



Microprogrammable Microprocessor Survey

Phillip M. Adams

Table of Contents

- Section 1. Introduction
- Section 2. Intel 3000
- Section 3. Monolithic Memories Inc. 5700/6700
- Section 4. Fairchild Macrologic 9400
- Section 5. Texas Instruments SBPO400
- Section 6. Motorola M10800
- Section 7. Texas Instruments 74S481 and 74S482
- Section 8. Advanced Micro Devices AM2900
- Section 9. Advanced Micro Devices 2903 and 2910
- Section 10. SIGNETICS 8X02
- Section 11. Perspective: Past-Present-Future
- Section 12. Conclusions/Recommendations

Section 1. Introduction

The purpose of this document is to provide a general overview of most of the commercially available microprogrammable microprocessors and associated microprogram sequencers. This document attempts to provide both an architectural overview as well as a microprogramming overview. No attempt is made in this document to provide the basic concepts of microprogramming, microprogramming terminology, etc. Instead, it is assumed that the reader has read or has access to the large amount of literature dealing with the topic of microprogramming, in general.

The order in which this document is compiled should aid in comparing various microprogrammable microprocessors. Each section deals with a different family of microprogrammable microprocessors, and is divided into the following sub-topics:

- Introduction
- Processing Element
- Sequencer (Microprogram Sequencer)
- Family (Support Devices)
- Advantages
- Disadvantages, and
- Annotated Bibliography

Of course the most subjective portion of this document is the portion dealing with the advantages and disadvantages of each device. This portion of the document should be read and viewed as one person's view based upon a given, yet never mentioned, set of design criteria. Possibly of most value is the annotated bibliography which includes a number of references to device specification manuals as well as device application reports.

Finally, it must be reiterated that this document attempts to provide only a general overview of popular microprogrammable microprocessors. For greater detail, it is recommended that the reader refer to the annotated bibliography. This will allow the reader to obtain enough detailed information to begin design or make processor decisions.

Section 2. Intel 3000

2.1 Introduction

The INTEL 3000 series or family basically consists of a sequencer, the Microprogram Control Unit (MCU) - 3001, and a processing element, the Central Processing Element (CPE) - 3002. These elements may be connected in a fashion similar to the one depicted in Figure 2.1.

Figure 2.2 depicts the design of a microinstruction necessary to control a system, such as depicted in Figure 2.1. The fields of the microinstruction of Figure 2.2 are separated and clearly marked, but should not be construed to mean that field overlap or assignment is disallowed.

Figures 2.3 and 2.4 depict the internal structure of the CPE and MCU respectively.

2.2 Processing Element

The 3002 CPE is a vertically complete yet horizontally expandable 2-bit processing element. Since each CPE contains its own registers, arithmetic and logic circuits, and internal busses it is possible to horizontally expand a system to virtually any desired word length. For example, to produce a 16-bit processor it is necessary to connect eight CPE's. Each CPE will, essentially, contribute its 2-bits to the current register word length and yet receive and function on the same 7-bit CPE field the other CPEs accept and act upon.

2.2.1 Register Structure

The 3002 CPE has twelve registers, namely, R0-R9, T and AC. A Memory Address Register (MAR) is also provided, R0-R9 and T usually serve as scratch pad registers, whereas, the data contained in the MAR or AC can be gated onto the output busses, the A-bus and the D-bus.

2.2.2 Processing Element Directives

In a typical configuration a 7-bit field, CPE field, in the microinstruction drives the CPE Function Bus. This determines the function to be executed by the CPE during a microinstruction cycle. The CPE performs over 40 Boolean and arithmetic operations including 2's complement arithmetic and logical AND, OR, NOT, and

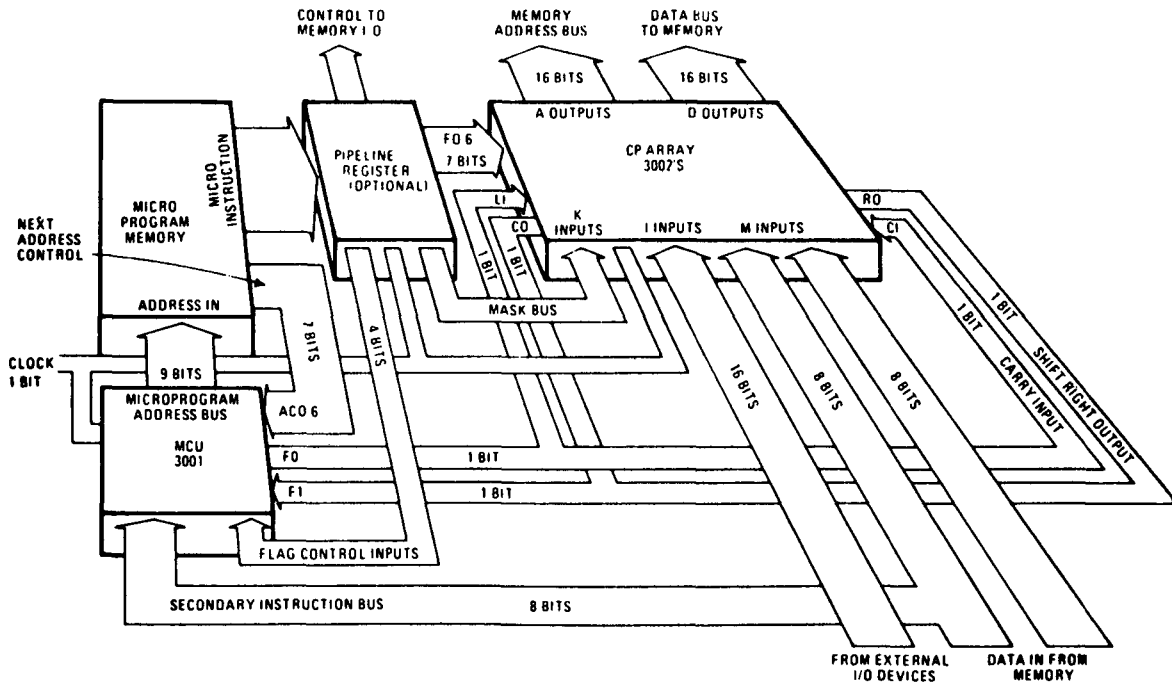


FIGURE 2.1. TYPICAL CONFIGURATION

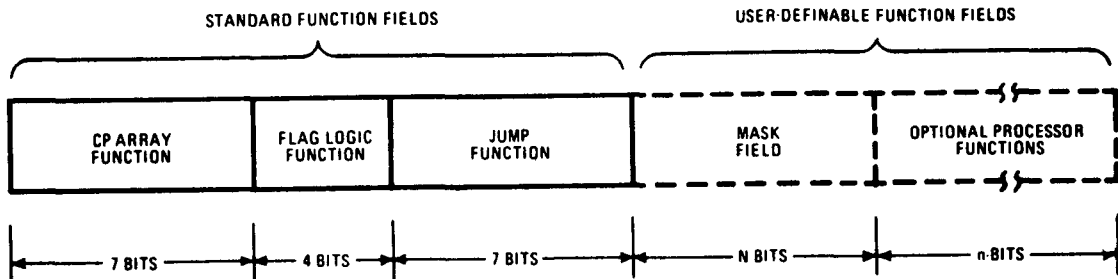


FIGURE 2.2. TYPICAL MICROINSTRUCTION FORMAT

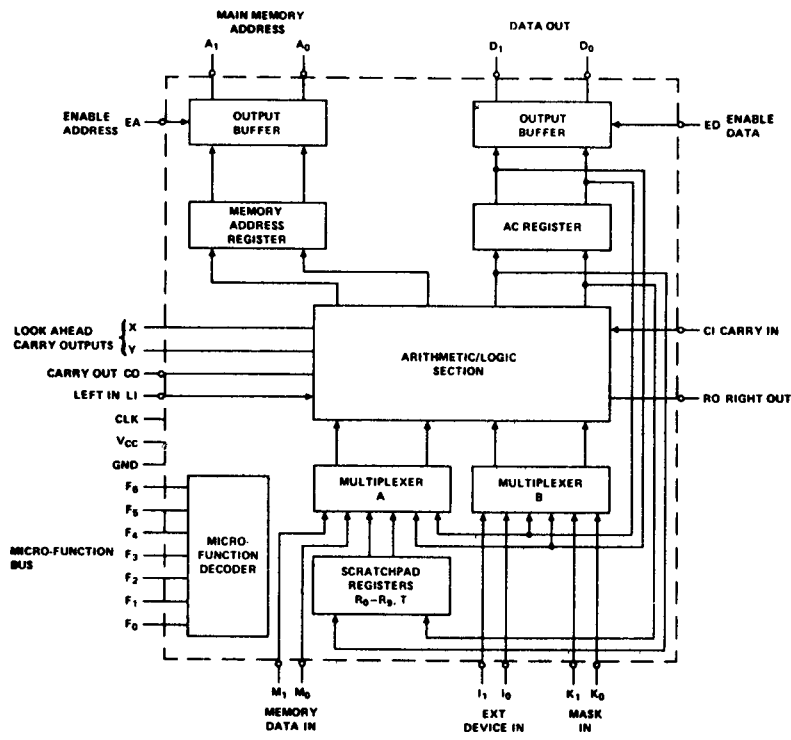


FIGURE 2.3. CPE INTERNAL STRUCTURE

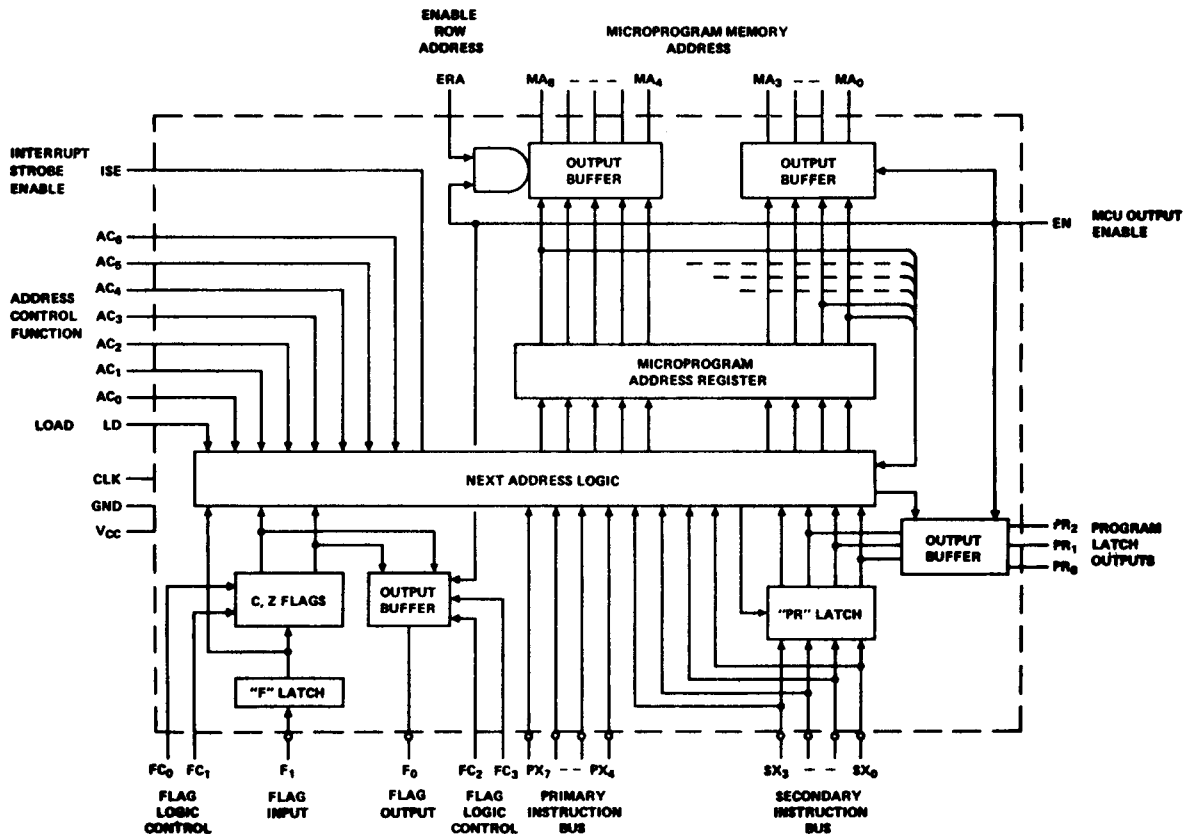


FIGURE 2.4. MCU INTERNAL STRUCTURE

exclusive-NOR.

