7 Instructions

The instruction set gives the programmer the ability to carry out all the functions necessary to program the system efficiently and may be divided into ten basic groups:

Load/Store Instructions
Arithmetic Instructions
Logical Instructions
Character Instructions
Branch Instructions
Shift Instructions
Control Instructions
Input/Output Instructions
External Transfer Instructions
Move Table Instructions

Within these groups efficiency is ensured by the possible use of up to eight different methods of forming one of the instruction's operands, the method to be used being chosen by the programmer with reference to the memory and timing requirements of any particular program.

Two formats for instruction layouts are used and where necessary two words are used to define an instruction.

INSTRUCTION FORMATS

Two instruction formats are possible and these are defined within the instruction by the most significant bit, bit 0, of the instruction word. Where instructions consist of two words the format bit is the most significant bit of the first word only.

Format 0 instructions are always short, that is one word. Format 1 instructions may be short or long, one or two words.

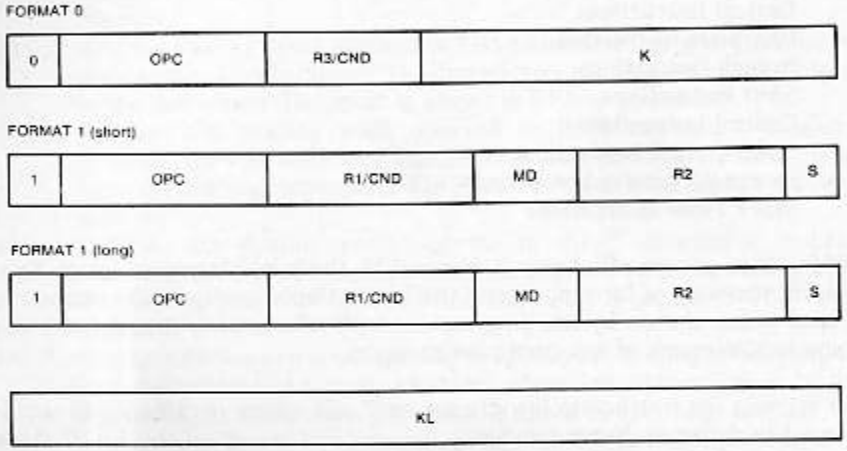

Figure 7.1 Layout of instruction formats.

OPC 4 bits, the pattern of which defines the instruction to be carried out.

R1 4 bits, specifies the working register to be used by the instruction, A0 - A15. It may contain one of the operands to be used and may also be used to hold the result of the instruction. In certain cases with R1 = 0 the addressed register, the P register, will not be used and in these cases R1 = 0 will qualify the operation code and define a different instruction than when R1 = 0.

R2 4 bits, specifies the second working register to be used by the instruction A1 - A15. It may contain the second operand or hold an address to be used in forming this operand. If R2 is made zero, no second working register is specified but this condition is used in deciding the method of forming the second operand.

R3 3 bits, specifies the working register to be used by the instruction A0 - A7. It may contain one of the operands to be used and may also be used to hold the result of the instruction. In certain cases with R3 = 0 the addressed register, the P register, will not be used and in these cases R3 = 0 will qualify the operation code and define a different instruction than when R3 = 0.

CND 3 bits, specifies the condition which must exist for a particular instruction to be carried out. Used to qualify conditional branch instructions and replaces R3 or the most significant 3 bits of R1.

MD 2 bits, specifies the mode of addressing to be used when forming the second operand of an instruction where this is applicable.

S 1 bit, applicable to certain instructions using memory. When present it specifies that the result of the instruction concerned is to be stored in the memory address specified by the instruction. When this bit is not present the result is placed into the working register specified by R1.

K 8 bits, these bits are used to specify the operand in format 0 instructions, and include short constant operands (k) and short displacements (m-for relative branch instructions). This field is also used to specify counts for shift instructions (n) and device addresses to I/O instructions (dev), in these cases a part of the field may be used to qualify the operation code.

KL 16 bits, this field is made up of the complete second word of a double length instruction and may specify a long constant (KL) or an address (m).

FORMING THE OPERAND

Many of the instructions may use various methods of forming one of the operands to be used. In all, eight methods of forming an operand are available governed by the values of the Format, Mode, and R2 fields of the instruction layout.

