Advanced Micro Devices

AmZ8000 Family Reference Manual

Principles of Operation AmZ8001/2 Processor Instruction Set

Copyright © 1979 by Advanced Micro Devices, Inc.

Material in this document has been generated from information supplied by Zilog, Inc.
Advanced Micro Devices, Inc. reserves the right to modify information contained in this document without notice.

AM-PUB086

PREFACE

The present state of MOS LSI semiconductor technology has permitted very powerful and complex general purpose processors to be economically incorporated into a single silicon chip. This capability ushers in a new era of system design, where for the first time low cost tools are available for solving many complex problems. Significant levels of computing power are now available inexpensively and can be used both to lower the cost of high performance systems and to improve the efficiency of programmers in their increasingly more complex tasks.

The AmZ8000 family is the first processor family to fully exploit this new era, breaking tradition with the legacy of compromised performance dictated by past manufacturing technologies. The two processors in the family incorporate many of the features heuristically evolved from both minicomputer and main frame systems. This gives the applications programmer, the systems programmer and the system designer the power and flexibility required for today's complex systems.

This document describes the Processor Instruction Set in detail. The descriptions have been arranged with one instruction per page for completeness and for easy reference. This approach has been found to be suitable for both hardware designers and for programmers. There is no intention to be concise, but instead to provide users with complete, detailed, easy-to-understand descriptions of all the processor instructions.

The information in this document will later be updated and incorporated as the Instruction Chapter in a forthcoming AmZ8000 family reference manual. This document is one of several in support of the AmZ8000 family.

CONTENTS

PROCESSOR ORGANIZATION

PROCESSOR ORGANIZATION

Introduction

The AmZ8001 and AmZ8002 are initial members of the AmZ8000 sixteen bit microprocessor family. These central processing units (CPUs) are software compatible and hence, unless otherwise indicated, information contained in this document applies to both. The AmZ8001 handles 23-bit addresses giving it 8 Megabyte (8,388,608 bytes) addressing capability. Memory associated with the AmZ8001 system is considered to consist of 128 segments with 64 Kilobytes (65,536 bytes) per segment. Thus, the AmZ8001 is also known as the segmented version. On the other hand, the AmZ8002 has 16-bit (64 Kilobyte) addressing capability and is also known as the non-segmented version.

Register Structure

The CPUs are centered around sixteen 16-bit general purpose registers identified as RØ through R15. The desired register is usually designated by a four bit field in an instruction. In general, the instructions operate on byte (8-bit), word (16-bit), or long word (32-bit) operands. For byte operations, the first eight general purpose registers (RØ through R7) are treated as sixteen 8-bit registers identified as RLØ, RHØ, RL1 and so on to RL7 and RH7. A four bit field in an instruction designates the desired byte register. For operations requiring long-words, the 16-bit general purpose registers are grouped in pairs. For example, the RØ, R1 pair is identified as RRØ, the R2, R3 pair as RR2 and so on to the R14, R15 pair as RR14. Thus, the general purpose registers can also be treated as eight 32-bit registers. The three most significant bits of a 4-bit field in an instruction designate the desired register pair and the fourth bit should be zero. For certain 64-bit operands, the general purpose registers can also be grouped in quads. For example, the RØ, R1, R2 and R3 group is identified as RQØ, the R4, R5, R6 and R7 group as RQ4 and so on to the R12, R13, R14 and R15 group as RQ12. The two most significant bits of a four bit field in an instruction designate the desired quad register and the remaining two bits should be zero. Figure 1 depicts the AmZ8001 register structure and Figure 2 shows the AmZ8002 register structure. Table 1 is a summary of register addressing in byte mode and Table 2 is a summary for 16-bit, 32-bit and 64-bit modes.