The M-bus, I-bus, and K-bus provide paths by which data may enter the CPE and be operated upon. The A-bus and D-bus, as mentioned earlier, are the output busses of the CPE.

The K-bus participates in every CPE operations. The K-bus inputs are always ANDed with the B-multiplexer outputs into the ALU. Therefore, all CPE directives given in Table 2.1 either reflect a K-bus value of all ones, which usually selects the data in the accumulator (AC), or all zeroes, which de-selects the data in the accumulator (AC) for a specific operation.

2.3 Sequencer

The 3001 MCU is a complete microprogram sequencer which provides the next microinstruction address. Due to the structure of the MCU, which we will describe in a moment, it is difficult to expand the microprogram addressing capability without adding external registers and paging bits in the K-bus field of the microinstruction.

The MCU is designed to directly address 512 microinstruction locations. This type of addressing capability is adequate for most applications. For example, in less than 512 microinstructions both an 8080 and a PDP-11/40 have been successfully emulated.

2.3.1 Sequencer Directives

To understand the MCU Jump Directives the Microprogram Memory must be conceptualized as a two-dimensional matrix consisting of 32 rows and 16 columns, providing 512 unique microinstruction locations. The location of a microinstruction is thus identified by its row and column address in the Microprogram Memory matrix.

The 9-bit microprogram memory address provided by the MCU specifies the row address in the high order five bits and the column address in the low order four bits. It is, however, possible to implement microprogram memory addresses larger than 512 by adding bits to the microinstruction to indicate which page of 512 microinstructions you are referencing. Of course, this extension to the microinstruction must be saved in an external register and thus becomes a significant overhead.

In a typical configuration, a 7-bit field in the microinstruction, the JUMP field, controls the Jump Function Bus to the MCU. The MCU has the capability of eleven unique jump functions. Each jump function is uniquely coded in the 7-bit Jump field of the microinstruction. From two to five bits of the Jump field select the JUMP function while the remaining bits supply part of the destination address.

The MCU has the capability of testing certain flags and latches (e.g., the G-flag, Z-flag, PR-latch, etc.) which facilitate conditional Jump functions. It must be remembered, that whether or not conditional or unconditional Jumping takes place, for a given location in the microprogram memory matrix and an MCU Jump Function, there is only a fixed subset of microprogram addresses that may be selected as the next microinstruction address.

A summary of the MCU Jump Functions is given in Figure 2.5 along with the proper jump mnemonics.

2.4 Intel 3000 Family

3001	Microprogram Control Unit
3002	Central Processing
3303	Look-ahead Carry Generator
3212	Multimode Latch Buffer
3214	Priority Interrupt Unit

3216	Non-inverting Bidirectional Bus Driver
3226	Inverting Bidirectional Bus Driver
3601	256 X 4-bit PROM
3604	512 X 8-bit PROM
3301	256 X 4-bit ROM
3304	512 X 8-bit ROM

2.5 Advantages

2.5.1 Processing Element

- Over 40 arithmetic and boolean ALU directives
- Dual port structure for increased data through-put
- Huffman Coding Technique which provides multiple functions with a small set of basic directives
- Considerable amount of academic and industrial usage with corresponding literature
- High speed Bipolar Technology

2.5.2 Sequencer

- Small sequencer directives in the microinstruction due to the fixed-jump capability of the MCU
- Conditional and Unconditional branch operations
- Multi-way branch functions
- High speed Bipolar Technology

2.6 Disadvantages

2.6.1 Processing Element

- Limited and non-expandable set of registers
- "Bottle-Neck" register design, Accumulator(AC) 2-bit slice design
- No detection of sign or overflow provided in terms of status lines

2.6.2 Sequencer

- Matrix Conceptualization of program space
- Limited Jump facilities dictate cumbersome and often wasteful usage of control store memory
- Lack of Microprogram Subroutines
- Fixed addressing to 512 microinstructions

2.7 Annotated Bibliography

1. INTEL Corporation, Microprogramming the Series 3000, INTEL Corporation, Santa Clara, California, 1976.

This manual provides the necessary background to begin microprogramming a 3000 series system. The support software, XMAS, and XMAP, is covered in detail and constituted the series 3000 support effort.

2. INTEL Corporation, Series 3000 Reference Manual, INTEL Corporation, Santa Clara, California, 1976.

This manual is partitioned into a section on the 3000 series component family and a section on series 3000 applications. The Applications section covers the design of a disk controller and the design of a 16-bit processor.

3. SIGNETICS Corporation, SIGNETICS 8080 Emulator Manual, SIGNETICS a subsidiary of U.S. Philips Corporation, Sunnyvale, California, 1977.

This manual provides a detailed technical design of an 8080 emulator using the Series 3000 (SIGNETICS Second Source) Chip Set. Both the schematic and microcode are provided in the appendices of the manual.

4. INTEL Corporation, 8080 Emulator (WF8080) Document, INTEL Corporation, Santa Clara, California, 1977.

This document provides a detailed design of an 8080 emulator using the Series 3000 chip set. Note, this design is an 8-bit emulator version as compared with SIGNETICS 16-bit emulator version.

5. SIGNETICS Corporation, "A Guide to the Selection of Support Components for the Series 3000

F-GROUP	R-GROUP	MICRO-FUNCTION
0	I	$R_n + (AC \wedge K) + CI \rightarrow R_n, AC$
	II	$M + (AC \wedge K) + CI \rightarrow AT$
	III	$AT_L \wedge (I_L \wedge K_L) \rightarrow RO$ $LI \vee [(I_H \wedge K_H) \wedge AT_H] \rightarrow AT_H$ $[AT_L \wedge (I_L \wedge K_L)] \vee [AT_H \vee (I_H \wedge K_H)] \rightarrow AT_L$
1	I	$K \vee R_n \rightarrow MAR$ $R_n + K + CI \rightarrow R_n$
	II	$K \vee M \rightarrow MAR$ $M + K + CI \rightarrow AT$
	III	$(\overline{AT} \vee K) + (AT \wedge K) + CI \rightarrow AT$
2	I	$(AC \wedge K) - 1 + CI \rightarrow R_n$
	II	$(AC \wedge K) - 1 + CI \rightarrow AT$
	III	$(I \wedge K) - 1 + CI \rightarrow AT$
3	I	$R_n + (AC \wedge K) + CI \rightarrow R_n$
	II	$M + (AC \wedge K) + CI \rightarrow AT$
	III	$AT + (I \wedge K) + CI \rightarrow AT$
4	I	$CI \vee (R_n \wedge AC \wedge K) \rightarrow CO$ $R_n \wedge (AC \wedge K) \rightarrow R_n$
	II	$CI \vee (M \wedge AC \wedge K) \rightarrow CO$ $M \wedge (AC \wedge K) \rightarrow AT$
	III	$CI \vee (AT \wedge I \wedge K) \rightarrow CO$ $AT \wedge (I \wedge K) \rightarrow AT$
5	I	$CI \vee (R_n \wedge K) \rightarrow CO$ $K \wedge R_n \rightarrow R_n$
	II	$CI \vee (M \wedge K) \rightarrow CO$ $K \wedge M \rightarrow AT$
	III	$CI \vee (AT \wedge K) \rightarrow CO$ $K \wedge AT \rightarrow AT$
6	I	$CI \vee (AC \wedge K) \rightarrow CO$ $R_n \vee (AC \wedge K) \rightarrow R_n$
	II	$CI \vee (AC \wedge K) \rightarrow CO$ $M \vee (AC \wedge K) \rightarrow AT$
	III	$CI \vee (I \wedge K) \rightarrow CO$ $AT \vee (I \wedge K) \rightarrow AT$
7	I	$CI \vee (R_n \wedge AC \wedge K) \rightarrow CO$ $R_n \oplus (AC \wedge K) \rightarrow R_n$
	II	$CI \vee (M \wedge AC \wedge K) \rightarrow CO$ $M \oplus (AC \wedge K) \rightarrow AT$
	III	$CI \vee (AT \wedge I \wedge K) \rightarrow CO$ $AT \oplus (I \wedge K) \rightarrow AT$

NOTES:

- 2's complement arithmetic adds 111 . . . 11 to perform subtraction of 000 . . . 01.
- R_n includes T and AC as source and destination registers in R-group 1 micro-functions.
- Standard arithmetic carry output values are generated in F-group 0, 1, 2 and 3 instructions.

SYMBOL	MEANING
I, K, M	Data on the I, K, and M busses, respectively
CI, LI	Data on the carry input and left input, respectively
CO, RO	Data on the carry output and right output, respectively
R_n	Contents of register n including T and AC (R-Group 1)
AC	Contents of the accumulator
AT	Contents of AC or T, as specified
MAR	Contents of the memory address register
L, H	As subscripts, designate low and high order bit, respectively
+	2's complement addition
-	2's complement subtraction
\wedge	Logical AND
\vee	Logical OR
\oplus	Exclusive-NOR
\rightarrow	Deposit into

TABLE 2.1. CPE FUNCTIONAL DIRECTIVES

The following ten diagrams illustrate the jump set for each of the eleven jump and jump/test functions of the MCU. Location 341, indicated by the black square, represents one current row (row_{21}) and current column (col_5) address. The grey boxes indicate the microprogram locations that may be selected by the particular function as the next address.