Figure 7.2 lists the eight methods of forming an operand and a brief description of each method is given following the figure.

Type	Format	Mode	R2	
T1	1	00	–	Reg/Reg.
T2	1	01	R2 = 0	Long Constant
T3	1	01	R2 ≠ 0	Address in Reg R2
T4	1	10	R2 = 0	Address in next word
T5	1	10	R2 ≠ 0	Indexed
T6	1	11	R2 = 0	Indirect
T7	1	–	R2 ≠ 0	Indexed Indirect
T8	0	–	–	Short Constant

Figure 7.2

T1. Register/Register - Format 1 (short)
The operand is the value in the register specified by R2 of the instruction format.

T2. Long Constant - Format 1 (long)
The operand is the value in the least significant word, all sixteen bits, of the double length instruction format.

T3. Address in Register - Format 1 (short)
The operand is held in memory. The memory address of the operand is the value in the register specified by R2 of the instruction format.

T4. Address in Next Word - Format 1 (long)
The operand is held in memory. The memory address of the operand is the value in the least significant word of the double length instruction.

T5. Indexed Address in Next Word - Format 1 (long)
The operand is held in memory. The memory address of the operand is found by adding the value in the register specified by R2 of the instruction format to the value in the least significant word of the double length instruction.

T6. Indirect Address in Next Word - Format 1 (long)
The operand is held in memory. The memory address of the operand is also held in memory. This indirect address is the value in the least significant word of the double length instruction.

T7. Indexed Indirect Address in Next Word - Format 1 (long)
The operand is held in memory. The memory address of the operand is also held in memory. This indirect address is found by adding the value in the register specified by R2 of the instruction format to the value in the least significant word of the double length instruction.

T8. Short Constant - Format 0
The operand is the value in the least significant eight bits of the instruction format.

INSTRUCTION TIMING

The timing of the instructions depends on various factors: the type of instruction itself, the memory, the method of forming the operand and the number of memory cycles required.
The instruction set offers the possibility of very rapid execution times where single word register/register or short constant operations are employed whilst the more complex register/memory instructions save execution time when compared with the routines they may replace.
Execution time is also reduced in the case of conditional instructions by carrying out the conditional check immediately after accessing the instruction and then only continuing if the required conditions are satisfied.

TRAP ACTION

The use of any invalid instruction causes the activation of the Trap action which consists of the following basic actions:
- the CPU does not attempt to carry out the instruction
- information with reference to the instruction address and processor status is saved in the stack
- interrupts are inhibited
- a user mode flag is reset when working in user mode
- an indirect branch is made to address /7E for a trap routine.

no RIT, automatically deactivated

THE INSTRUCTION SET

The instructions within the basic groups, together with their mnemonic, addressing type(s) and the execution time for the different types of memory are listed here:

Load/Store Instructions		Addressing types	Execution times in µs	
			1.2 µs memory	0.7 µs memory
LD	Load	T4 - T7	3.7 - 5.0 µs	2.2 - 3.0 µs
LDR	Load Register	T1, T3	1.4 - 2.5 µs	1.2 - 1.8 µs
LDK	Load Constant	T8, T2	1.3 - 2.5 µs	0.9 - 1.5 µs
ST	Store	T4 - T7	3.8 - 5 µs	2.4 - 3.3 µs
STR	Store Register	T3	2.8 µs	2.1 µs
ML	Multiple Load	T4 - T7	2.8 - 4.1 + nx1.3 µs	2.6 - 3.5 + nx0.8 µs
MLR	Multiple Load Register	T3	2.0 - 2.4 + nx1.3 µs	1.9 - 2.3 + nx0.8 µs
MLK	Multiple Load Constant	T2	2.9 + nx1.3 µs	2.7 + nx0.8 µs
MS	Multiple Store	T4 - T7	2.8 - 4.1 + nx1.3 µs	2.6 - 3.5 + nx0.8 µs
MSR	Multiple Store Register	T3	2.5 - 3.1 + nx1.3 µs	2.3 - 2.9 + nx0.8 µs
EL	Extended Load (MMU)	T4 - T7	3 - 4.1 µs	2.4 - 3.3 µs
ELR	Extended Load Register (MMU)	T3	2.5 µs	2.1 µs
ES	Extended Store (MMU)	T4 - T7	3 - 4.1 µs	2.4 - 3.3 µs
ESR	Extended Store Register (MMU)	T3	2.5 µs	2.1 µs
Arithmetic Instructions				
AD	Add	T4 - T7	3.8 - 6.3 µs	2.2 - 3.9 µs
ADR	Add Register	T1, T3	1.4 - 3.8 µs	1.2 - 2.6 µs
ADK	Add Constant	T8, T2	1.3 - 2.5 µs	0.9 - 1.5 µs
SU	Subtract	T4 - T7	3.8 - 6.3 µs	2.2 - 3.9 µs
SUR	Subtract Register	T1, T3	1.4 - 3.8 µs	1.2 - 2.6 µs
SUK	Subtract Constant	T8, T2	1.3 - 2.5 µs	0.9 - 1.5 µs
MU	Multiply	T4 - T7	9.7 - 11 µs	8.6 - 9.5 µs
MUR	Multiply Register	T1, T3	7.8 - 8.5 µs	7.6 - 8.9 µs
MUK	Multiply Constant	T2	8.5 µs	7.9 µs
DV	Divide	T4 - T7	10 - 11.3 µs	8.8 - 9.5 µs
DVR	Divide Register	T1, T3	7.8 - 8.8 µs	7.6 - 8.9 µs
DVK	Divide Constant	T2	8.8 µs	8.2 µs

