Bipolar Bit Slice Microprocessors

Mack Duggan Cooley

University of Central Florida

Find similar works at: https://stars.library.ucf.edu/rtd
University of Central Florida Libraries http://library.ucf.edu

This Masters Thesis (Open Access) is brought to you for free and open access by STARS. It has been accepted for inclusion in Retrospective Theses and Dissertations by an authorized administrator of STARS. For more information, please contact STARS@ucf.edu.

STARS Citation
https://stars.library.ucf.edu/rtd/406
BIPOLAR BIT SLICE MICROPROCESSORS

BY

MACK DUGGAN COOLEY, JR.
B.S.E., Florida Technological University, 1972

RESEARCH REPORT

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Engineering in the Graduate Studies Program of the College of Engineering at the University of Central Florida

Fall Quarter
1979
BIPOLAR BIT SLICE MICROPROCESSORS

BY

MACK DUGGAN COOLEY, JR.

ABSTRACT

The purpose of this paper is to undertake a survey of state of the art high speed bipolar bit slice microprocessor elements. Three systems are discussed: The 2900 microprocessor family; the 10800 microprocessor family and the 8X02 microcontroller. A description of the microprocessor and microprogram controller elements for each family is presented and an example of the 2900 family is discussed.
ACKNOWLEDGMENTS

The author is indebted to Dr. Benjamin W. Patz for his invaluable counsel throughout this project and would also like to express his gratitude to Mr. Warren Miller, Applications Engineer, of Advanced Micro Devices and Myra L. Cooley for their assistance.
TABLE OF CONTENTS

ACKNOWLEDGMENTS ........................................ iii
LIST OF FIGURES ........................................... v

I. INTRODUCTION ........................................ 1
II. THE MICROPROCESSOR FAMILIES ........................ 3
III. THE MICROPROCESSOR SEGMENTS ...................... 8
IV. THE MICROPROGRAM CONTROL SEGMENTS .............. 14
V. SYSTEM COMPLEXITY AND TIMING ....................... 23
VI. THE RK11 DISK DRIVE CONTROLLER ................... 31
VII. CONCLUSION ........................................ 40
REFERENCES ............................................ 43
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Microprocessor functional areas</td>
<td>4</td>
</tr>
<tr>
<td>2.</td>
<td>2901 Microprocessor block diagram</td>
<td>9</td>
</tr>
<tr>
<td>3.</td>
<td>10800 Microprocessor block diagram</td>
<td>11</td>
</tr>
<tr>
<td>4.</td>
<td>2911 Microprogram sequencer block diagram</td>
<td>15</td>
</tr>
<tr>
<td>5.</td>
<td>10801 Microprogram sequencer block diagram</td>
<td>17</td>
</tr>
<tr>
<td>6.</td>
<td>8X02 Microprogram sequencer block diagram</td>
<td>20</td>
</tr>
<tr>
<td>7.</td>
<td>2901/2911 System Architecture</td>
<td>26</td>
</tr>
<tr>
<td>8.</td>
<td>10800/10801 System Architecture</td>
<td>27</td>
</tr>
<tr>
<td>9.</td>
<td>Block diagram of the RKll Disk Controller</td>
<td>33</td>
</tr>
</tbody>
</table>
I. INTRODUCTION

Since the appearance of the microprocessor and microcomputer in the early 1970's, the microprocessor has continually changed in a technological evolution. With the advent of large scale integrated (LSI) circuit technology, the microprocessor has become cheaper, smaller, faster, more flexible and requires less power [1,2]. The first type of LSI technology used to manufacture microprocessors consisted of metal oxide semiconductor (MOS) devices. These devices were characterized by single chip processors and relatively slow execution speeds. As demand for greater speed arose, the bipolar Schottky, and emitter coupled logic (ECL) technology were developed. With this technology the bit slice microprocessor was developed. Its characteristics consist of multiple CPU devices, programmable instruction sets, and increased speeds of 10 to 1 (over MOS microprocessors).