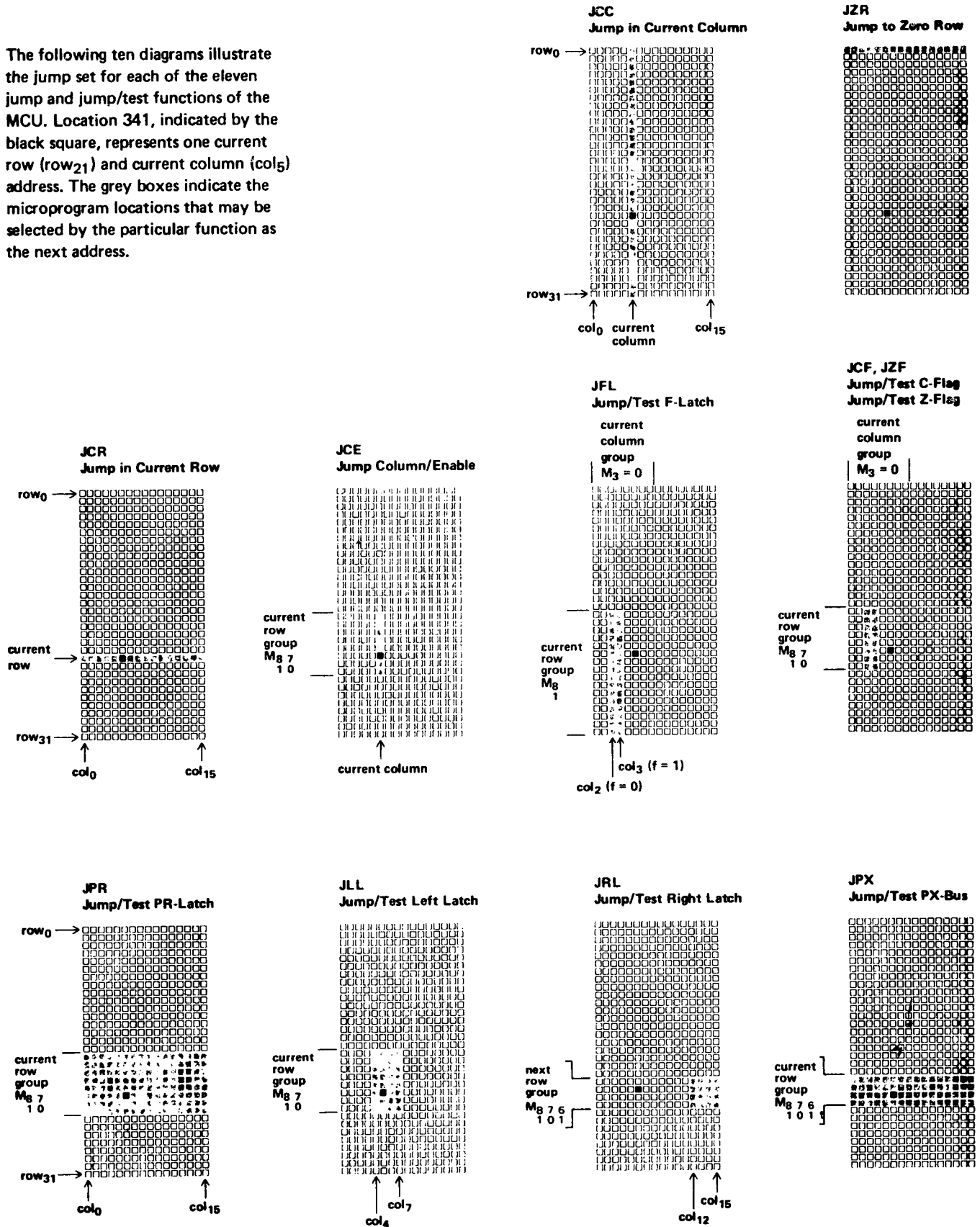


FIGURE 2.5. MCU FUNCTIONAL JUMP DIAGRAMS

Microprocessor-1", SIGNETICS a subsidiary of U.S. Philips Corporation, Sunnyvale, California, 1976.

This document provides a reference to support circuitry required in a Series 3000 design.

6. SIGNETICS Corporation, A User's Guide to the Series 3000 Microprocessor Chip Set-2", SIGNETICS a subsidiary of U.S. Philips Corporation, Sunnyvale, California, 1976.

This document provides a detailed, yet concise, description of the Series 3000 Microprocessor Chip Set. A few design examples are also included.

7. McWilliams, T.M. et al., "Using LSI Processor Bit-Slices to Build A PDP-11: A case Study in Microcomputer Design", Carnegie-Mellon University, Pittsburgh, Pa., 1976.

This document provides a detailed technical design of a PDP-11/40 minicomputer emulator employing the Series 3000 microprocessor chip set. This document covers such things as number of 16-pin equivalent packages required, execution speed, and cost.

8. Maud, Azhar, "Emulation of the Processor for Distributed Processor/Memory System on Intel 3000 Microprocessor Chip Set", Air Force Institute of Technology Air University, Masters Thesis, 1976.

This thesis provides another design using the Series 3000 microprocessor chip set. This emulation is of a military processor and is, therefore, not universally understood by non-military personnel.

Section 3. MONOLITHIC MEMORIES 5701/6701

3.1 Introduction

The MMI 5700/6700 family consists of a sequencer, referred to as a microprogram controller (MPC) - 57110/67110, and a processing element referred to as a micro-controller - 5701/6701. These elements may be connected in a fashion similar to the one depicted in Figure 3.1.

The microinstruction comprises an encoded 8-bit processing element directive, an encoded 10-bit sequencer directive, and various other direct and encoded fields used to control the ancillary hardware components in the system.

Figures 3.2 and 3.3 depict the internal structure of the microcontroller and the MPC respectively.

3.2 Processing Element

The 5701/6701 are vertically complete yet horizontally expandable 4-bit processing elements. With today's LSI Bipolar Technology, this 4-bit slice structure is the largest slice structure which can be economically produced. Utilizing such a 4-bit structure reduces the required CPU components by half when compared with a 2-bit structure.

The 5701/6701 provide all of the functions necessary to design a high-performance micro-programmable computing system. Such features include the following:

- . Full Internal Carry Lookahead
- . ALU Result Shifting Facility
- . True/Complement ALU Sourcing
- . Auxiliary 'Q' Register for Multiply and Divide Implementation
- . Negative, Positive, Zero & Overflow Detection

3.2.1 Register Structure

The 5701/6701 incorporates a 16-word by 4-bit 2-port RAM with two Read Addresses A and B, and

the B Address forming the Write Address. The A and B Address fields are completely independent so that the address field of A can specify any one of the 16 locations in the register file and obtain the data stored in that location on the A output port of the RAM at the same time that the B field is specifying any one of the same 16 locations in the register file and obtain its contents out on the B output port of the RAM. There are no restrictions on addressing in the RAM (register file) - both addresses can simultaneously specify the same address (location) and the contents of that location would be produced on both the A and B output ports of the RAM.

Data is stored in to the RAM (register file) location specified by the B address only. The 'Q' Register can function as an extension to the Accumulator or as an additional register with full shifting capabilities. Normally, the 'Q' Register is used to implement the multiplication and division routines required by most processors.

3.2.2 Processing Element Directives

The Processing Element Directives control the operation of the ALU, the sourcing of the A and B ports to the ALU, and the desired LOAD/SHIFT/DATA OUT Control. Each directive may be in either positive or negative logic. Table 3.1 indicates the various positive logic 5701/6701 directives, while Table 3.2 indicates the various directives in negative logic.

Complete single and double length arithmetic and logical shifting may be performed with various end condition propagation and/or deletions.

3.3 Sequencer

The 57110/67110 Microprogram Controller (MPC) can directly address up to 512 words of micro-program control memory; larger microprogram control memories can be accommodated using a memory paging technique.

One of up to six flag signals can be selected from the 5701/6701 4-bit slice to provide two way conditional branching at every step in a micro-program on the value of the flag or its stored value. Four way conditional branches are also possible by using the on-chip control counter.

The generality of the MPC allows it to be used with a number of different microprogrammable microprocessors other than the 5701/6701. Due to the complexity of the MPC it is possible to use the device in a stand-alone function to simply act as an intelligent controller. In this mode, the MPC would monitor system lines and via a ROM/PROM would generate appropriate system control signals.

3.3.1 Sequencer Directives

Basically, the MPC has the capability of executing 8 unique instructions including the following:

- . Conditional and Unconditional Jumps
- . Conditional and Unconditional Subroutine Jumps to a single level, and
- . Subroutine Returns

To better understand the MPC directives, they will be broken down into Control Options, Shift Options, and Flag Status Control Options. The MPC Control Options are given as 3-bit encoded data items in the microinstruction. These functions are given in Table 3.3. The Shift Options are double length shift operations and are given in Table 3.4. Finally, the flags, C (Carry), N (Negative), V (Overflow), and Z (Zero); can be used in a number of conditional branching operations given in Table