		Addressing types	Execution times in µs	
			1.2 µs memory	0.7 µs memory
DA	Double Add	T4 - T7	5.6 - 6.9 µs	3.9 - 4.8 µs
DAR	Double Add Register	T1, T3	3.1 - 4.5 µs	3.0 - 3.6 µs
DAK	Double Add Constant	T2	4.4 µs	3.2 µs
DS	Double Subtract	T4 - T7	5.6 - 6.9 µs	3.9 - 4.8 µs
DSR	Double Subtract Register	T1, T3	3.1 - 4.5 µs	3.0 - 3.6 µs
DSK	Double Subtract Constant	T2	4.4 µs	3.2 µs
C2	Two's Complement	T4 - T7	5.3 - 6.5 µs	3.5 - 4.4 µs
C2R	Two's Complement Register	T3	4.0 µs	3.1 µs
IM	Increment Memory	T4 - T7	5.0 - 6.3 µs	3.0 - 4.0 µs
IMR	Increment Memory Register	T3	3.8 µs	2.6 µs
NGR	Negate Register	T1	2.0 µs	1.9 µs
CM	Clear Memory	T4 - T7	3.8 - 5.0 µs	2.4 - 3.3 µs
CMR	Clear Memory Register	T3	2.8 µs	2.1 µs
CW	Compare Word	T4 - T7	3.8 - 5.0 µs	2.2 - 3.0 µs
CWR	Compare Word Register	T1, T3	1.4 - 2.5 µs	1.2 - 1.8 µs
CWK	Compare Word Constant	T2	2.5 µs	1.5 µs
FFL	Integer to Floating Point	T1		3.7 µs
FFX	Floating Point to Integer	T1		5.1 µs
FAD	Floating Point Addition	T4-T7		6.2 - 9.6 µs
FADR	Floating Point Addition/Register	T3		5.9 - 8.4 µs
FSU	Floating Point Subtract	T4-T7		6.2 - 9.6 µs
FSUR	Floating Point Subtract/Register	T3		5.9 - 8.4 µs
FMU	Floating Point Multiply	T4-T7		8.8 - 12.2 µs
FMUR	Floating Point Multiply/Register	T3		8.4 - 11.0 µs

