B_3 B_2 B_1 B_0	D_3 D_2 D_1 D_0		
0													
1													
2													
3													
4													
5													
6													
7		0010		011		011	1	000		0000		CONT	$R_0 = R_0 + 1$
8	0111	0000		001								BR7 F≠0	
9		0010		011		011	1	000		0001		CONT	$R_1 = R_1 + 1$
10	0111	0000		001								BR7 F≠0	
11		0010		011		011	1	000		0010		CONT	$R_2 = R_2 + 1$
12	0111	0000		001								BR7 F≠0	
13		0010		011		011	1	000		0011		CONT	$R_3 = R_3 + 1$
14	0111	0000		001								BR7 F≠0	
15	0111	0001		001								BR7	

BLANK = DON'T CARE

INSTRUCTION $R_0 = R_0 + 1$



Summary of Exercises

The exercises presented in Section VI of the Am2900 Evaluation and Learning Kit Instruction Manual have presented a number of different ideas associated with microprogramming. The user might attempt a number of other exercises to gain additional experience with this kit. Some ideas for these exercises are presented below.

1. Multiplication by a constant integer.

2. Division by a constant integer.

3. Counting the number of zeros in two register words.

4. Find the highest numeric value among four words.

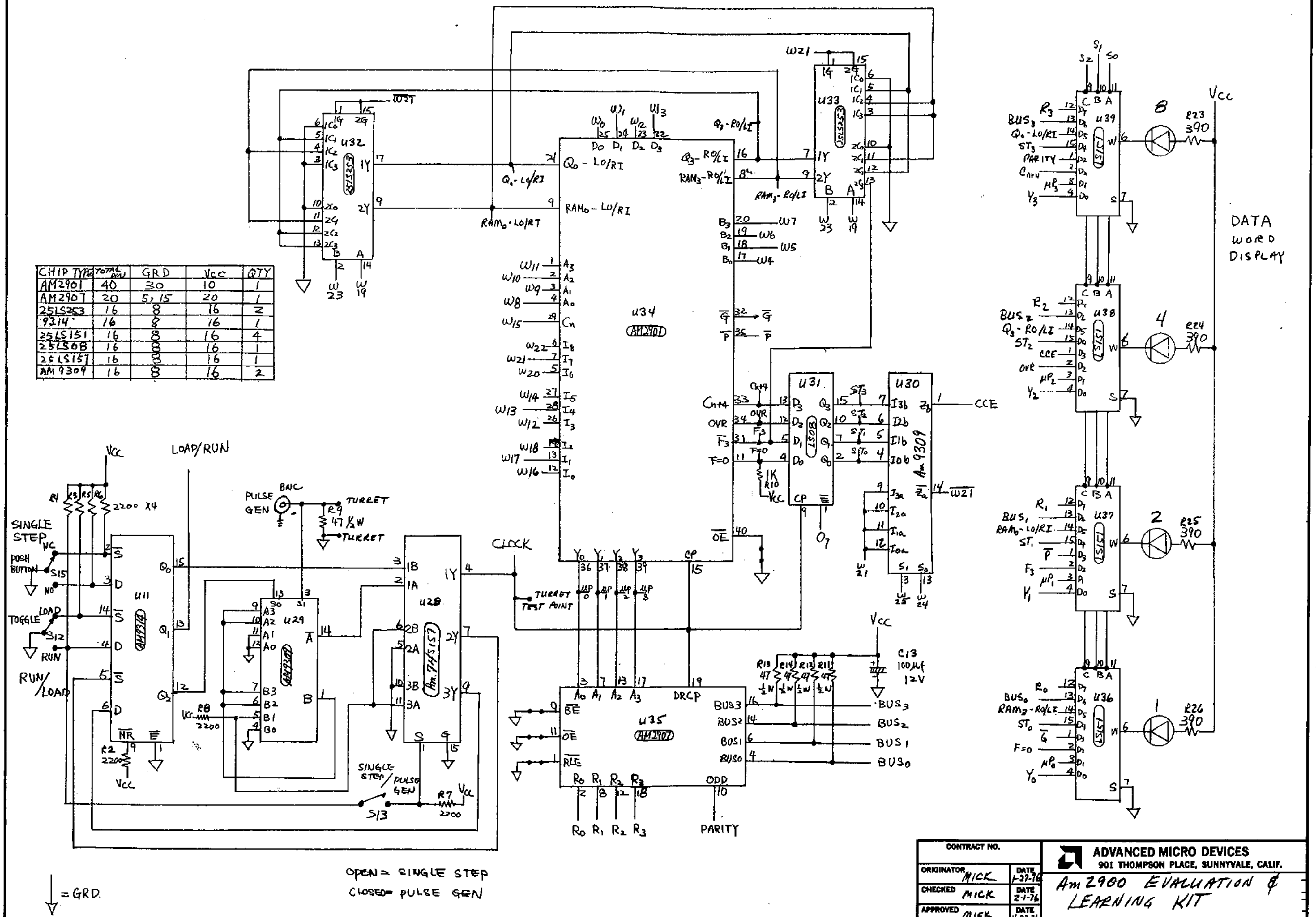
5. Order three or four words in descending numerical order.