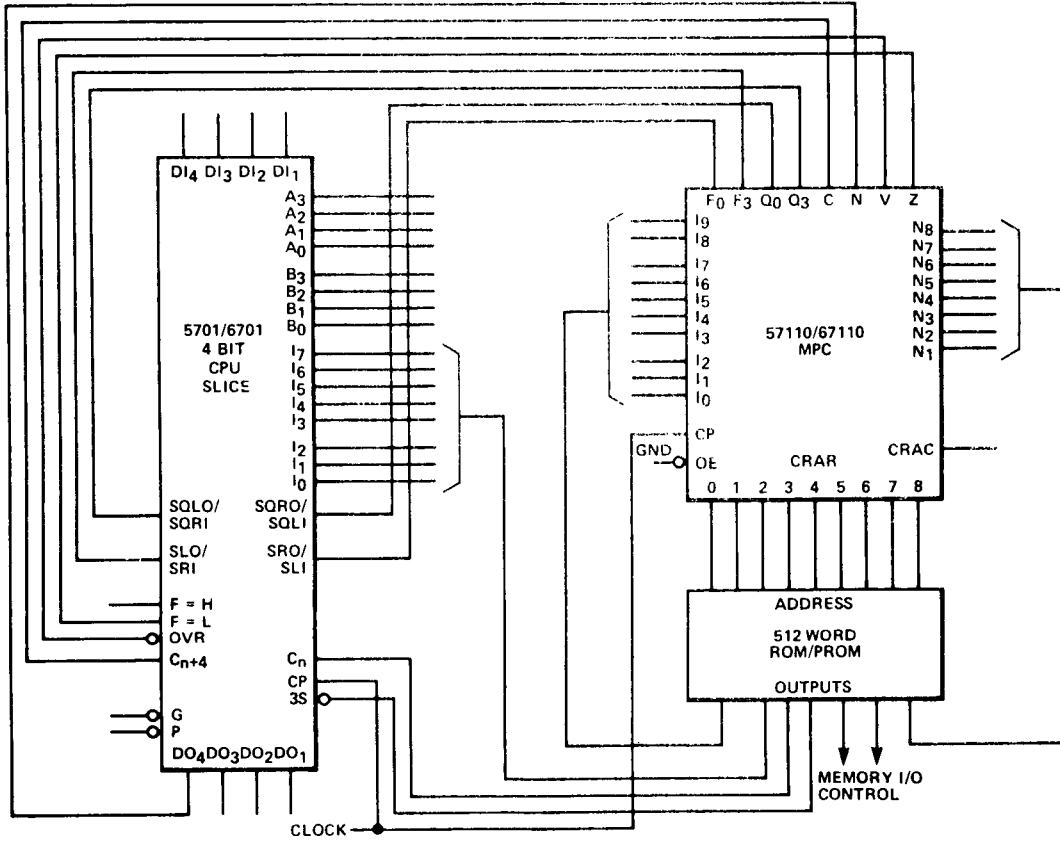


FIGURE 3.1. TYPICAL CONFIGURATION

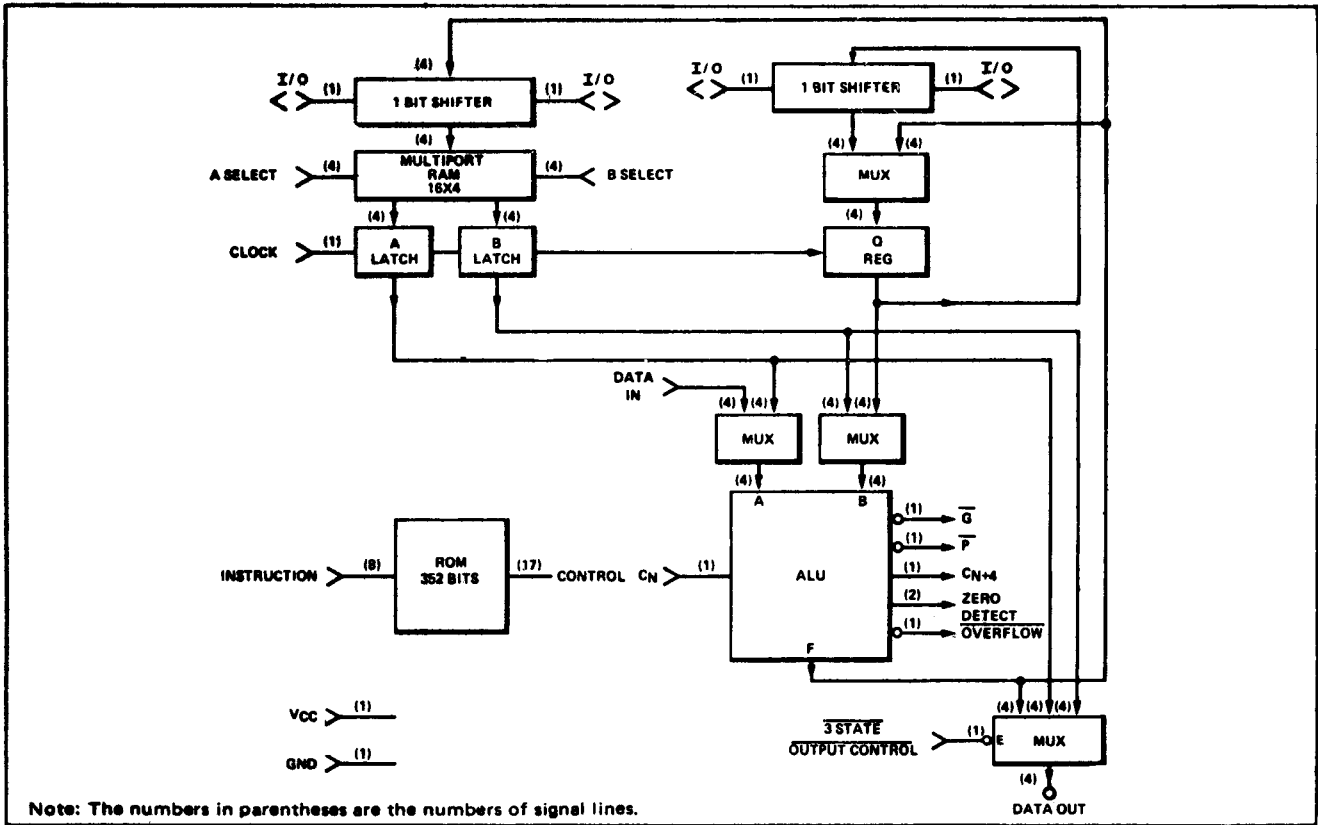


FIGURE 3.2. MICROCONTROLLER INTERNAL STRUCTURE

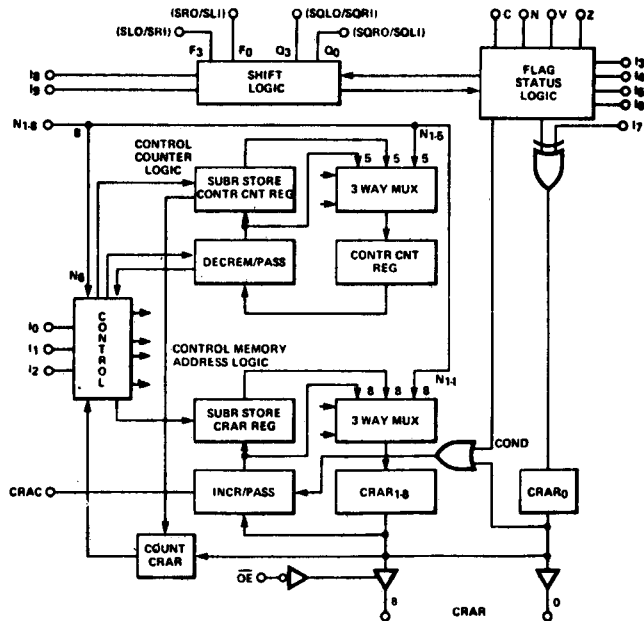


FIGURE 3.3. MPC INTERNAL STRUCTURE

ROM WORD								ALU Instruction	ALU OUTPUT		TYPICAL USES
I_7	I_6	I_5	I_4	I_3	Decimal	Octal	No Carry In ($C_N = L$)		With Carry In ($C_N = H$)		
L	L	L	L	L	0	00	LLLL + HHHH + C_N	Force 1111	Force 0000	Initialization (Force 1s or 0s)	
L	L	L	L	H	1	01	AND A_i & B_j	$A_i \wedge B_j$	$A_i \wedge B_j$	AND A_i & B_j	
L	L	L	H	L	2	02	AND D_i & B_j	$D_i \wedge B_j$	$D_i \wedge B_j$	" D_i & B_j	
L	L	L	H	H	3	03	OR A_i & B_j	$A_i \vee B_j$	$A_i \vee B_j$	OR A_i & B_j	
L	L	H	L	L	4	04	OR D_i & B_j	$D_i \vee B_j$	$D_i \vee B_j$	" D_i & B_j	
L	L	H	L	H	5	05	Exclusive OR A_i & B_j	$A_i \nabla B_j$	$A_i \nabla B_j$	Exclusive OR A_i & B_j	
L	L	H	H	L	6	06	Exclusive OR D_i & B_j	$D_i \nabla B_j$	$D_i \nabla B_j$	" D_i & B_j	
L	L	H	H	H	7	07	$\overline{A_i} + HHHH + C_N$	$\overline{A_i} + 1111$	$\overline{A_i}$	Invert A_i	
L	H	L	L	L	8	10	$\overline{D_i} + HHHH + C_N$	$\overline{D_i} + 1111$	$\overline{D_i}$	" D_i	
L	H	L	L	H	9	11	$\overline{B_j} + HHHH + C_N$	$\overline{B_j} + 1111$	$\overline{B_j}$	" B_j	
L	H	L	H	L	10	12	$\overline{Q} + HHHH + C_N$	$\overline{Q} + 1111$	\overline{Q}	" Q	
L	H	L	H	H	11	13	$\overline{A_i} + LLLL + C_N$	$\overline{A_i}$	$\overline{A_i} + 0001$	2s Complement Of A_i	
L	H	H	L	L	12	14	$\overline{D_i} + LLLL + C_N$	$\overline{D_i}$	$\overline{D_i} + 0001$	" D_i	
L	H	H	L	H	13	15	$\overline{B_j} + LLLL + C_N$	$\overline{B_j}$	$\overline{B_j} + 0001$	" B_j	
L	H	H	H	L	14	16	$\overline{Q} + LLLL + C_N$	\overline{Q}	$\overline{Q} + 0001$	" Q	
L	H	H	H	H	15	17	$A_i + LLLL + C_N$	A_i	$A_i + 0001$	Transfer Or Increment A_i	
H	L	L	L	L	16	20	$D_i + LLLL + C_N$	D_i	$D_i + 0001$	" D_i	
H	L	L	L	H	17	21	$B_j + LLLL + C_N$	B_j	$B_j + 0001$	" B_j	
H	L	L	H	L	18	22	$Q + LLLL + C_N$	Q	$Q + 0001$	" Q	
H	L	L	H	H	19	23	$A_i + HHHH + C_N$	$A_i + 1111$	A_i	Decrement Or Transfer A_i	
H	L	H	L	L	20	24	$D_i + HHHH + C_N$	$D_i + 1111$	D_i	" D_i	
H	L	H	L	H	21	25	$B_j + HHHH + C_N$	$B_j + 1111$	B_j	" B_j	
H	L	H	H	L	22	26	$Q + HHHH + C_N$	$Q + 1111$	Q	" Q	
H	L	H	H	H	23	27	$A_i + B_j + C_N$	$A_i + B_j$	$A_i + B_j + 0001$	Add A_i & B_j	
H	H	L	L	L	24	30	$D_i + B_j + C_N$	$D_i + B_j$	$D_i + B_j + 0001$	" D_i & B_j	
H	H	L	L	H	25	31	$A_i + Q + C_N$	$A_i + Q$	$A_i + Q + 0001$	" A_i & Q	
H	H	L	H	L	26	32	$D_i + Q + C_N$	$D_i + Q$	$D_i + Q + 0001$	" D_i & Q	
H	H	L	H	H	27	33	$A_i + \overline{B_j} + C_N$	$A_i - B_j - 0001$	$A_i - B_j$	Subtract A_i & B_j	
H	H	H	L	L	28	34	$B_j + \overline{A_i} + C_N$	$B_j - A_i - 0001$	$B_j - A_i$	" B_j & A_i	
H	H	H	L	H	29	35	$D_i + \overline{B_j} - C_N$	$D_i - B_j - 0001$	$D_i - B_j$	" D_i & B_j	
H	H	H	H	L	30	36	$B_j + \overline{D_i} + C_N$	$B_j - D_i - 0001$	$B_j - D_i$	" B_j & D_i	
H	H	H	H	H	31	37	$D_i + \overline{Q} + C_N$	$D_i - Q - 0001$	$D_i - Q$	" D_i & Q	

Instruction Modifiers in the 8 x 8 ROM – Positive Logic (1 = H \approx 3V) Interpretation

Rom Word				Load Control		Shift Control			Data Out Control		
I_2	I_1	I_0	Decimal	Load Ram B_j	Load Q	Shift Left	Shift Right	Don't Shift	A Latch	B Latch	ALU Output F
L	L	L	0	X				X			X
L	L	H	1	X				X	X		
L	H	L	2	X				X		X	
L	H	H	3	X		X					X
H	L	L	4	X			X				X
H	L	H	5	X	X	X					X
H	H	L	6	X	X		X				X
H	H	H	7		X			X			X

TABLE 3.1. MICROCONTROLLER FUNCTIONAL DIRECTIVES (POSITIVE LOGIC)

ROM WORD								ALU Instruction	ALU OUTPUT		TYPICAL USES
I ₇	I ₆	I ₅	I ₄	I ₃	Decimal	Octal	No Carry In (C _N = H)		With Carry In (C _N = L)		
L	L	L	L	L	31	37	LLLL + HHHH + C _N	Force 1111	Force 0000	Initialization (Force 1s or 0s)	
L	L	L	L	H	30	36	OR A _i & B _j	A _i ∨ B _j	A _i ∨ B _j	OR A _i & B _j	
L	L	L	H	L	29	35	OR D _i & B _j	D _i ∨ B _j	D _i ∨ B _j	D _i & B _j	
L	L	L	H	H	28	34	AND A _i & B _j	A _i ∧ B _j	A _i ∧ B _j	AND A _i & B _j	
L	L	H	L	L	27	33	AND D _i & B _j	D _i ∧ B _j	D _i ∧ B _j	D _i & B _j	
L	L	H	L	H	26	32	Exclusive NOR A _i & B _j	$\overline{A_i \vee B_j}$	$\overline{A_i \vee B_j}$	Exclusive NOR A _i & B _j	
L	L	H	H	L	25	31	Exclusive NOR D _i & B _j	$\overline{D_i \vee B_j}$	$\overline{D_i \vee B_j}$	D _i & B _j	
L	L	H	H	H	24	30	$\overline{A_i} + HHHH + C_N$	$\overline{A_i}$	$\overline{A_i} + 0001$	2s Complement Of A _i	
L	H	L	L	L	23	27	$\overline{D_i} + HHHH + C_N$	$\overline{D_i}$	$\overline{D_i} + 0001$	D _i	
L	H	L	L	H	22	26	$\overline{B_j} + HHHH + C_N$	$\overline{B_j}$	$\overline{B_j} + 0001$	B _j	
L	H	L	H	L	21	25	$\overline{Q} + HHHH + C_N$	\overline{Q}	$\overline{Q} + 0001$	Q	
L	H	L	H	H	20	24	$\overline{A_i} + LLLL + C_N$	$\overline{A_i} + 1111$	$\overline{A_i}$	Invert A _i	
L	H	H	L	L	19	23	$\overline{D_i} + LLLL + C_N$	$\overline{D_i} + 1111$	$\overline{D_i}$	D _i	
L	H	H	L	H	18	22	$\overline{B_j} + LLLL + C_N$	$\overline{B_j} + 1111$	$\overline{B_j}$	B _j	
L	H	H	H	L	17	21	$\overline{Q} + LLLL + C_N$	$\overline{Q} + 1111$	\overline{Q}	Q	
L	H	H	H	H	16	20	A _i + LLLL + C _N	A _i + 1111	A _i	Decrement Or Transfer A _i	
H	L	L	L	L	15	17	D _i + LLLL + C _N	D _i + 1111	D _i	D _i	
H	L	L	L	H	14	16	B _j + LLLL + C _N	B _j + 1111	B _j	B _j	
H	L	L	H	L	13	15	Q + LLLL + C _N	Q + 1111	Q	Q	
H	L	L	H	H	12	14	A _i + HHHH + C _N	A _i	A _i + 0001	Transfer Or Increment A _i	
H	L	H	L	L	11	13	D _i + HHHH + C _N	D _i	D _i + 0001	D _i	
H	L	H	L	H	10	12	B _j + HHHH + C _N	B _j	B _j + 0001	B _j	
H	L	H	H	L	9	11	Q + HHHH + C _N	Q	Q + 0001	Q	
H	L	H	H	H	8	10	A _i + B _j + C _N	A _i + B _j	A _i + B _j + 0001	Add A _i & B _j	
H	H	L	L	L	7	07	D _i + B _j + C _N	D _i + B _j	D _i + B _j + 0001	D _i & B _j	
H	H	L	L	H	6	06	A _i + Q + C _N	A _i + Q	A _i + Q + 0001	A _i & Q	
H	H	L	H	L	5	05	D _i + Q + C _N	D _i + Q	D _i + Q + 0001	D _i + Q	
H	H	L	H	H	4	04	A _i + $\overline{B_j} + C_N$	A _i - B _j - 0001	A _i - B _j	Subtract A _i & B _j	
H	H	H	L	L	3	03	B _j + $\overline{A_i} + C_N$	B _j - A _i - 0001	B _j - A _i	B _j & A _i	
H	H	H	L	H	2	02	D _i + $\overline{B_j} + C_N$	D _i - B _j - 0001	D _i - B _j	D _i & B _j	
H	H	H	H	L	1	01	B _j + $\overline{D_i} + C_N$	B _j - D _i - 0001	B _j - D _i	B _j & D _i	
H	H	H	H	H	0	00	D _i + $\overline{Q} + C_N$	D _i - Q - 0001	D _i - Q	D _i & Q	