		Addressing types	Execution times in µs	
			1.2 µs memory	0.7 µs memory
FDV	Floating Point Division	T4-T7		8.8 - 12.2 µs
FDVR	Floating Point Division/Register	T3		8.4 - 11.0 µs
Logical Instructions				
AN	Log. AND	T4 - T7	3.8 - 6.3 µs	2.2 - 3.9 µs
ANR	Log. AND Register	T1, T3	1.4 - 3.8 µs	1.2 - 2.6 µs
ANK	Log. AND Constant	T8, T2	1.3 - 2.5 µs	0.9 - 1.5 µs
OR	Log. OR	T4 - T7	3.8 - 6.3 µs	2.2 - 3.9 µs
ORR	Log. OR Register	T1, T3	1.4 - 3.8 µs	1.2 - 2.6 µs
ORK	Log. OR Constant	T8, T2	1.3 - 2.5 µs	0.9 - 1.5 µs
XR	Exclusive OR	T4 - T7	3.8 - 6.3 µs	2.2 - 3.9 µs
XRR	Ex. OR Register	T1, T3	1.4 - 3.8 µs	1.2 - 2.6 µs
XRK	Ex. OR Constant	T8, T2	1.3 - 2.5 µs	0.9 - 1.5 µs
TM	Test Mask	T1	1.4 µs	1.2 µs
TNM	Test Not Mask	T1	1.4 µs	1.2 µs
Cl	One's Complement	T4 - T7	3.8 - 6.3 µs	2.2 - 3.9 µs
ClR	One's Complement Register	T1, T3	1.4 - 3.8 µs	1.2 - 2.6 µs
Character Handling Instructions				
LC	Load Character	T4 - T7	3.8 - 5.0 µs	2.7 - 3.6 µs
LCR	Load Character Register	T3	2.8 µs	2.4 µs
LCK	Load Character Constant	T2	2.8 µs	2.4 µs
SC	Store Character	T4 - T7	3.8 - 5.0 µs	2.4 - 3.3 µs
SCR	Store Character Register	T3	2.8 µs	2.1 µs
CC	Compare Character	T4 - T7	3.8 - 5.0 µs	2.7 - 3.6 µs
CCR	Compare Char. Register	T3	2.8 µs	2.3 µs
CCK	Compare Char. Constant	T2	2.8 µs	2.3 µs
ECR	Exchange Char. Register	T1	1.4 µs	1.2 µs
Branch Instructions				
AB	Absolute Branch	T8, T2	1.3 - 2.1 µs	0.9 - 2.0 µs
ABR	Absolute Branch Register	T1, T3	1.6 - 2.6 µs	1.2 - 2.4 µs
ABI	Absolute Branch	T4 - T7	4.0 - 5.2 µs	1.1 - 3.2 µs

		Addressing types	Execution times in µs	
			1.2 µs memory	0.7 µs memory
RB	Relative Backward Branch	T8	1.3 µs	1.1 µs
RF	Relative Forward Branch	T8	1.3 µs	1.1 µs
CF	Call Function	T2	4.8 µs	4.0 µs
CFR	Call Function Register	T1, T3	4.2 - 4.9 µs	3.6 - 4.1 µs
CFI	Call Function	T4 - T7	5.5 - 6.6 µs	4.5 - 5.4 µs
RTN	Return	T3	3 - 4.4 µs	2.7 - 4.1 µs
EX	Execute	T4 - T7	depends on instruction in operand	
EXR	Execute Register	T1, T3		
EXK	Execute Constant	T2		
Shift Instructions				
SLA	Left Arithmetic Shift	T8	2.0 + nx0.3 µs	1.9 + nx0.3 µs
SRA	Right Arithmetic Shift	T8	1.9 + nx0.3 µs	1.7 + nx0.3 µs
SLL	Left Logical Shift	T8	1.9 + nx0.3 µs	1.7 + nx0.3 µs
SRL	Right Logical Shift	T8	1.8 + nx0.3 µs	1.6 + nx0.3 µs
SLC	Left Circular Shift	T8	1.9 + nx0.3 µs	1.7 + nx0.3 µs
SRC	Right Circular Shift	T8	1.8 + nx0.3 µs	1.6 + nx0.3 µs
SLN	Left Shift and Normalize	T8	4.2 + nx0.5 µs	4.0 + nx0.5 µs
SRN	Right Shift and Normalize	T8	4.1 + nx0.5 µs	3.9 + nx0.5 µs
DLA	Double Left Arith Shift	T8	3.1 + nx0.3 µs	3.0 + nx0.3 µs
DRA	Double Right Arith Shift	T8	3.1 + nx0.3 µs	3.0 + nx0.3 µs
DLL	Double Left Log. Shift	T8	2.4 + nx0.3 µs	2.2 + nx0.3 µs
DRL	Double Right Log. Shift	T8	2.4 + nx0.3 µs	2.2 + nx0.3 µs
DLC	Double Left Circular Shift	T8	2.4 + nx0.3 µs	2.2 + nx0.3 µs
DRC	Double Right Circ. Shift	T8	2.4 + nx0.3 µs	2.2 + nx0.3 µs
DLN	Double Left and Norm Shift	T8	4.5 + nx0.5 µs	4.3 + nx0.5 µs
DRN	Double Right and Norm Shift	T8	4.9 + nx0.5 µs	4.7 + nx0.5 µs
Control Instructions				
ENB	Enable	T8	3.5 µs	3.4 µs
HLT	Halt	T8	1.7 µs	1.6 µs
RIT	Reset Internal Interrupt	T8	1.7 µs	1.6 µs