It is therefore the intent of this paper to present information concerning the state of the art high speed bipolar bit slice microprocessors. Elements of three
systems are discussed and an example of one system implementation is presented.

To provide current information associated with the bit slice microprocessors, the industry was surveyed. The survey revealed the major technology used was bipolar Schottky, with a minor usage being that of emitter coupled logic [3,4]. This paper therefore presents the 2900 family, and 8X02 controller as bipolar Schottky devices, and the 10800 family as the emitter coupled logic devices.
II. THE MICROPROCESSOR FAMILIES

The bit slice microprocessor families discussed consist of the AM 2900 family, the MC 10800 family, and the 8X02, a microprogram sequencer. The discussion of these microprocessor families is based upon the architecture of the microprocessor, its control and its interface within a computer system. The microprocessor may be functionally split into the sections of processing, control, and interface (Figure 1). The processing section provides data manipulation, temporary data storage, and status information associated with the data manipulation. The control section provides control over data transfers, data manipulation, status information monitoring and program sequencing. The interface section provides a communication link between memory devices, input-output devices and the processing/control sections.

The bit slice microprocessor is an architecture of a microprocessor which contains segmented processor and control sections. It is from this segmenting that the term "bit slice" is derived. These segments are totally independent sections, which can be cascaded to yield a
Figure 1. Microprocessor Functional Areas
microprocessor design dependent upon a specific application. The term "bit slice" will hereafter be replaced by the term "segment".

The AM 2900 family is a product of the Advanced Micro Devices (AMD) Company. It utilizes LSI low-power Schottky technology and was first introduced in 1975. Today, this family of devices is widely used and second sourced by several other manufacturers.

In developing the 2900 family, AMD selected eight functional areas for device development. The eight areas chosen were data manipulation, microprogram control, input/output control, memory control, direct memory access control, priority interrupt control, and control panel operation. Of these, the main functions of interest are data manipulation, and microprogram control.

To provide the data manipulation and the microprogram control, Advanced Micro Devices developed a four bit microprocessor and a four bit microprogram controller. These areas are further separated into the 2901 and 2903 microprocessors for data manipulation, and the 2909, 2910 and 2911 sequencers for microprogram control.

The MC 10800 family is a product of Motorola Semiconductor Products. It utilizes LSI emitter coupled logic (ECL) technology and was first announced in 1976 [6].
Today this family consists of four devices with the sole source supplier being Motorola.

In developing the 10800 bit slice family, Motorola selected four functional areas for device development. The areas chosen were data manipulation, microprogram control, memory interface control, and timing. The main functions of interest are data manipulation and microprogram control.

To provide these functions, Motorola has developed a family of segmented devices. Each device is organized in a four bit wide configuration. The data manipulation function is handled by the arithmetic logic unit (ALU), which is designated the 10800 device. The microprogram control function is provided by the 10801, whereas the memory interface and timing functions are handled by the 10803 and 10802, respectively [7,8].

The 8X02 microprogram sequencer is a product of the Signetics Corporation. It utilizes LSI low-power Schottky technology and was first announced in 1977. Today, this device is only supplied by Signetics [9].

When studying these microprocessor families, the topics of system complexity and timing became of interest. To cover these topics, two comparisons will be discussed. For data manipulation, a comparison between the 2901
microprocessor and 10800 microprocessor architectures will be discussed. For microprogram control, a comparison between the 2911 microprogram sequencer, the 10801 microprogram sequencer and the 8X02 microprogram sequencer will be discussed. However, prior to the comparisons, a brief description of each device will be given.
III. THE MICROPROCESSOR SEGMENTS

The functional area of data manipulation is provided by the microprocessor segments of the 2900 and 10800 families. These segments consist of the 2901 microprocessor and the 10800 microprocessor. A brief description of each microprocessor will be presented.