Instruction Modifiers is the 8 x 8 ROM – Negative Logic (1 = L ≈ 0V) Interpretation

Rom Word				Load Control			Shift Control			Data Out Control		
I ₂	I ₁	I ₀	Decimal	Load Ram with Ram Shifter Output	Load Q		Shift Left	Shift Right	Don't Shift	A Latch	B Latch	ALU Output F
					with Q Shifter Output	with ALU Output						
L	L	L	7	X					X			X
L	L	H	6	X					X	X		
L	H	L	5	X					X		X	
L	H	H	4	X			X					X
H	L	L	3	X				X				X
H	L	H	2	X	X		X					X
H	H	L	1	X	X			X				X
H	H	H	0			X			X			X

TABLE 3.2. MICROCONTROLLER FUNCTIONAL DIRECTIVES (NEGATIVE LOGIC)

Control Code			Control Action	Address Field Destination
I ₂	I ₁	I ₀		
0	0	0	Continue to next μ instruction	None
0	0	1	Continue to next μ instruction	Control Counter, Clear SRFF if N ₆ = 1
0	1	0	Jump to next μ instruction if Control Counter \neq 0, Decrement Control Counter	None/CRAR (Cond. Jump)
0	1	1	Subroutine Jump to next μ instruction if Control Counter \neq 0, Decrement Control Counter	None/CRAR (Cond. Subr. Jump)
1	0	0*	Return from Subroutine	None
1	0	1**	Jump to next μ instruction Return from Subroutine when Control Counter Subroutine Latch \equiv CRAR ₀₋₄	CRAR (Jump Subroutine)
1	1	0	Jump to next μ instruction	CRAR (Jump)
1	1	1	Subroutine Jump to next μ instruction	CRAR (Jump Subroutine)

* The MPC will only return to the calling program if it entered a Subroutine via a Subroutine Jump instruction; otherwise it will continue to the next μ instruction in sequence.

** This operation allows the MPC to branch to a section of code, perform the operations outlined by the code and return after a preprogrammed CROM address has been reached or if a return is encountered.

TABLE 3.3. MPC CONTROL OPTIONS

Control Code		Shifting Operation	Bidirectional Shift Lines Acting as Outputs			
I _g	I _g		F ₃ (SLO/SRI)	F ₀ (SRO/SLI)	Q ₃ (SOLO/SQRI)	Q ₀ (SQRO/SQLI)
0	0	Arithmetic Shift Left	—	Q ₃	—	Flag (SC)
0	1	Arithmetic Shift Right	N \oplus V	—	F ₀	—
1	0	Rotate Shift Left	—	F ₃	—	Flag (SC)
1	1	Rotate Shift Right	F ₀	—	F ₀	—

— High Impedance State

SC Contents of Carry Flip Flop

TABLE 3.4. SHIFT OPTIONS

3.5.

A sample microinstruction field which controls the MPC is depicted in Figure 3.4. Note, the required number of bits needed to control the MPC.

3.4 Monolithic Memories Complete Family

57110/67110	Microprogram Controller
5701/6701	4-bit Expandable Microcontroller
5/63125	1024 X 4-bit Registered PROM
5/63105	2048 X 4-bit Registered PROM
5/67401	64 X 4-bit FIFO Serial Memory

3.5 Advantages

3.5.1 Processing Element

- . 4-bit microprocessor-slice design
- . Powerful and Comprehensive set of micro-instructions
- . 16-word register file on board
- . Extension 'Q' register for implementing multiprecision operations
- . Both positive and negative logic operation
- . Carry, Overflow, Negative, and Zero Status lines
- . High Speed Bipolar Technology

3.5.2 Sequencer

- . Classical memory array designed sequencing
- . Microprogram subroutines allowed
- . High speed Bipolar Technology

3.6 Disadvantages

3.6.1 Processing Element

- . Single Port design
- . No register expansion capability
- . Relatively large directives

3.6.2 Sequencer

- . Large directives required
- . Fixed addressing to 512 microinstructions
- . Only single-level subroutine capability

3.7 Annotated Bibliography

1. Ghest, Clive, "5701/6701 4-Bit Expandable Bipolar Microcontroller", Monolithic Memories Incorporated, Sunnyvale, California, 1976.

This Applications Note describes in detail the processing element, 5701/6701, manufactured by MMI. The directives for the 5701/6701 are given as well as specific system designs.

2. Monolithic Memories Incorporated, "57110/67110 Microprogram Controller", Monolithic Memories Incorporated, Sunnyvale, California, 1977.

This document describes the sequencer manufactured by MMI. The directives, timings, and interconnects are described in detail.

3. Ghest, Clive, "A Powerful Microprogram Controller The 67110", Monolithic Memories Incorporated, Sunnyvale, California, 1976.

This document describes the 67110 in a concise manner. Various interconnects are presented and explained in detail.

Section 4. FAIRCHILD MACROLOGIC

4.1 Introduction

The Fairchild Macrologic family consists of a sequencer, referred to as a Microprogram Sequencer 9408, and a processing element, referred to as an Arithmetic Logic Register Stack (ALRS) - 9405. These elements are, typically, 150 to 180 gates per chip complexity which is approximately half of the complexity of other microprogrammable microprocessors. In general, this results in an increase in circuitry needed to accomplish a specific task. For example, the above elements connected in a typical fashion are depicted in Figure 4.1. Notice the added, but necessary, circuitry.

Figure 4.1 may also be referenced to obtain an understanding of what a typical microinstruction format for the Fairchild Macrologic might look like. Figures 4.2 and 4.3 are included to depict the internal structure of the ALRS and Microprogram Sequencer, respectively.

4.2 Processing Element

The Fairchild Macrologic ALRS is a device which may be purchased in a 10 MHz version. The 10 MHz part is the 9405, while the 13 MHz part is suffixed with an 'A', i.e., 9405A. The ALRS is a 4-bit ALU with RAM, 3-state registers, and decode logic to form a 4-bit CPU which is expandable in multiples of 4-bits.

Some of the major characteristics of this device are:

- . High Speed - 10 or 13 MHz
- . Provides for Ripple or Carry Lookahead
- . Implements 64 Microinstructions
- . Provides Zero, Negative, & Overflow Status, and
- . Is Fully TTL Compatible.

4.2.1 Register Structure

The Arithmetic Logic Register Stack (ALRS) is designed to implement accumulators in high performance microprogrammed/microprogrammable computer systems. The device contains a 4-bit arithmetic logic unit (ALU), an 8-word by 4-bit RAM, and associated control logic.

The Address Inputs ($A_0 - A_2$) select one of eight RAM words. These eight RAM locations are used for both source and destination for the ALRS's ALU. Note, no auxiliary or extension registers are provided on the ALRS. The lack of such a register or registers makes the implementation of double-precision operations, such as multiplication and division, somewhat difficult to accomplish. However, considering the gate complexity of the device, it is obvious that the first thing to go was extra registers, and yet, a sufficient number remain to accomplish most tasks.

4.2.2 Processing Element Directives

The ALU implements eight arithmetic and logic functions where one 4-bit operand is supplied from an external source (input data bus) and the second operand is supplied internally from one of the eight RAM words selected by the Address Inputs ($A_0 - A_2$). The result of the operation is loaded into the same RAM location and simultaneously, is loaded into the Output Register making it available at the 3-state output data bus.

The instructions for the ALRS consist of two fields a & l ; $A_0 - A_2$ specify the desired location of the RAM and $l_0 - l_2$ specify the desired function to be performed. Table 4.1 lists all of the processing element directives. In essence, the ALRS provides eight accumulators ($R_0 - R_7$) and eight different operations which may be performed on any of the eight accumulators.

As mentioned earlier, the functions of the ALU affect the Carry Out (C), Negative (N), Overflow (V), and Zero (Z) status.

4.3 Sequencer

The 9408 Microprogram Sequencer controls the order in which microinstructions are fetched from the Control Store Memory. The 9408 contains a 10-bit program counter for directly addressing up to 1024 microinstructions, a 4-level last-in first out (LIFO) stack with appropriate control logic, and all ancillary support logic. The addressing capability of the Microprogram Sequencer can be

Control Code				Action		
I6	I5	I4	I3	Operation	CRAR ₀	Branch
0	0	0	0	None	I7	Unconditional*
0	0	0	1	Store C	I7	Unconditional*
0	0	1	0	Store N, V, Z	I7	Unconditional*
0	0	1	1	Store C, N, V, Z	I7	Unconditional*
0	1	0	0	Shift Flag Register into Q ₀	I7	Unconditional*
0	1	0	1	Shift Flag Register out of Q ₀	I7	Unconditional*
0	1	1	0	Instantaneous value of Q ₀ to CRAR ₀	Q ₀ ⊕ I7	Conditional**
0	1	1	1	Instantaneous value of Q ₃ to CRAR ₀	Q ₃ ⊕ I7	Conditional**
1	0	0	0	Stored value of C to CRAR ₀	SC ⊕ I7	Conditional**
1	0	0	1	Stored value of N to CRAR ₀	SN ⊕ I7	Conditional**
1	0	1	0	Stored value of V to CRAR ₀	SV ⊕ I7	Conditional**
1	0	1	1	Stored value of Z to CRAR ₀	SZ ⊕ I7	Conditional**
1	1	0	0	Instantaneous value of C to CRAR ₀	C ⊕ I7	Conditional**
1	1	0	1	Instantaneous value of N to CRAR ₀	N ⊕ I7	Conditional**
1	1	1	0	Instantaneous value of V to CRAR ₀	V ⊕ I7	Conditional**
1	1	1	1	Instantaneous value of Z to CRAR ₀	Z ⊕ I7	Conditional**

Code bit I7 inverts the status of output line so that the condition is dependent upon \bar{C} , etc. For the first six entries in the table if I7 = 0 there is an unconditional branch to X,0. If I7 = 1 an unconditional branch to X,1.

SC, SN, SV, SZ are Contents of Carry, Sign, Overflow, and Zero Flip Flops.