		Addressing types	Execution times in μs	
			1.2 μs memory	0.7 μs memory
INH	Inhibit Interrupt	T8	1.7 μs	1.6 μs
LKM	Link To Monitor	T8	3.5 μs	3.4 μs
SMD	Set Mode	T8	1.7 μs	1.6 μs
Input/Output Instructions				
CIO	Control Input/Output	T8	4.4 μs	4.3 μs
INR	Input to Register	T8	5.3 μs	5.2 μs
OTR	Output from Register	T8	4.4 μs	4.3 μs
SST	Send Status	T8	5.3 μs	5.2 μs
TST	Test Status	T8	5.3 μs	5.2 μs
External Transfer Instructions				
WER	Write External Register	T8	4.6 μs	4.5 μs
RER	Read External Register	T8	5.1 μs	5.0 μs
TL	Segment Table Load (MMU)	T4 - T7	15.4 - 16.8 μs	12 - 13 μs
TLR	Segment Table Load Register (MMU)	T3	15.1 μ	11.7 μs
TS	Segment Table Store (MMU)	T4 - T7	15.4 - 16.8 μs	12 - 13 μs
TSR	Segment Table Store Register (MMU)	T3	15.1 μs	11.7 μs
FLD	Floating Point Load	T4-T7		4.4 - 5.3 µ
FLDR	Floating Point Load/Register	T3		4.1 µs
FST	Floating Point Store	T4-T7		3.7 - 4.6 µs
FSTR	Floating Point Store/Register	T3		3.4 µs
Move Table Instructions				
MVF	Move Table Forward (P857 standard)	T8	4.7 + nx2 μs	4.5 + nx1.8 μs
MVB	Move Table Backward (P857 standard)	T8	4.3 + nx2 μs	4.1 + nx1.8 μs
MVUS	Move Table from User to System (MMU)	T8	4.3 + nx2 μs	4.1 + nx1.8 μs
MVSU	Move Table from System to User (MMU)	T8	4.7 + nx2 μs	4.5 + nx1.8 μs

8 Data Flow and Control

The control of data flow within the system is governed by the action which is being carried out at the time. The main sources of control being the instruction set, the input/output processors, the interrupt system, and the bus control system. Data flow within the system is carried out via the general purpose bus. The input/output processors use conventional control circuitry whilst the control exercised by the instruction set, the interrupt system and the bus controller are via a microprogram held permanently within the control ROM of the CPU.

The following examples of data flow cover only the instruction set. The data flow and control of the input/output processors is covered later in chapter 10. As all the instructions are controlled in a generally similar manner only one instruction, an add instruction, is shown.

Figures 8.1 and 8.2 show a flowchart of the microprogram actions carried out during an add instruction which places the result in a register. The required microprogram instruction words would be accessed in sequence from the address generated by the ROM address generator. Three separate actions take place to carry out the complete operation:

1. The instruction is accessed from memory using the address in S REG and P REG are then incremented by 2 in preparation for the next action.
2. The method of forming the operand is decided. The operand is accessed and placed into REG M and Q.
3. The arithmetic action is carried out and the result placed into the specified register. The Condition Register is updated.
   At the same time the next instruction is fetched and the registers P and S are incremented by 2.