6. Perform a byte swap on one word.

7. Perform a logic compare on two words and count the number of bits not matching.

8. Add two registers and test for an arithmetic overflow

There are many such examples of small microprogram sequences of instructions that can be generated using this kit. Remember, however, the goal of this kit is to allow the engineer not familiar with microprogramming to grasp the concepts involved in microprogramming and not necessarily be able to use the kit to perform all possible combinations of microprogram sequences for instructions that can be suggested. Also, the Am2901, Am2907, and Am2909 dynamic performance can be evaluated.



CHIP TYPE	TOTAL PIN	GRD	Vcc	QTY
AM2901	40	30	10	1
AM2907	20	5, 15	20	1
25LS253	16	8	16	2
9314	16	8	16	1
25LS151	16	8	16	4
25LS08	16	8	16	1
25LS157	16	8	16	1
AM9309	16	8	16	2

CONTRACT NO.	
ORIGINATOR MICK	DATE 1-27-76
CHECKED MICK	DATE 2-1-76
APPROVED MICK	DATE 4-22-76
APPROVED	DATE
APPROVED MICK	DATE 5-1-76

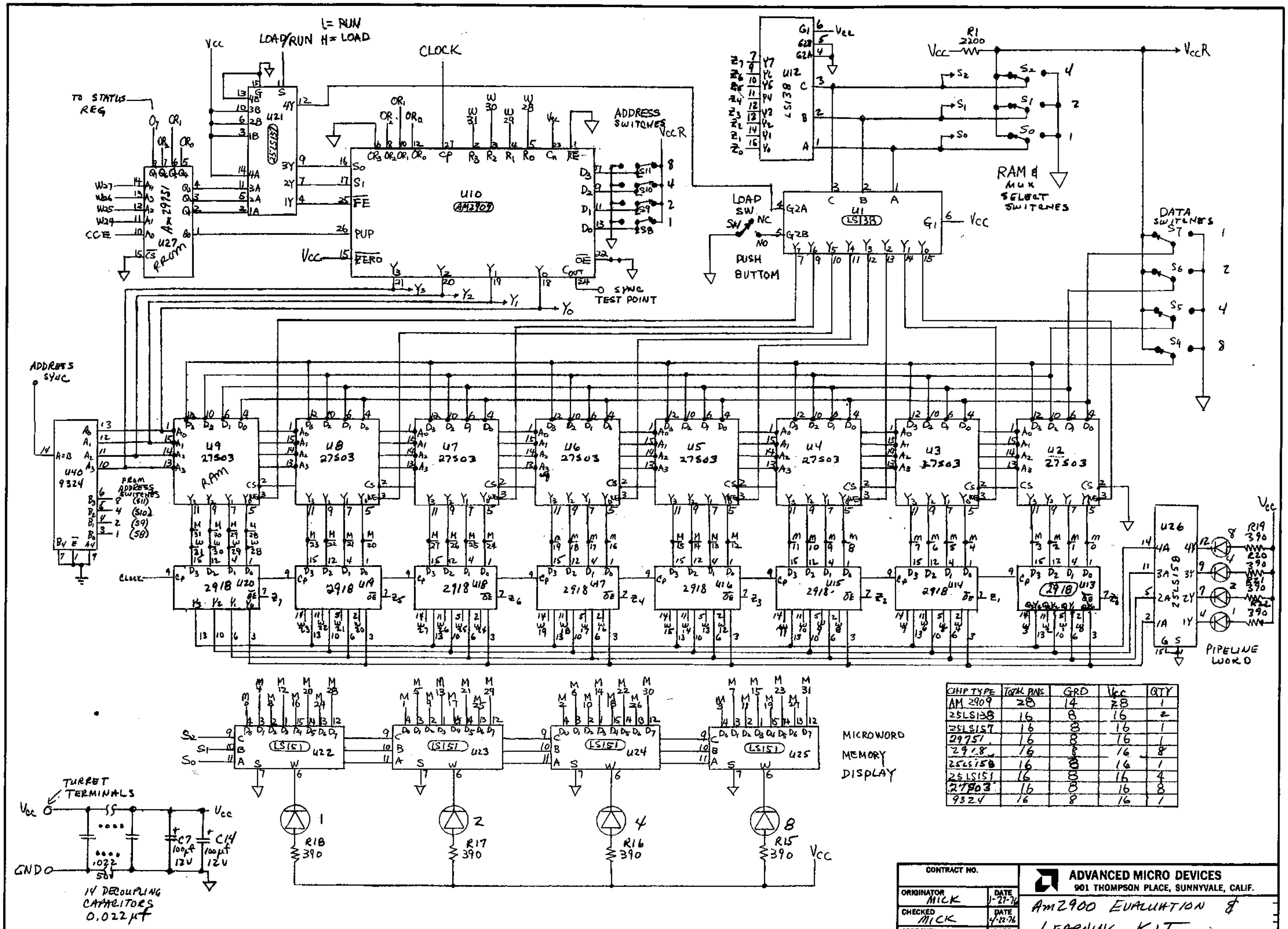
ADVANCED MICRO DEVICES
901 THOMPSON PLACE, SUNNYVALE, CALIF.

Am2900 EVALUATION & LEARNING KIT

SIZE	CLASS	DRAWING NO.
C	SCHEMATIC	Am2900EKL

SCALE: SHEET 2 OF 2

BISHOP GRAPHICS/ACCUPRESS REORDER NO. 2046



CHIP TYPE	TOTAL PINS	GRD	Vcc	QTY
AM 2909	28	14	28	1
25LS138	16	8	16	2
25LS157	16	8	16	1
29751	16	8	16	1
2918	16	8	16	8
25LS158	16	8	16	1
25LS151	16	8	16	4
27S03	16	8	16	8
9324	16	8	16	1

CONTRACT NO.		ADVANCED MICRO DEVICES 901 THOMPSON PLACE, SUNNYVALE, CALIF.
ORIGINATOR MICK	DATE 1-27-76	Am2900 EVALUATION & LEARNING KIT
CHECKED MICK	DATE 4-22-76	
APPROVED	DATE	
APPROVED	DATE	SIZE C · CLASS SCHEMATIC · DRAWING NO. Am2900 EKL
APPROVED MICK	DATE 5-1-76	SCALE: · SHEET 1 OF 2

BISHOP GRAPHICS/ACCUPRESS REORDER NO. 2546