* Incrementation of CRAR_{1-g} occurs if CRAR₀ = 1.

** Incrementation of CRAR_{1-g} always occurs.

TABLE 3.5. FLAG STATUS CONTROL OPTIONS

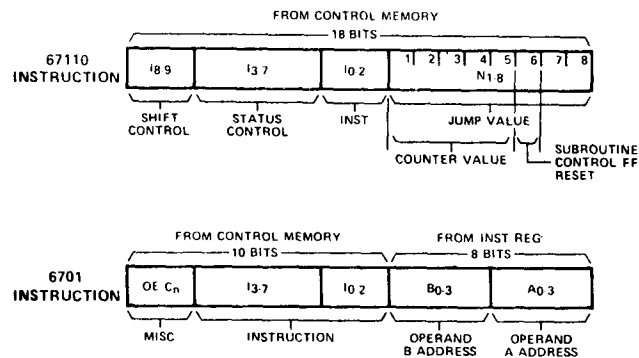


FIGURE 3.4. 5700/6700 TYPICAL MICROINSTRUCTION

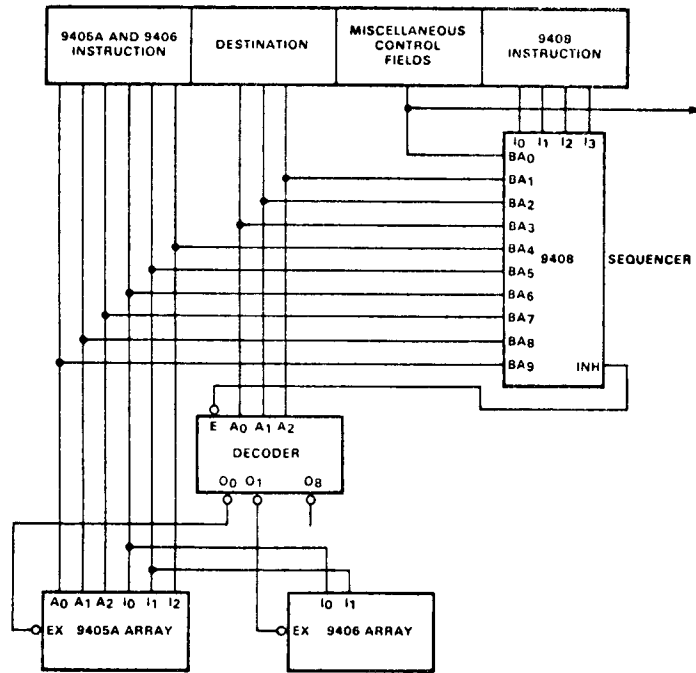


FIGURE 4.1. TYPICAL CONFIGURATION

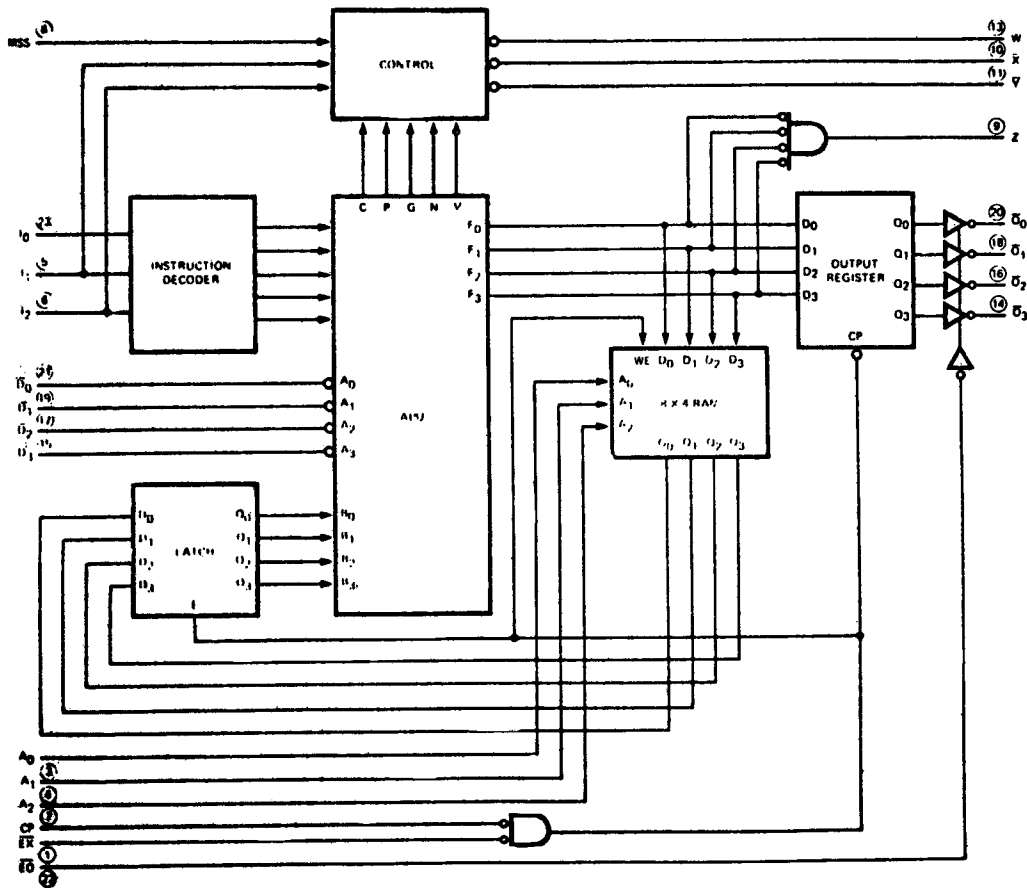


FIGURE 4.2. ALRS INTERNAL STRUCTURE

I_2	I_1	I_0	INTERNAL OPERATION	
L	L	L	$R_x \text{ plus } \overline{D\text{-Bus}} \text{ plus } 1 \rightarrow R_x$	Accumulate
L	L	H	$R_x \text{ plus } \overline{D\text{-Bus}} \rightarrow R_x$	Accumulate
L	H	L	$R_x \cdot \overline{D\text{-Bus}} \rightarrow R_x$	Logical AND
L	H	H	$\overline{D\text{-Bus}} \rightarrow R_x$	Load
H	L	L	$R_x \rightarrow \text{Output Register}$	Output
H	L	H	$R_x + \overline{D\text{-Bus}} \rightarrow R_x$	Logical OR
H	H	L	$R_x \oplus \overline{D\text{-Bus}} \rightarrow R_x$	Exclusive OR
H	H	H	$\overline{D\text{-Bus}} \rightarrow R_x$	Load Complement

NOTES:

1. R_x is the RAM location addressed by A_0 - A_2 .
2. The result of any operation is always loaded into the Output Register.

TABLE 4.1. PROCESSING ELEMENT DIRECTIVES

expanded using a paging technique previously described.

The 9408 has seven test inputs - four of which participate in conditional branching, and three participate in multiway branching. The conditional test lines can be tested individually by appropriate branch instructions. The three multiway test inputs are used to form the three least significant bits of the branch address for a multiway branch. Thus, branching occurs at one of eight unique locations depending on the bit pattern on these three input lines.

The 9408 is designed, as most of the other microprogrammable microprocessors, to operate in either a pipeline or non-pipeline system design.

4.3.1 Sequencer Directives

Basically, the Fairchild Microprogram Sequencer is very similar in functional capability to the capability of the SIGNETICS 8X02. The 9408 has 16 instructions, four flip-flop buffered test inputs for conditional branches, and directives to initiate an 8-way branch.

The 9408 supports subroutine nesting and a multitude of conditional and unconditional transfer directives. To better understand the 9408's capabilities, Table 4.2 is included as a summary of the Microprogram Sequencer directives.

4.4 Fairchild Macrologic Family

- . 9401 CRC Generator/Checker
- . 9403 First-in First-out (FIFO) Buffer Memory
- . 9404 Data Path Switch
- . 9405 Arithmetic Logic Register Stack
- . 9405A Arithmetic Logic Register Stack
- . 9406 Program Stack
- . 9407 Data Access Register
- . 9408 Microprogram Sequencer
- . 9410 Register Stack . 16 X 4 RAM with 3-state Output Register

4.5 Advantages

4.5.1 Processing Element

- . 4-bit microprocessor-slice design
- . On-board register file
- . Zero, Negative, and Overflow Status
- . High Speed Isoplan Injected Integration Logic (I³L) Technology.

4.5.2 Sequencer

- . Direct addressability to 1024 words of control store
- . Unconditional and Conditional branch capability
- . 8-way branch capability
- . Microprogram Subroutining Capability

4.6 Disadvantages

4.6.1 Processing Element

- . Limited and non-expandable set of 8 4-bit registers
- . Single Port design
- . Limited Source and Destination Specification
- . Somewhat less complex than other microprogrammable microprocessors

4.6.2 Sequencer

- . Limited addressing to 1024 words of control store memory

4.7 Annotated Bibliography

1. Fairchild Camera and Instrument Corporation, Macrologic Bipolar Microprocessor Databook, Fairchild Camera and Instrument Corporation, Mountain View, California, 1976.

This book details the complete Fairchild macrologic family of bipolar microprocessors. Sample designs

and interconnects are presented to aid in understanding the product line.

Section 5. Texas Instruments SBP0400

5.1 Introduction

The Texas Instruments SBP0400 is an Integrated Injection Logic (I²L) device. Since Texas Instruments has not announced a companion I²L sequencer for the SBP0400, this section will only discuss the processor element itself and relate interconnection with a 74S482 which will be described in detail in section 7.

The SBP0400 is an I²L 4-bit processing element and by virtue of its I²L technology maintains high-speed and high density on a single chip. The functional gate density on the SBP0400 is greater than 1450 with gate delays of 4-5 ns.

Figure 5.1 illustrates a typical configuration with the 74S482 as the sequencer element. Figure 5.2 illustrates a typical microinstruction used by the SBP0400. A more detailed description of the SBP0400 microinstruction field is given under "Processing Element Directives".

The internal structure of the SBP0400 is depicted in Figure 5.3.

5.2 Processing Element

The SBP0400 is a vertically complete yet horizontally expandable 4-bit I²L processing element. Due to the I²L technology, the SBP0400 is characterized by the following:

- Complexity due to I²L Density
- High-Speed
- Ultra-Low Power, and
- High Noise Margin

Due to the architecture of the SBP0400, certain characteristics are possible. They are:

- Parallel Access to all control, data and address functions
- Dual scaled-shifters with on-chip handling of end conditions
- Versatile Factory Programmable Logic Array (FPLA) Generates on-chip control transformation
- Internal Operation Register and Independent Program Counter Access Control Provide Instruction Look-Ahead Capability (Pipelining)
- Relative Position Control Defines Bit-Slice Rank in N-bit Applications
- Serial and Parallel Access to or From Working Registers
- Word or Byte Incrementation of Program Counter and

ALU Bypass for Direct Register-File Access

However, the most noticeable characteristics of the SBP0400 are derived from its I²L technology. This technology promises to provide the best of both NMOS and BIPOLAR technologies.

5.2.1 Register Structure

The SBP0400 incorporates an 8-word by 4-bit general register file including a program counter with hardware incrementor. Two other registers are provided on the SBP0400 for use in both single and double-length operations. These two extra registers are referred to as the Working Register (WR) and the Extended Working Register (XWR). Each of these two registers is 4-bits wide allowing such routines as signed/unsigned multiply and divide to be implemented with ease.