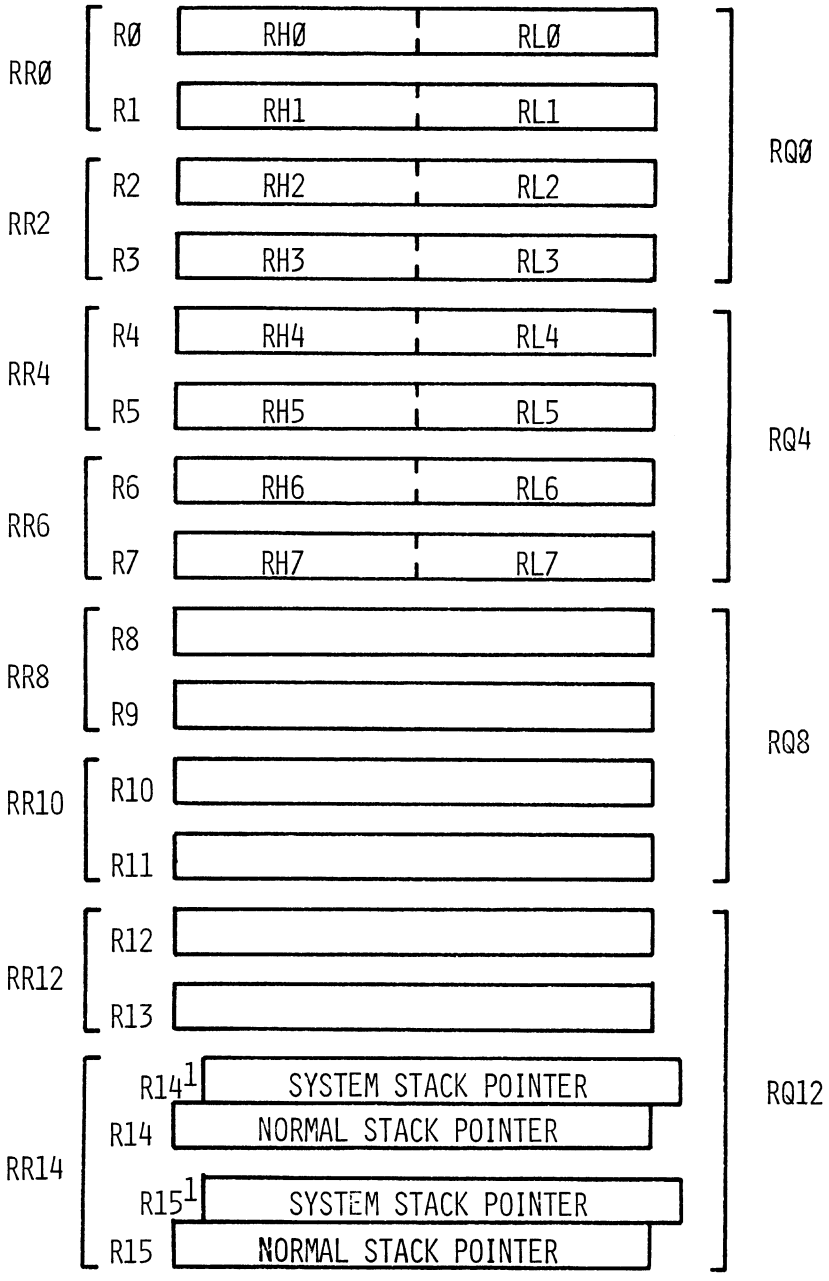


FIGURE 1 AMZ8001 REGISTERS

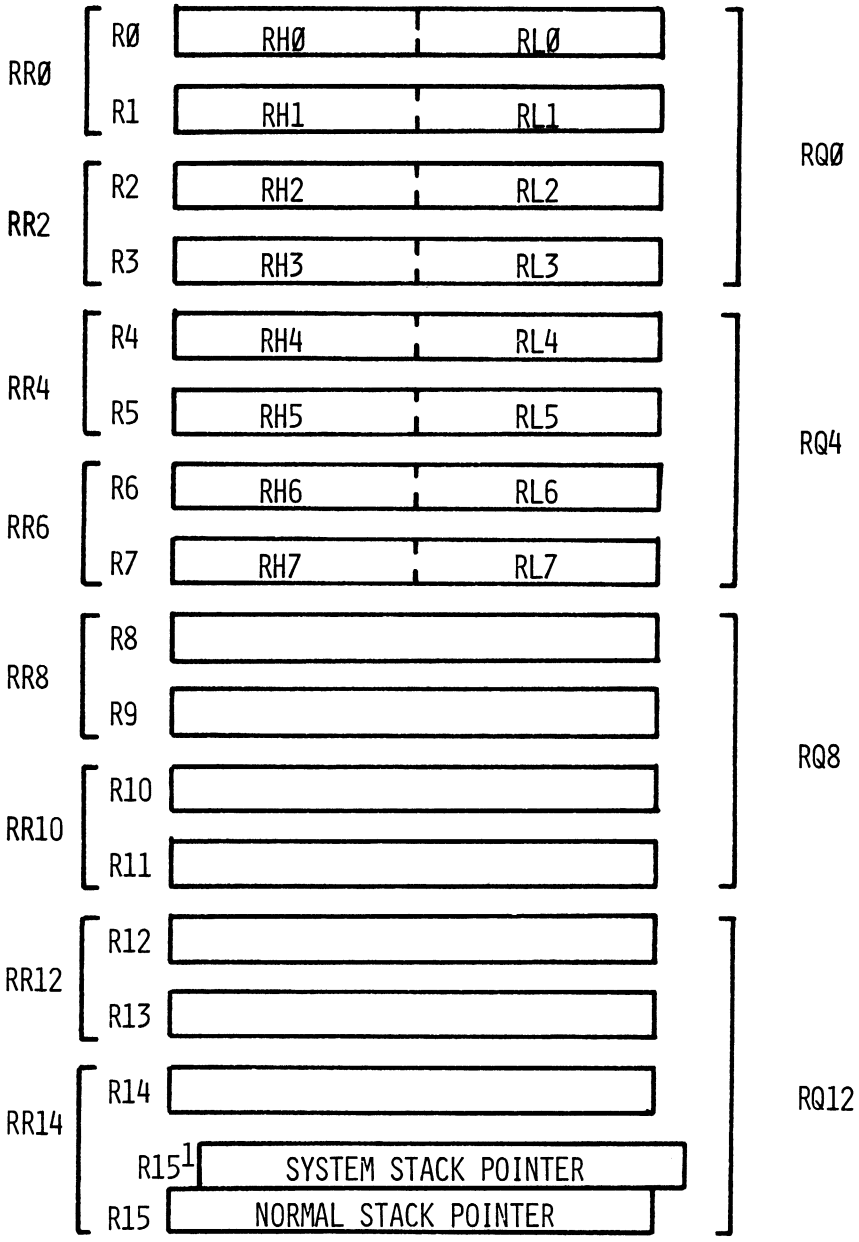


FIGURE 2 AMZ8002 REGISTERS

Designation Field				8-Bit Mode
Ø	Ø	Ø	Ø	RHØ
Ø	Ø	Ø	1	RH1
Ø	Ø	1	Ø	RH2
Ø	Ø	1	1	RH3
Ø	1	Ø	Ø	RH4
Ø	1	Ø	1	RH5
Ø	1	1	Ø	RH6
Ø	1	1	1	RH7
1	Ø	Ø	Ø	RLØ
1	Ø	Ø	1	RL1
1	Ø	1	Ø	RL2
1	Ø	1	1	RL3
1	1	Ø	Ø	RL4
1	1	Ø	1	RL5
1	1	1	Ø	RL6
1	1	1	1	RL7

Table 1. Byte Mode Register Addressing

Designation Field				16-BIT Mode	32-BIT Mode	64-BIT Mode
0	0	0	0	R0	RR0	RQ0
0	0	0	1	R1		
0	0	1	0	R2	RR2	
0	0	1	1	R3		
0	1	0	0	R4	RR4	RQ4
0	1	0	1	R5		
0	1	1	0	R6	RR6	
0	1	1	1	R7		
1	0	0	0	R8	RR8	RQ8
1	0	0	1	R9		
1	0	1	0	R10	RR10	
1	0	1	1	R11		
1	1	0	0	R12	RR12	RQ12
1	1	0	1	R13		
1	1	1	0	R14	RR14	
1	1	1	1	R15		