The 2901 is a four bit wide bipolar microprocessor segment. The main elements of this device consist of a high speed arithmetic logic unit (ALU), and a sixteen word by four bit random access memory (RAM); Figure 2.

The random access memory is a two port addressable memory with data selection being provided by the A or B address words. The RAM input is provided by a three input multiplexer used to select the ALU output in a non-shifted, shift right or shift left configuration. During the process the B word address is used to designate the memory address. The RAM output is provided by two outputs, A and B. Each of these ports are linked to output latches. The latch outputs therefore provide an ALU input.

The arithmetic logic unit is a high speed element which performs binary arithmetic and logic operations. The input to the ALU is provided by a data source selector.
Figure 2. 2901 Microprocessor Block Diagram
This selector provides the ability of selecting five data sources for the two ALU inputs, R and S. The data sources available consist of a direct data source (external to the segment), the A and B output ports of the RAM, a zero valued input and the output of the Q register. The output of the ALU is provided by the F port. This port provides the ALU output to the output data selector, the Q register and the RAM shift network. ALU status control is provided by five signals. They consist of a carry in (CIN), carry out (CN+4), carry generate (G), carry propagate (P), and overflow signals.

The output of the 2901 microprocessor is provided by the Y output port. The source of this port is determined by the output data selector. This device provides the capability of selecting the RAM A output port or the ALU output port as the microprocessor output.

The 10800 is a four bit wide emitter coupled logic device. The main element of this device consists of an arithmetic logic unit, and an accumulator (Figure 3).

The arithmetic logic unit is a high speed element which performs binary arithmetic, BCD arithmetic, and logical operations. To provide inputs to the ALU, a multiplexer, and latch/mask network are used. With these elements the A port, the 0 port, or the accumulator can
Figure 3. 10800 Microprocessor Block Diagram
be selected as ALU inputs. To provide status control for the ALU, six signals are provided. They consist of a carry in (CIN), carry out (COUT), group propagate (PG), group generate (GG), overflow (OF) and parity carry (PAR CAR) signals. At the output of the ALU is a shift network. This network provides shift left, logical shift right, arithmetic shift right, and ripple through operations. Status control of the shift network is provided by the zero detect (ZD) and parity results (PAR RES) signals. The output of this network is provided to the accumulator. For interconnection of segments, the R1 and R4 signals are available. These signal paths allow shifted data to be transferred to other segments.

The accumulator is a high speed element used as a temporary storage register for any repeated numerical operation. Input selection to the accumulator is provided by a four input multiplexer. Input data may be selected from the shift network, the input port, the output port, or the accumulator itself. The output of the accumulator provides data to the ALU input, the ALU multiplexer input, and the output port.

The 10800 microprocessor provides three input ports and two output ports. This interface is accomplished by
two bidirected ports (I and O) and a single direction input port (A).
IV. THE MICROPROGRAM CONTROL SEGMENTS

The functional area of microprogram control is provided by the microprogram sequencer segments. These segments consist of the 2911 microprogram sequencer, the 10801 microprogram sequencer, and the 8X02 microprogram sequencer. A brief description of each sequencer will be presented.

The 2911 microprogram sequencer is a four bit wide segment. The main elements consist of an address register, a microprogram counter register, a four word last-in-first-out stack, and an input multiplexer as shown in Figure 4.

The multiplexer is a four input element used to select the address register, the direct inputs (i.e., inputs external to the segment), the microprogram counter or the stack as the source of the next microinstruction address. The next address control for the 2911 microprogram sequencer is provided by five input signals. They consist of two multiplexer select signals (S0, S1), a register enable (RE) signal, a file enable signal and a push/pop signal.

The microprogram counter consists of an incrementer
Figure 4. 2911 Microprogram Sequencer Block Diagram
followed by a register. Carry in (CIN) and carry out (CN+4) signals are provided in the incremeneter to allow for cascading of several segments.

The stack consists of a four word file with built in stack pointer. Each word in the file is 4 bits wide. The stack pointer is used as an up/down counter, with separate file enable and push/pop inputs being provided.