Since the WR and XWR work in combination double-length operations, it is reasonable to

	MNEMONIC	DEFINITION	I ₃ I ₂ I ₁ I ₀	T ₃ T ₂ T ₁ T ₀	O ₉ O ₈ O ₇ ...O ₂ O ₁ O ₀	VIA ₁ VIA ₀	INH	DESCRIPTION OF OPERATION
Unconditional Branch Instructions	BRV ₀	Branch VIA ₀	L H L L	X X X X	BA ₉ BA ₈ --BA ₁ BA ₀	L L	H	BA ₀ - BA ₉ → PC
	BRV ₁	Branch VIA ₁	L H L H	X X X X	BA ₉ BA ₈ --BA ₁ BA ₀	L H	H	BA ₀ - BA ₉ → PC
	BRV ₂	Branch VIA ₂	L H H L	X X X X	BA ₉ BA ₈ --BA ₁ BA ₀	H L	H	BA ₀ - BA ₉ → PC
	BRV ₃	Branch VIA ₃	L H H H	X X X X	BA ₉ BA ₈ --BA ₁ BA ₀	H H	H	BA ₀ - BA ₉ → PC
	BMW	Branch Multiway	L L H H	X X X X	BA ₉ BA ₃ --MW ₂ MW ₀	L L	H	MW ₀ - MW ₂ , BA ₃ - BA ₉ → PC
	BSR	Branch to Subroutine	L L L H	X X X X	BA ₉ BA ₈ --BA ₁ BA ₀	L L	H	BA ₀ - BA ₉ → PC & Push the Stack
Conditional Branch Instructions	BTH ₀	Branch on T ₀ HIGH	H H L L	X X X H X X X L	BA ₉ BA ₈ --BA ₁ BA ₀ PC+1	L L	H	If Test Register 0 is HIGH: BA ₀ - BA ₉ → PC If Test Register 0 is LOW: PC+1 → PC
	BTH ₁	Branch on T ₁ HIGH	H H L H	X X H X X X L X	BA ₉ BA ₈ --BA ₁ BA ₀ PC+1	L L	H	If Test Register 1 is HIGH: BA ₀ - BA ₉ → PC If Test Register 1 is LOW: PC+1 → PC
	BTH ₂	Branch on T ₂ HIGH	H H H L	X H X X X L X X	BA ₉ BA ₈ --BA ₁ BA ₀ PC+1	L L	H	If Test Register 2 is HIGH: BA ₀ - BA ₉ → PC If Test Register 2 is LOW: PC+1 → PC
	BTH ₃	Branch on T ₃ HIGH	H H H H	H X X X L X X X	BA ₉ BA ₈ --BA ₁ BA ₀ PC+1	L L	H	If Test Register 3 is HIGH: BA ₀ - BA ₉ → PC If Test Register 3 is LOW: PC+1 → PC
	BTL ₀	Branch on T ₀ LOW	H L L L	X X X L X X X H	BA ₉ BA ₈ --BA ₁ BA ₀ PC+1	L L	H	If Test Register 0 is LOW: BA ₀ - BA ₉ → PC If Test Register 0 is HIGH: PC+1 → PC
	BTL ₁	Branch on T ₁ LOW	H L L H	X X L X X X H X	BA ₉ BA ₈ --BA ₁ BA ₀ PC+1	L L	H	If Test Register 1 is LOW: BA ₀ - BA ₉ → PC If Test Register 1 is HIGH: PC+1 → PC
	BTL ₂	Branch on T ₂ LOW	H L H L	X L X X X H X X	BA ₉ BA ₈ --BA ₁ BA ₀ PC+1	L L	H	If Test Register 2 is LOW: BA ₀ - BA ₉ → PC If Test Register 2 is HIGH: PC+1 → PC
	BTL ₃	Branch on T ₃ LOW	H L H H	L X X X H X X X	BA ₉ BA ₈ --BA ₁ BA ₀ PC+1	L L	H	If Test Register 3 is LOW: BA ₀ - BA ₉ → PC If Test Register 3 is HIGH: PC+1 → PC
Miscellaneous Instructions	RTS	Return from Subroutine	L L L L	X X X X	Contents of the Stack Addressed by Read Pointer	L L	L	Pop the Stack
	FTCH	FETCH	L L H L	X X X X	PC+1	L L	L	PC+1 → PC

L - LOW Level
H - HIGH Level
X - Don't Care

TABLE 4.2. MICROPROGRAM SEQUENCER DIRECTIVES

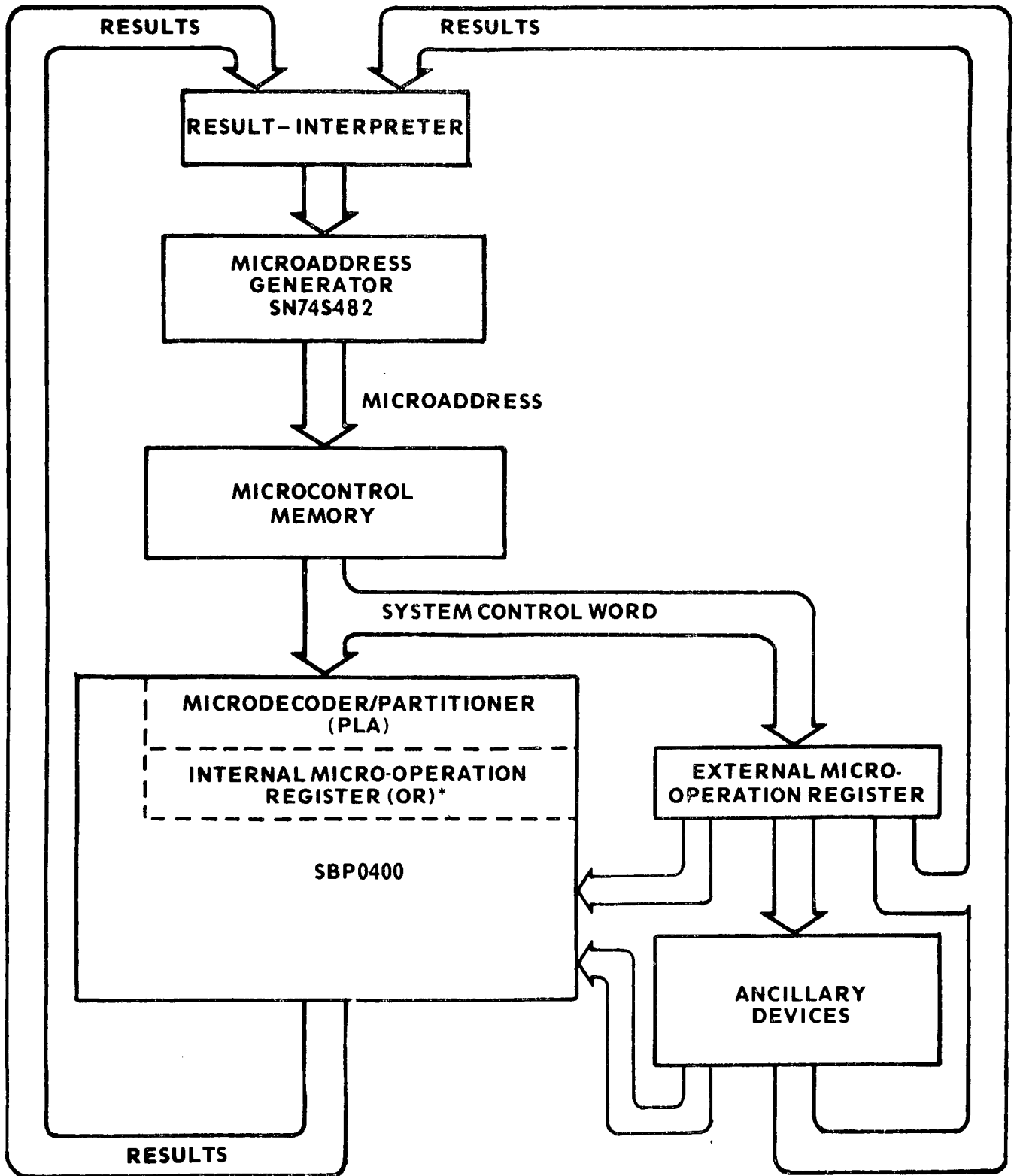


FIGURE 5.1. TYPICAL CONFIGURATION

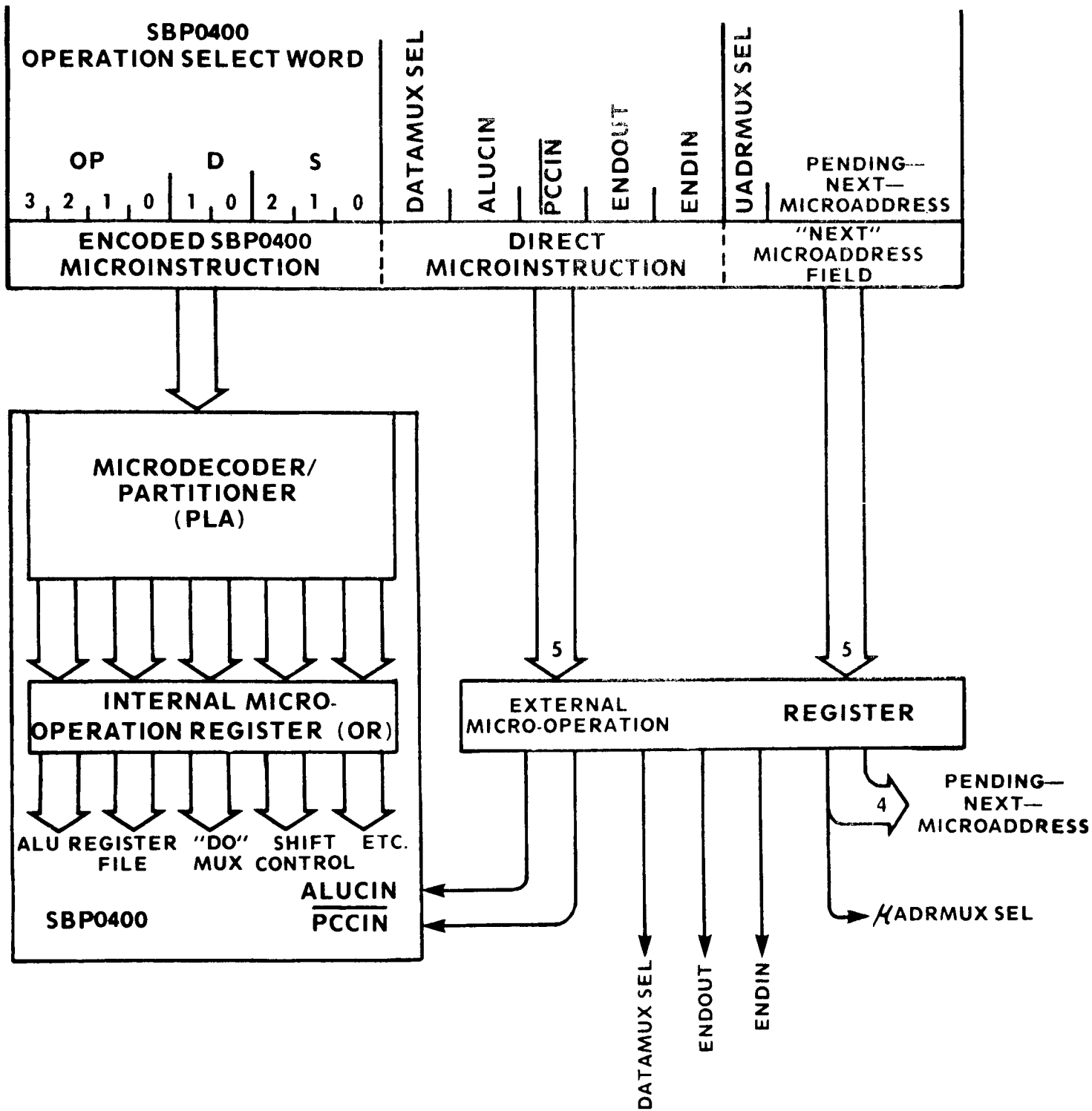


FIGURE 5.2. TYPICAL MICROINSTRUCTION

conceptualize the WR and XWR as a fully functional double-length accumulator (register). Each of the registers in the SBP0400 can function independently and have full operation capability.