Table 2. Register Addressing

The registers may contain operands or address information. When a register pair contains a long-word operand, the even numbered register of the pair holds the most significant 16-bit while the odd numbered register of the pair holds the least significant 16-bits. When a register quad is specified for 64-bit data, the first register holds the most significant 16-bits and the last register of the quad holds the least significant 16-bits. For example, RØ is the first register and R3 is the last register of the quad RQØ, R4 is the first and R7 is the last of the quad RQ4 and so on.

In AmZ8001 a register pair will be needed to specify the required 23-bit address. The 7-bit segment number is always specified in the even numbered register and 16-bit offset is specified in the odd numbered register of the pair.

Stack Pointer

The architecture allows the creation and maintenance of stacks in the memory. Any of the general purpose registers (except RRØ in AmZ8001 and RØ in AmZ8002) can be designated as a stack pointer in the PUSH and POP instructions. However, for the CALL and RETURN instructions, specific general purpose registers are implied as stack pointers.

In the AmZ8001, the general purpose register pair RR14 is the implied stack pointer. The seven bit segment number is contained in R14 and R15 contains the 16-bit offset value. The segment number together with the offset value forms a 23-bit segmented address. For the AmZ8002, the general purpose register R15 is the implied stack pointer and contains the required 16-bit address. It should be remembered that the implied stack pointers are still general purpose registers. In other words, certain implied general purpose registers are given stack pointer attributes in addition to their normal general purpose characteristics.

The processors can operate in one of two selectable modes: SYSTEM and NORMAL. The SYSTEM mode is sometimes called supervisor or privileged

mode and the NORMAL mode is sometimes known as problem or non-privileged mode. Separation of system and normal stacks is a desirable in order to facilitate sophisticated system designs. This is accomplished by providing SYSTEM STACK POINTER in addition to NORMAL STACK POINTER.

In the AmZ8001 two additional registers $R14^1$ and $R15^1$ are provided corresponding to R14 and R15. When AmZ8001 is operating in the SYSTEM mode, $R14^1$ will be used as the general purpose register whenever R14 is specified. Similarly $R15^1$ will be used instead of R15 in the SYSTEM mode for both AmZ8001 and AmZ8002. Thus, the register pair $R14^1$, $R15^1$ (identified as $RR14^1$) is the implied SYSTEM STACK POINTER for the AmZ8001 and $R15^1$ is the implied SYSTEM STACK POINTER for the AmZ8002. Although R14 and R15 are not used in the SYSTEM mode, instructions are provided such that these two general purpose registers can be accessed without actually switching the operating mode. The SYSTEM STACK POINTER will be used during program interruptions to save the pre-interrupt status irrespective of the selected operating mode.

Program Counter

The CPU operation is controlled by instructions fetched from the memory. The address for instruction fetch is supplied by the PROGRAM COUNTER (PC). Figure 3 shows the AmZ8001 program counter. It consists of two words, seven bits of the first word are used to specify the segment number and 16-bit offset is specified in the second word. The segment number designates one of 128 segments and the 16-bit offset designates a memory location in that segment. The instructions are always word aligned and the PC is incremented by multiples of 2 to fetch instructions from sequential memory locations. It should be noted that incrementing the offset cannot affect the segment number. In other words, carry from offset will not propagate into the segment number of the PC; instead the offset counter will simply wrap around. Figure 4 shows the AmZ8002 PC format consisting of 16 bits. Except for the absence of the segment number portion PC operation of the AmZ8001 and AmZ8002 are identical. When reset, the AmZ8001 PC SEG will be automatically loaded from memory address 4 and PC OFFSET will be automatically loaded from address 6. AmZ8002 PC will be automatically loaded from memory address 2 upon reset. All these memory addresses are located in segment Ø.