At the output of the multiplexer, a zero input signal is provided. This signal is used in conjunction with combinational logic to force all multiplexer output lines to a zero state. The output of this logic is connected to the microprogram counter incremeneter input and to the 2911 output port (Y).

The 10801 microprogram sequencer is a four bit wide segment. The main elements consist of an address register, three general purpose registers, a last-in-first-out four word stack, an incremeneter and a next address logic network as shown in Figure 5.

The memory address register (CRO) holds the present microprogram address. The inputs to this register are controlled by the next address logic and consist of the repeat register (CR1), the instruction register (CR2), the stack, the incremeneter, the next address (NA) input, the status register (CR3), the memory address register (CRO),
Figure 5. 10801 Microprogram Sequencer Block Diagram
the input port (I Bus), and the output port (O Bus). The selection of the specific source is determined by the next address control logic. This next address control logic performs the function of selecting the next address.

The repeat/counter register (CR1) is designed to be an index counter. This is used for repeating single or multiple microinstruction routines. A second use for this register is that of a temporary storage register. During branch instructions the present contents of the address register are stored in the repeat/counter register. This address may then be transferred back to the address register when a repeat instruction is requested.

The instruction register (CR2) is used as an instruction storage register. After a memory fetch has occurred, the subsequent memory branching address can be stored in the instruction register. This data can then be transferred through the next address logic to the address register (CRO) for addressing the next microinstruction.

The status register (CR3) is used for storing flag information. It can be loaded from the next address lines (i.e., NA0-NA3) or input bus lines (i.e., I Bus). Each bit can be set or cleared from the DIN input. Control is provided by the status control logic. With this capability, the CR3 status information can be used with external information for generating special branch
conditions.

The stack is a four word LIFO file consisting of registers CR4-CR7. Each word is 4 bits wide. Control of the push/pop operation is determined by the four input lines and the next address logic. Information transfers between the stack and the input and output bus ports are controlled by select lines CS6-CS7. The Stack Full status indication is provided by monitoring CR7 for non-zero information.

The incrementer provides an interface between the next address control logic and the address register. It is used in conjunction with the microprogram control instructions to provide sequential execution of the microprogram or to increment the index register (CR1) for repeated microinstructions. Sequential execution of the microprogram is provided by adding the contents of the address register and the carry in (CIN) signal. The carry in and carry out (COUT) signals are used when cascading to larger addresses.

The 8X02 is a ten bit wide microprogram unit. The main elements consist of an address register, an address multiplexer, a last-in-first-out four word stack, and a next address logic decode network as shown in Figure 6.

The address register consists of ten D-type flip-flops
Figure 6. 8X02 Microprogram Sequencer Block Diagram
with a common enable signal. This register is loaded on the leading edge of a low to high clock transition, with the data input being provided by the address multiplexer.

The address multiplexer is used to select the source of the next microcode address. These sources consist of the direct input, a plus 1 adder, a plus 2 adder, the stack and the zero input signals. The source selection is controlled by the decode logic circuit which decodes the address control functions and the test input signal. The address control functions are eight instructions used to control the next address, the stack, and the stack pointer.

The stack is a ten bit by four word file register. This file is used as a first-in-last-out device with the stack pointer indicating the next address. The input to the stack is provided by a multiplexer. This multiplexer is used to designate the memory address register, the MAR incremented by one, or the MAR incremented by two. The selection control of this multiplexer and the stack operation are provided by the decode logic circuit. The output of the stack provides an input to the address multiplexer.

The direct input (i.e., B0-B9) is used to provide branch addressing capabilities to the controller. In
conjunction with the branch input, the zero input signal is used. The decode logic allows the address multiplexer to select a zero valued input. The next address is therefore forced to be a zero.

The next address control functions are used to provide flexible sequence control. Implementation of these functions is provided by the decoded address control (AC0-AC2) signals and the test input.