5.2.2 Processing Element Directives

The processing element directives for the SBP0400 control multiple operations acting simultaneously within one clock cycle. However, most processing element directives can be functionally separated into data transfers, arithmetic operations, and minor control operations.

The data transfer directives are concerned with enabling data paths to and from SBP0400 resources. The ALU directives control a two-input ALU. Each input to the ALU (A-input, B-input) has a multiplexer attached to it which allows operations to occur on any 2 of 5 possible inputs.

Functionally, the SBP0400 is characterized by its ability to perform, within one clock cycle, any one of its repertoire of 512 standard operations. Of course, the ALU functions are a subset of the instruction repertoire, but provide a powerful 16-operation arithmetic/logic unit (ALU) with symmetrical subtraction and full carry look-ahead capability. The repertoire may be summarized in the following way:

- Operand modifications or combination via eight Arithmetic or eight Boolean Functions on the ALU
- Single or Double-length Arithmetic shifts of single or Double-length binary words
- Single or Double-length logical shifts or circulates
- Single Clock ALU/Shift Combinations to Simplify Implementation of Iterative Multiply and Non-Restore Divide Algorithms, and

Special Select Operations and Data Transfers.

These types of characteristics provide the necessary kernel to emulate a large number of systems with full software compatibility and with no loss of software investment. An added feature is the ability to upgrade or protect proprietary software without the need of custom LSI parts. Table 5.1 indicates the complete SBP0400 processing element directive repertoire, and Figure 5.4 depicts a detailed SBP0400 microinstruction format.

5.3 Advantages

5.3.1 Processing Element

- Complex architecture due to the density provided by the Integrated-Injection Logic (I²L) Technology
- Ultra-low power and low degree of noise susceptibility
- 4-bit microprocessor-slice design
- Powerful and comprehensive set of microinstructions
- On-board register file
- Extension registers provided for multi-precision operations

5.4 Disadvantages

5.4.1 Processing Element

- Single Port Design (Input)
- Non-expandable register file
- Slower than other bipolar slices
- Difficult to master the 512 microinstructions

5.5 Annotated Bibliography

1. Texas Instruments Inc., SBP0400 Applications Manual and Guide to Microprogramming, Texas Instruments, Inc., Dallas, Texas, 1976.

This manual provides a tutorial on microprogramming

for the beginner, a comprehensive description of the SBP0400 microinstructions, and several detailed designs to illustrate the capabilities of the SBP0400.

2. Texas Instruments Inc., "SBP0400 Processor Element Product Specification Manual", Texas Instruments Inc., Houston, Texas, 1977.

This manual provides the detailed specifications on the SBP0400. Waveforms, Timings, and a concise description of the microinstructions are given.

3. Texas Instruments Learning Center, Microprogrammer LCM-1001 Instruction Manual, Texas Instruments Inc., Dallas, Texas, 1976.

This manual is to be used in conjunction with the "Microprogrammer" LCM-1001 as a first course in microprogramming microprogrammable microprocessors. Designs and microprograms are presented in an orderly and understandable way. The complete functional capability of the SBP0400 is presented in a "hands-on" fashion, assuming one has the LCM-1001.

4. Texas Instruments Inc., Bipolar Microcomputer Components Data Book for Design Engineers, Texas Instruments Inc., Dallas, Texas, 1977.

This book describes in detail TI's bipolar microcomputer components and related devices including diagrams, timings, and waveforms. An invaluable tool when working with the SBP0400, SBP0400A, and related devices.

OPERATION	OP. FORM/TYPE	OP3 → OPO OP FIELD	D1D0 D-FIELD	S2 → S0 S-FIELD
dip DOP	III f	HHHH	HH	LHL
DIP DOP	III f	LLLL	HH	LHL
DIP RF	III a	HHHH	HL	LLL → HHH
DIP WR	III s	LHHL	HL	XXX
DIP WR	IIId	LHHL	HH	LHL
DIP XWR	III e	LLLL	HH	LHL
DIP ALU WR DOP	Ic	LLL → HHHH	HH	LLL
DIP ALU WR WR	Id	LLLL → HHHH	HH	LLH
DIP ALU WR → XWR	If	LLLL → HHHH	HH	HII
DIP ALU XWR → DOP	Ih	LLLL → HHHH	HH	HHH
DIP ALU XWR → WR	Ie	LLLL → HHHH	HH	LHH
DIP ALU XWR → XWR	Ig	LLLL → HHHH	HH	HHI
(DIP plus WR plus ALUCIN) LCIR → WR, XWR	IVb	LLLH	HH	LHL
(DIP plus WR plus ALUCIN) RSA → WR, XWR	IVq	HLHH	HH	LHL
DIP plus RF plus ALUCIN → RF	II d	LHHH	HI	LLL → HHH
DIP plus RF plus ALUCIN → WR	II b	LHLL	HL	LLL → HHH
DIP plus RF plus ALUCIN → XWR	II c	LHLH	HL	LLL → HHH
DIP plus WR plus ALUCIN → DOP	III	LHHH	HH	LHL
DIP plus WR plus ALUCIN → XWR	III h	LLHH	HH	LHL
DIP plus XWR plus ALUCIN → WR	III j	HHLL	HH	LHL
DIP plus XWR plus ALUCIN → XWR	III k	HHLH	HH	LHL

OPERATION	OP. FORM/TYPE	OP3 → OPO OP FIELD	D1 D0 D-FIELD	S2 → S0 S-FIELD
RF → DOP	III b	LLLL	HL	LLL → HHH
RF → XWR	III c	LLLH	HL	LLL → HHH
RF ALU WR → RF	Ia	LLLL → HHHH	LL	LLL → HHH
RF ALU WR → WR	Ib	LLLL → HHHH	LH	LLL → HHH
RF plus DIP plus ALUCIN → RF	II d	LHHH	HL	LLL → HHH
RF plus DIP plus ALUCIN → WR, XWR	II b	LHLL	HL	LLL → HHH
RF plus DIP plus ALUCIN → XWR	II c	LHLH	HL	LLL → HHH
(RF plus WR plus ALUCIN) LCIR → WR, XWR	IV d	LLLH	HL	LLL → HHH
(RF plus WR plus ALUCIN) XWR, RSA → WR, XWR	IV i	HLHH	HL	LLL → HHH
RF plus WR plus ALUCIN → XWR	II a	LLHH	HL	LLL → HHH
RF plus XWR plus ALUCIN → WR	II e	HHLL	HL	LLL → HHH
RF plus XWR plus ALUCIN → XWR	II f	HHLH	HL	LLL → HHH

OPERATION	OP FORM/TYPE	OP3 → OPO Op FIELD	D1 D0 D-FIELD	S2 → S0 S-FIELD
XWR ALU DIP → DOP	Ih	LLLL → HHHH	HH	HHH
XWR ALU DIP → WR	Ie	LLLL → HHHH	HH	LHH
XWR ALU DIP → XWR	Ig	LLLL → HHHH	HH	HHL
XWR plus ALUCIN → DOP	III	HHHL	HH	LHL
XWR plus ALUCIN → RF	III j	HHHL	HL	LLL → HHH
XWR plus DIP plus ALUCIN → WR	III k	HHLL	HH	LHL
XWR plus DIP plus ALUCIN → XWR	III e	HHLH	HH	LHL
XWR plus RF plus ALUCIN → WR	III f	HHLL	HL	LLL → HHH
XWR plus RF plus ALUCIN → XWR		HHLH	HL	LLL → HHH

TABLE 5.1. SBP0400 FUNCTIONAL DIRECTIVES

OPERATION	OP. FORM/ TYPE	OP3 → OP0 OP FIELD	D1 D0 D- FIELD	S2 → S0 S-FIELD
WRALU DIP → DOP	lc	LLLL → HHHH	HH	LLL
WR ALU DIP → WR	ld	LLLL → HHHH	HH	LLH
WR ALU DIP → XWR	lf	LLLL → HHHH	HH	HLL
WR ALU RF → RF	la	LLLL → HHHH	LL	LLL → HHH
WR ALU RF → WR	lb	LLLL → HHHH	LH	LLL → HHH
(WR minus DIP minus 1 plus ALUCIN, XWR) LCIR → WR, XWR	IVa	HLLL	HH	LHL
(WR minus DIP minus 1 plus ALUCIN, XWR) rsa → WR, XWR	IVf	LLHL	HH	LHL
(WR minus RF minus 1 plus ALUCIN, xwr) LCIR → WR, XWR	IVc	HLLL	HL	LLL → HHH
(WR minus RF minus 1 plus ALUCIN, XWR) RSA → WR, XWR	IVh	LLHL	HL	LLL → HHH
(WR plus ALUCIN) RSA → WR, XWR	IVe	HLHL	HL	XXX
(WR plus ALUCIN) RSA → WR, XWR	IVe	HLHL	HH	LHL
(WR plus ALUCIN) LCIR → WR	Vd	LLHH	HH	HLH
(WR plus ALUCIN) LCIR → WR	Vd	HLHH	HH	HLH
(WR plus ALUCIN) LSA → WR	Vc	LLHL	HH	HLH
(WR plus ALUCIN) LSL → WR	Vf	HLHL	HH	HLH
(WR plus ALUCIN) RCIR → WR	Vb	LLLH	HH	HLH
(WR plus ALUCIN) RCIR → WR	Vb	HLLH	HH	HLH
(WR plus ALUCIN) RSA → WR	Va	LLLL	HH	HLH
(WR plus ALUCIN) RSL → WR	Ve	HLLL	HH	HLH
(WR plus ALUCIN, XWR) LCIR → (WR, XWR)	Vid	HHHH	HH	HLH
(WR plus ALUCIN, XWR) LCIR → (WR, XWR)	Vid	LHHH	HH	HLH
(WR plus ALUCIN, XWR) LSA → (WR, XWR)	Vlc	LHHL	HH	HLH
(WR plus ALUCIN, XWR) LSL → (WR, XWR)	Vlf	HHHL	HH	HLH
(WR plus ALUCIN, XWR) RCIR → (WR, XWR)	Vlb	HHLH	HH	HLH
(WR plus ALUCIN, XWR) RCIR → (WR, XWR)	Vlb	LHLH	HH	HLH
(WR plus ALUCIN, XWR) RSA → (WR, XWR)	Vla	LHLL	HH	HLH
(WR plus ALUCIN, XWR) RSL → (WR, XWR)	Vle	HHLL	HH	HLH
WR plus DIP plus ALUCIN → DOP	Ili	LHHH	HH	LHL
WR plus DIP plus ALUCIN → XWR	Iih	LLHH	HH	LHL
(WR plus DIP plus ALUCIN) LCIR → WR, XWR	IVb	HLLH	HH	LHL
(WR plus DIP plus ALUCIN) RSA → WR, XWR	IVg	HLHH	HH	LHL
WR plus RF plus ALUCIN → XWR	Ila	LLHH	HL	LLL → HHH
(WR plus RF plus ALUCIN) LCIR → WR, XWR	IVd	HLLH	HL	LLL → HHH
(WR plus RF plus ALUCIN) RSA → WR, XWR	IVI	HLHH	HL	LLL → HHH

TABLE 5.1. CONTINUED

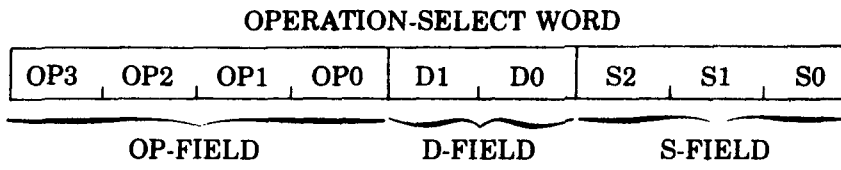


FIGURE 5.4. DETAILED SBPO400 MICROINSTRUCTION