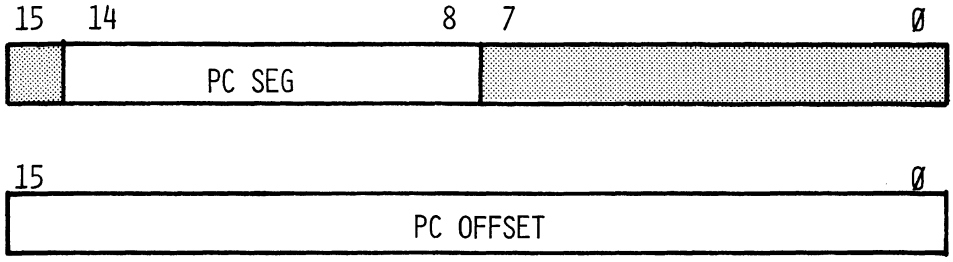


FIGURE 3 AmZ8001 PROGRAM COUNTER

FIGURE 4 AmZ8002 PROGRAM COUNTER

Processor Status Information

The contents of the program counter and FLAG AND CONTROL WORD (FCW) are collectively called the Processor Status Information. The FCW of the AmZ8001 is shown in Figure 5. The most significant byte contains five control bits - Segmentation Enable (SEG), Normal/System (N/S), Stop Enable (SE), Vectored Interrupt Enable (VIE) and Non-Vectored Interrupt Enable (NVIE). The least significant byte contains six CPU flag bits - Carry (C), Zero (Z), Sign (S), Parity/Overflow (P/V), Decimal Adjust (DA), and Half Carry (H). The remaining bits are reserved for future expansion. The FCW for the AmZ8002 is shown in Figure 6. It is identical to Figure 5 except that the SEG bit is not available in the AmZ8002.

The flag portion of the FCW contains processor flags necessary to characterize the results from data manipulation operations. The half carry (H) flag is affected during arithmetic operations involving byte operands. A byte consists of two 4-bit digits. The H flag is used to indicate occurrence of a carry from the least significant digit into the most significant digit.

The Decimal Adjust (DA) flag is provided to facilitate conversion operations required to accomplish BCD arithmetic. The Parity/Overflow (P/V) is used to indicate parity of the result following certain non-arithmetic operations and occurrence of overflow condition following arithmetic operations. If both operands participating in an add operation have the same sign and the result has the opposite sign, an overflow has occurred. Subtraction is addition with the 2's complement of the subtrahend.

The CPU uses two's complement representation of numbers and hence the most significant bit is considered to be the sign bit. A "one" in the most significant bit position represents a negative number. The Sign reflects the state of the most significant bit position of a result after an arithmetic or logic instruction.

The Zero (Z) flag when set to 1 indicates that all bits (including sign) of a result are zero. In general, this flag is affected for both arithmetic and logic instructions.

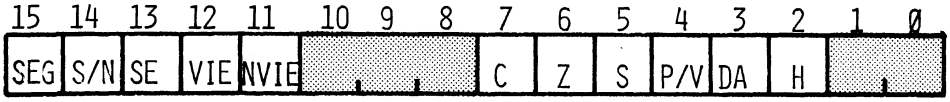


FIGURE 5 AMZ8001 FLAG AND CONTROL WORD

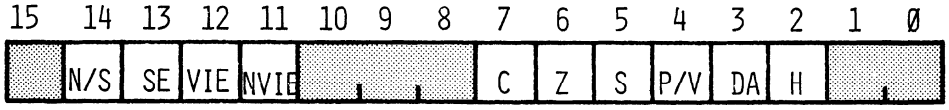


FIGURE 6 AMZ8002 FLAG AND CONTROL WORD

The Carry (C) flag is used to indicate occurrence of a carry from the most significant bit position after an arithmetic instruction. By convention, bits are numbered starting from zero, hence bit 7 (eighth bit) is the most significant for bytes, bit 15 (sixteenth bit) is the most significant for words and bit 31 (thirty-second bit) is the most significant for long words.