Located at the controller's output is an output buffer. This element provides tri-state logic at the 8X02 address output and is controlled by a common enable signal (EN).
V. SYSTEM COMPLEXITY AND TIMING

The study of the 2900 microprocessor family, the 10800 microprocessor family, and the 8X02 microprogram sequencer has been based upon the architectures of the microprocessor and microprogram control segment. The topics of system complexity and timing become of great importance when a system is to be implemented. These topics will be discussed in the following paragraphs. The topic of system complexity is separated into microprocessor and microprogram control segment comparisons. The timing topic is covered by a comparison of a 2901 microprocessor/2911 microprogram sequencer architecture with that of a 10800 microprocessor/10801 microprogram sequencer architecture.

A comparison of the 2901 and 10800 microprocessor architectures exhibits two main differences. First, the 10800 microprocessor does not have a register file, and secondly, the input/output port organization between the two microprocessors differ with the 10800 using bidirectional ports and the 2901 using single directional ports (Figures 2 and 3). It is from these items that the 10800 implementation becomes more complex than the 2901 imple-
mentation. The 10800 microprocessor contains no internal register file, and therefore external memory must be linked to the microprocessor. More control is then required to maintain this memory link. A second point of concern is the bidirectional data ports. Since the ports provide bidirectional data processing, additional control over the port is required.

A comparison of the 2911, 10801 and 8X02 microprogram sequencer architecture exhibits two differences. First, the 2911 and 8X02 microprogram sequences use address multiplexers, the 10801 uses a next address logic network. Secondly, the 2911 and 8X02 sequencers use single directional data ports, the 10801 uses bidirectional ports as shown in Figures 4, 5 and 6.

The 2911 sequencer architecture is centered around an address multiplexer. This element provides next address selection from four sources, with the next address logic control (S0, S1) being provided by a device external to the 2911 sequencer. The multiplexer's output is provided by a single unidirectional port. The 8X02 sequencer is similar in architecture to the 2911. The 8X02 also makes use of an address multiplexer with address selection being provided from five sources. The next address control of the multiplexer is provided by a decode logic network external to the
8X02 controller. The multiplexer output consists of a single unidirectional port. The 10801 microprogram sequencer's architecture is based upon a next address logic network. The next address logic network is a multiplexer where the address source selection is determined by the instructional control. Data access to the 10801 sequencer is provided by three bidirectional ports, the I Bus, the O Bus and the extender bus. Each port may receive or transmit microprogram address data.

To this point, the microprocessor and microprogram control segments have been compared separately. However, to investigate the timing of these segments an architecture involving a combination of the segments is required. To perform this investigation, two similar architectures for the 2901/2911 and the 10800/10801 systems are used. For each system a 16 bit microprocessor, a sequencer, a microprogram memory and a pipeline register are configured. Figure 7 presents the 2901/2911 architecture whereas Figure 8 presents the 10800/10801 architecture.

The evaluation of each system involves two aspects. First, the operation to be compared between the two systems must be defined and secondly, the data path loop and the control path loop must be considered. The operation used for comparison is an add being performed with the
Figure 7. 2901/2911 System Architecture
Figure 8. 10800/10801 System Architecture
microprocessor register file supplying the operands. The resultant is then transferred to an output register. For each system, a look ahead carry device is used in the cascading of the microprocessor segments. The times used to accumulate a minimum cycle time for each system are maximum propagation delays determined at a 25 degree centigrade temperature.