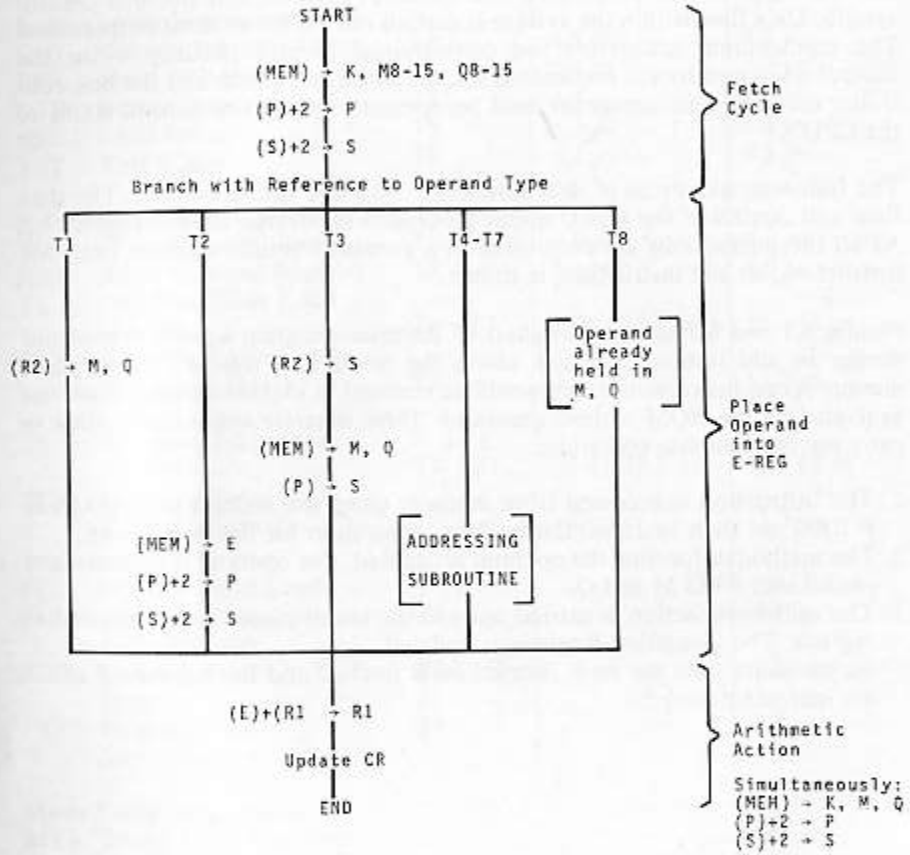

Figure 8.1 Instruction Microprogram

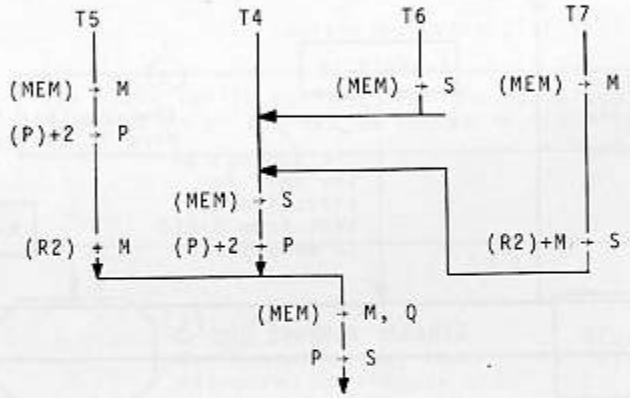

Figure 8.2 Microprogram Addressing Routine

Figures 8.3 to 8.6 show dia matically the data flow involved in the basic arithmetic operations, with r ect to the overall system block diagram on page 2-3.