The CPU can handle three types of external interrupts: non-maskable, non-vectored and vectored. The NVIE and VIE bits control the latter two. The non-maskable interrupt cannot be disabled in the CPU. On the other hand, vectored and non-vectored interrupts can be disabled by clearing the corresponding FCW bit to Ø. In other words, an interrupt enable bit must be 1 before the corresponding external interrupt signal will be recognized. The non-maskable interrupt has the highest priority while vectored interrupt has the next lower and non-vectored has the lowest priority.

The Stop Enable (SE) bit is provided to facilitate stopping the processor after executing a single instruction. There is an external input to the processor that participates in this operation. When the SE bit is 1, the external input signal will be honored.

The N/S bit determines the operating mode; logical "one" indicates SYSTEM and logical "zero" indicates NORMAL. In the System mode, all instructions are valid and are executed. In the Normal mode, only those instructions are valid that cannot be used to affect the system integrity. The instructions that are not valid in the Normal mode are called System or Privileged instructions. The System instructions include those that inspect or modify the control bits of the FCW, those that participate in interprocessor communication and those that perform input/output operations. If a system instruction is encountered while operating in the Normal mode, the instruction execution is suppressed and a trap will be generated to cause program interruption.

The AmZ8001 deals with 23-bit segmented addresses whereas AmZ8002 uses 16-bit non-segmented addressing. The SEG bit in the FCW allows the AmZ8001 to operate in the non-segmented mode. When this bit is 1, the CPU is operating in the segmented mode. This bit will always be zero in the AmZ8002.

The most significant byte of the FCW contains the control bits. Hence any instruction that operates on these bits is a privileged instruction. Significance of the FCW bits is summarized in Table 3.

When reset, the FCW in AmZ8001 will be automatically loaded from memory location 2 (segment Ø) and the FCW in AmZ8002 will be automatically loaded from location Ø.

New Program Status Area Pointer

When a program interruption occurs, the CPU automatically saves the program status in the system stack. The program status consists of the processor status information and information relating to the reason for interruption called Identifier. After storing the pre-interrupt program status, new program status will be loaded into the FCW and PC. This new program status is obtained from pre-determined locations in the memory called New Program Status Area designated by the NEW PROGRAM STATUS AREA POINTER (NPSAP). The NPSAP in AmZ8001 is shown in Figure 7. It consists of two 8-bit registers, one for the 7-bit segment number and the other for the most significant eight bits of the offset. On the other hand, only one 8-bit register is used in the AmZ8002 as shown in Figure 8. This register specifies the most significant 8-bits of the 16-bit address. Access to the NPSAP is by using the LDCTL instruction.

Refresh Counter

Both AmZ8001 and AmZ8002 contain a refresh counter to facilitate dynamic memory system implementations. The refresh counter is illustrated in Figure 9 consisting of a 9-bit binary ROW COUNTER and 6-bit binary RATE COUNTER and a REFRESH ENABLE (RE) bit. The RATE COUNTER is a programmable modulo 64 counter clocked at 25% of the frequency of the clock driving the CPU. The ROW COUNTER is clocked whenever the RATE COUNTER overflows. The