For each of the 2901/2911 and 10800/10801 systems, a data loop time and control loop time will be given. The minimum cycle time of the system will be the maximum value of the data or control time. Referring to Figure 8, the 10800/10801 timing is as follows:

<table>
<thead>
<tr>
<th>Location</th>
<th>Data Loop</th>
<th>Time (nsec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Clock to output (MC10176)</td>
<td>5</td>
</tr>
<tr>
<td>B</td>
<td>R.F. Address to A/O bus (MC10145)</td>
<td>24</td>
</tr>
<tr>
<td>C</td>
<td>A Bus/O Bus to P,G (MC10800)</td>
<td>21</td>
</tr>
<tr>
<td>C</td>
<td>P,G to Cout (MC10179 - look ahead carry)</td>
<td>6</td>
</tr>
<tr>
<td>C</td>
<td>Cn to I Bus (sum carry out)</td>
<td>20</td>
</tr>
<tr>
<td>D</td>
<td>Register setup time</td>
<td>$\frac{5}{81n}$ seconds</td>
</tr>
<tr>
<td>Location</td>
<td>Control Loop</td>
<td>Time (nsec)</td>
</tr>
<tr>
<td>----------</td>
<td>--------------------------------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>A</td>
<td>Clock to output (MC10176)</td>
<td>5</td>
</tr>
<tr>
<td>D</td>
<td>Address select to output (MC10801)</td>
<td>33</td>
</tr>
<tr>
<td>E</td>
<td>Memory access (MC1049)</td>
<td>25</td>
</tr>
<tr>
<td>A</td>
<td>Pipeline setup</td>
<td>5</td>
</tr>
</tbody>
</table>

Reviewing the above data, the data loop time is 81 nano seconds and the control loop time is 68 nano seconds. The minimum cycle time for this system is 81 nano seconds.

Referring to Figure 7, the 2901/2911 timing is as follows:

<table>
<thead>
<tr>
<th>Location</th>
<th>Data Loop</th>
<th>Time (nsec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Clock to output (AM25LS273)</td>
<td>15</td>
</tr>
<tr>
<td>B</td>
<td>A,B address select to G,P (3 least significant seq)</td>
<td>65</td>
</tr>
<tr>
<td>B</td>
<td>G0, P0 to CN+Z (AM2902 look ahead carry)</td>
<td>11</td>
</tr>
<tr>
<td>B</td>
<td>CN to Y (most significant segment)</td>
<td>50</td>
</tr>
<tr>
<td>C</td>
<td>Register setup</td>
<td>146n seconds</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Location</th>
<th>Control Loop</th>
<th>Time (nsec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Clock to output (AM25LS273)</td>
<td>15</td>
</tr>
<tr>
<td>D</td>
<td>Address select to output (AM2911)</td>
<td>48</td>
</tr>
</tbody>
</table>
Reviewing the above data, the data loop time is 146 nano seconds and the control loop time is 123 nano seconds. The minimum cycle time is therefore 146 nano seconds.

Comparison of the two system times, 2901/2911 and 10800/10801, indicates the 10800/10801 system is faster. The largest deviation in the two systems is that of memory access time, 55 nano seconds for the 2901/2911 system and 25 nano seconds for the 10800/10801 system. It is also noted that the 10800 microprocessor register file is external from the microprocessor but does not add additional time delays compared to the internal register file of the 2901 microprocessor.
VI. THE RK11 DISK DRIVE CONTROLLER

Having presented a brief description of the 2900 family microprocessor and microprogram control segments, the 10800 family microprocessor and microprogram control segments, and the 8X02 microprogram sequencer, an article demonstrating an application of the 2900 microprocessor family will be discussed. This article presents a hardware emulation of a disk drive controller. However, prior to presenting the application, the disk controller to be emulated is discussed.

The RK11 is the disk controller used in conjunction with the RK05 disk drive to form a disk drive system that interfaces with the PDP-11 through a unibus. The RK05 is a moving head disk drive that uses the RK03-KA disk cartridge for data storage [10]. This equipment is manufactured by the Digital Equipment Corporation.

The development of the hardware disk drive emulator is based upon the ideas of higher density LSI systems, and the software emulation of disk drive controllers. With the advent of high density LSI system, the total number of components required to implement a system has been reduced. This provides more cost effective systems
and reduces the physical size of the system. The second development idea is software emulation. With the development of segmented microprocessor systems, microprogram control was developed. With this microprogram control, a writeable control store memory may be used. The control store memory may then be loaded with various system instruction sets, thereby providing software emulation.