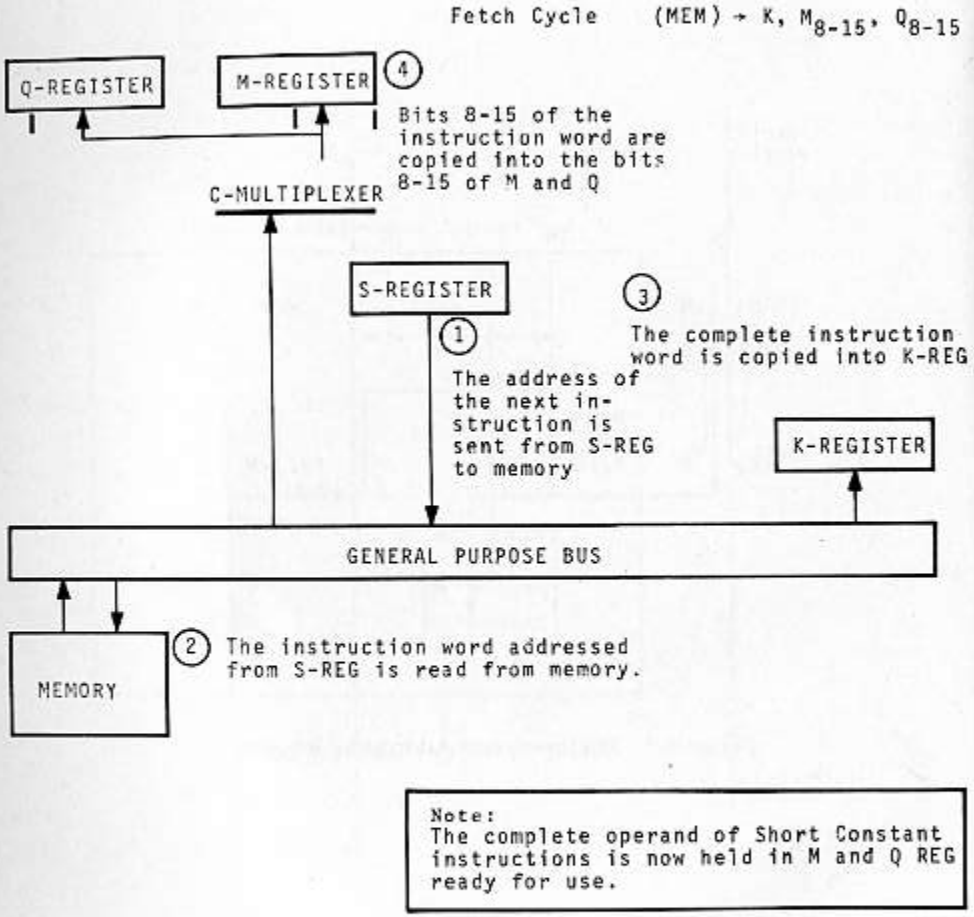

Figure 8.3 Accessing an Instruction

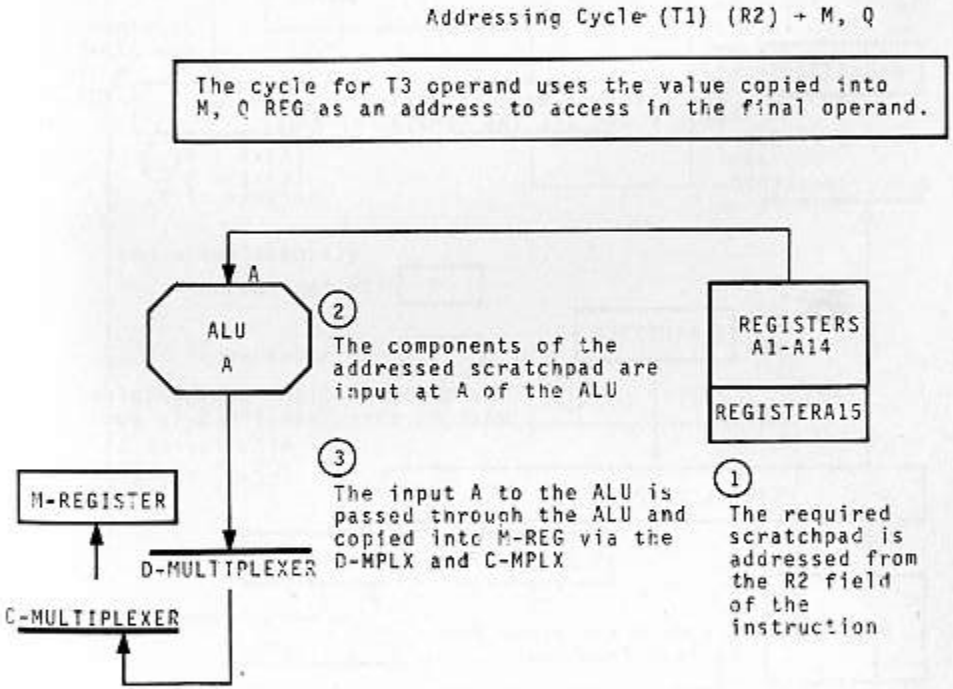

Figure 8.4 Addressing cycle (T1)