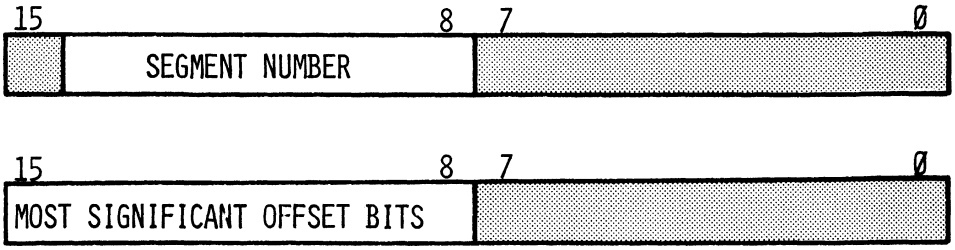


FIGURE 7 AmZ8001 NEW PROGRAM STATUS AREA POINTER

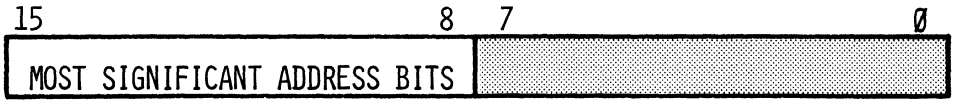


FIGURE 8 AmZ8002 NEW PROGRAM STATUS AREA POINTER

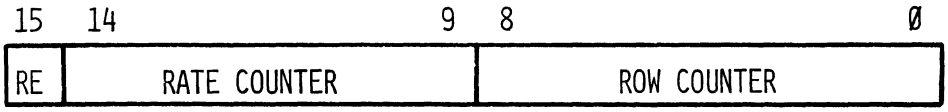


FIGURE 9 REFRESH COUNTER

The automatic refresh feature can be disabled by loading a zero into REFRESH ENABLE bit. When the CPU is reset for initialization, this bit is set to "1" i.e. refresh is enabled. Access to the refresh counter is made using the LDCTL instruction.

Addressing Modes

Operands needed to execute an instruction are designated by register addresses, memory addresses or I/O addresses. The addressing mode of a given instruction not only designates the relevant address space but also defines the method to be used in computing the operand address. Addressing modes are either explicitly specified or implied by the instruction. Eight explicit addressing modes are provided: Register (R), Indirect Register (IR), Direct Address (DA), Immediate (IM), Indexed (X), Base Address (BA), Base Indexed (BX) and Relative Address (RA). Autoincrement and Autodecrement are the two implied addressing modes in block and string manipulation instructions.

The following is a detailed explanation of explicit addressing modes.

Register (R) Mode: The operand used by the instruction is located in a general purpose register as shown in Figure 10. The instruction specifies the length of the operand (byte, word or long word) and a 4-bit field in the instruction designates the intended register.

Indirect Register (IR): The instruction designates a general purpose register; contents of the designated register are not the operand but address of the operand. The AmZ8001 deals with 23-bit segmented addresses and hence a register pair is designated by the instruction. The first register contains the 7-bit segment number and the second register contains the 16-bit offset as shown in Figure 11. Any general purpose register pair except RRØ can be designated for this addressing mode. The AmZ8002 requires only 16-bit addresses as shown in Figure 12 and hence any general purpose register except RØ can be designated for IR addressing mode.

Direct Address (DA): The instruction itself explicitly specifies an address and the operand used by the instruction is located at that address. In AmZ8001 direct addresses are specified in one of two formats - long offset and short offset. For the long offset, the memory word immediately

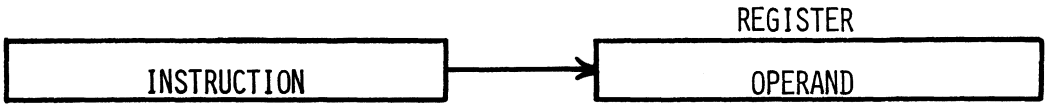


FIGURE 10 AMZ8001 AND AMZ8002 REGISTER ADDRESSING MODE

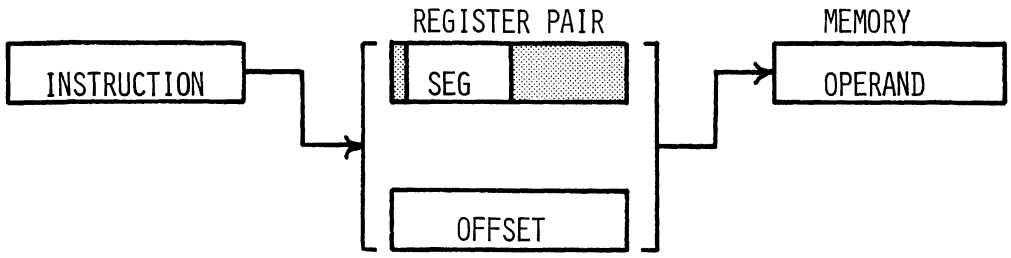


FIGURE 11 AMZ8001 INDIRECT REGISTER ADDRESSING MODE

FIGURE 12 AMZ8002 INDIRECT REGISTER ADDRESSING MODE

following the instruction opcode word contains the 7-bit segment number and the memory word immediately following the segment number word is the 16-bit offset as shown in Figure 13A. For the shoft offset, the memory word immediately following the instruction opcode word contains both 7-bit segment number and 8-bit offset as shown in Figure 13B. In AmZ8002, the memory word immediately following the instruction opcode word contains the 16-bit address as shown in Figure 14.