To provide hardware emulation of the RK11 controller, the functional areas of bus control, status control and disk control are configured as shown in Figure 9 [11,12]. The architecture consists of a unibus interface, a disk interface, control and status registers, a microprogram store, a sequencer, and a microprocessor. The unibus interface provides a link between the microprocessor bus (M Bus) and the unibus. The disk interface provides a link between the M Bus and the disk. The control and status registers provide an interface between the unibus and the microprogram control store. The sequencer consists of three 2911 segments providing control and address information to the microprocessor. The microprocessor consists of two 2901 segments.

To understand the operation of this hardware emula-
Figure 9. RK11 Block Diagram
tion, reference is made to the general 2901 microprocessor/2911 microprogram control architecture of Figure 7. In this architecture, the elements of a microprocessor, a microprogram sequencer, a microprogram memory and a pipeline register are combined to create a microprocessor system. The architecture of the RK11 controller consists of two 2901 microprocessor segments, three 2911 microprogram sequencers, a microprogram memory (control store), and a pipeline register (microprogram register, Figure 9). Communication between the microprocessor system and the peripheral interfaces, unibus and disk, is provided by an 8 bit wide microprocessor bus (M Bus). Microprogram control storage is accomplished through control and status registers, a RAM write register file, and a 48 bit data bus. The control and status registers provide an interface with the unibus, while the RAM write registers provide a temporary storage location for incoming control store data.

The functional areas of the RK11 controller consist of bus control, status control and disk control. The bus control provides an interface between the unibus and the controller. Address selection, bus control and data
transfer is provided. The hardware emulator accomplishes this task by using the unibus interface, the M Bus, the microprocessor, and the sequencer. The unibus interface provides a 36 bit parallel input/output port with the unibus. The port is separated into a 16 bit data field, an 18 bit address field, and a 2 bit control field. Data transfer operations between the unibus and disk interface consist of (1) data being received from the unibus, (2) data temporarily being stored in the unibus interface, (3) 8 bit bytes transferred onto M Bus, (4) 8 bit bytes transferred to the microprocessor for sum check calculations, (5) 8 bit bytes transferred to disk interface for parallel to serial formatting. For each of the above operations, control is provided by the microprogram sequencer. The sequencer selects a microprogram address, the address data is transferred to the microprogram memory, and the memory data is in turn transferred to the microprogram register (pipeline register).

To provide status control between the RK11 and the unibus, seven programmable registers are provided. These registers consist of the following functions: disk drive status, word count information, bus address, disk address, and data buffer. With the use of these registers
the RK11 controller is able to perform four control and data transfer functions. The control functions consist of control reset, seek, drive reset, and write lock; whereas the data transfer functions are read, write, write check and read check.

The implementation of status control is provided by the unibus interface, M Bus, sequencer, and microprocessor. The seven programmable registers are contained in the 2901 microprocessor segments RAM register file. These registers are loaded from the disk interface or the unibus interface via the M Bus. Load control and status response due to changes in these registers is controlled by the microprogram sequencer.

The disk control of the RK11 is provided to convert data into a serial format and to provide a control interface between the disk drive and the status control. Status control is based upon the state of the seven programmable registers.

To provide the function of disk control, the disk interface, the M Bus, the microprocessor and the sequencer are used. The disk interface consists of two parallel input/output ports and a serial interface port. These
three ports provide an interface with the disk drive. The parallel input/output ports are single directional with one providing a disk status output and one providing disk control input. The disk interface provides a link between these ports and the M Bus. The M Bus provides a link between the microprocessor status registers and the disk interface. Control of the M Bus is provided by the microprogram sequencer. A second function of the disk interface is to provide a parallel to serial data formatting operation. This operation is accomplished with the use of a first-in-first-out (FIFO) register file. Data transferred from the unibus to the disk input consist of the following operations: (1) data is loaded into the unibus interface from the unibus, (2) the data is transferred from the unibus interface onto the M Bus, (3) the data is loaded from the unibus into the FIFO, (4) the data is converted from parallel data to serial data, (5) the data is outputted to the disk drive via the serial interface. Control of these operations is provided by the microprogram sequencer.

The discussion of the RK11 has centered around hardware emulation. In conjunction with this emulation, the RK11 disk drive controller also provides a writeable
microprogram memory. This memory may be loaded via the unibus, through the status and control registers to the microprogram memory (Figure 9). With this capability, the RK11 disk drive controller may be used to provide software emulation of various disk drive controllers.

The unibus interface provided for microprogram control store operation is referred to as the control and status registers, and RAM write registers of Figure 9. This interface consists of a 36 bit parallel input/output port and a Random Access Memory. The port is separated into a 16 bit data field, an 18 bit address field, and a 2 bit control field. The memory consists of four RAM devices arranged to provide sixteen 16 bit registers. This memory in turn provides microprogram data buffer registers (RAM write registers) which drive the RAM data (RD) bus. The RD bus is 48 bits wide and provides the data path between the unibus interface and the control store.

The control store memory is 48 bits wide by 1K words deep. The memory control is provided by the sequencer. The data input and output ports are bussed together for form common input-output lines. These input/output lines provide data links with the RD bus and the microprogram register.
The loading of the microprogram memory is initiated when a control signal is received from the unibus. Once this has occurred the sequencer, the RAM write registers and the control store memory (microprogram memory) are initialized. Data transfer from the unibus to the control store memory may then occur. With the use of this writeable microprogram memory, the RK11 may execute whatever microprogram instruction set has been loaded. Therefore, the RK11 may provide the control function of the RK11 controller or different controllers as the requirement may arise.
VII. CONCLUSION

Elements of the microprocessor systems of the 2900 family, the 10800 family, the 8X02, and the RK11 disk drive controller have been presented. As mentioned previously, these data were assembled through a survey of current literature. In concluding this paper, three items are noted. They consist of the lack of detailed applications available, the implementation of a writeable control in the RK11, and the typographical errors noted in the RK11 article (Reference 11).

When the survey was performed, a search was made of periodicals, conferences, and similar publications over the past four years. The object was to obtain several detailed system applications of the 2900, 10800 and 8X02 devices. However, the articles found were not to the depth required by this paper. These articles generally presented an overview of a particular application with little attention directed to details. The one article, however, that did present these details, was the RK11 disk drive controller using the 2900 family.
The second item of interest is that of an implementation of a writeable control store memory in a segmented microprocessor system, and the use of this segmented LSI architecture to minimize the physical size of the system. The discussion of the RK11 disk drive controller centered around the hardware emulation of a SSI disk controller. The purpose of this emulation was two fold. First, the use of LSI circuitry reduces the total number of devices required, and secondly, the use of microprogram control allows the development of writeable control store memories. For this reason, the capability of creating an instruction set emulation of various systems is provided.

The final item of interest is associated with the RK11 article (Chapter VI), and the typographical errors noted (Reference 12) in this article (Reference 11). To obtain a working knowledge of the disk controller, each schematic was reviewed in detail. This analysis revealed the following errors.

1. Figure 8 of reference 11 - On the lower 25LS138 device (M bus destination control) the R23 line should be connected to pin 4 and Vec should be connected to pin 6.
2. Figure 4 of reference 11 - The microprogram register output bus lines show an input bus not an output.

3. Figure 4 of reference 11 - The CP and CLR lines of the leftmost 25LS273 device (microprogram register) were not connected to the microprogram clock and LD MCODE L lines.

4. Figure 7 of reference 11 - On the topleft side, the 8837 device pin 5 is labeled UBG5 H. It should be labeled UBG6 H.

With these corrections to the RK11 disk controller article (Reference 11), the design appears to provide an emulation of the SSI RK11 disk controller.
REFERENCES