Immediate (IM): The instruction itself contains the operand as shown in Figure 15. In general, the memory word immediately following the instruction opcode word contains the immediate operand. In case of 32-bit immediate operand, two memory words immediately following the instruction opcode word are used.

Indexed (X): The instruction designates a 16-bit general purpose register as the index register. Any general purpose register except RØ can be used as the index register. The instruction also specifies an address as in the direct address mode. In the AmZ8001, the 16-bit contents of the designated index register are added to the 16-bit offset value specified in the instruction. Both index and offset are treated as 16-bit unsigned integers and any carry from the most significant bit position during this addition is ignored. The resulting 16-bit sum together with the 7-bit segment number specified in the instruction is used as 23-bit segmented address as depicted in Figure 16A. The operand will be located at this address in memory. If short offset is used in the AmZ8001 for indexed addressing mode, the memory word immediately following the instruction opcode word contains both a 7-bit segment number and an 8-bit offset as shown in Figure 16B.

A 16-bit unsigned integer is formed whose least significant byte is the 8-bit offset specified and most significant byte is zero. The 16-bit word thus formed is added to the 16-bit unsigned integer contained in the designated general purpose register. Any carry from the most significant bit position during this addition is ignored. The 16 bits resulting from this addition together with the 7-bit segment number specified is the 23-bit address. The operand will be located in the memory at this address.

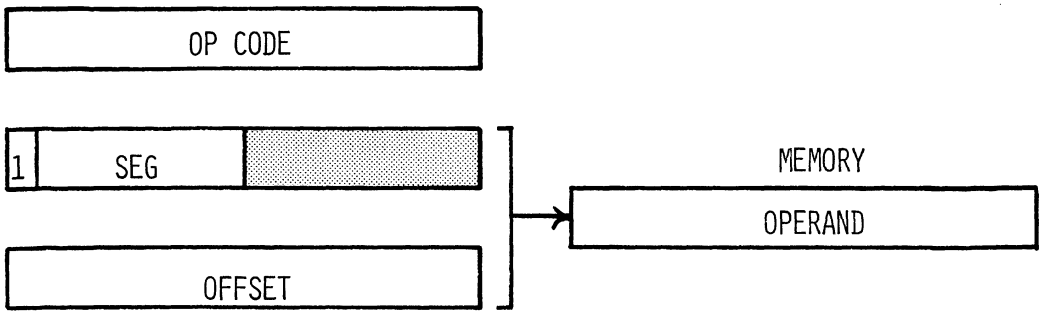


FIGURE 13A AmZ8001 DIRECT ADDRESSING MODE--LONG OFFSET

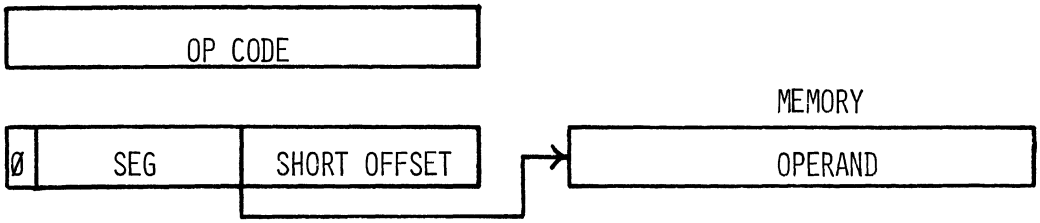


FIGURE 13B AmZ8001 DIRECT ADDRESSING MODE--SHORT OFFSET

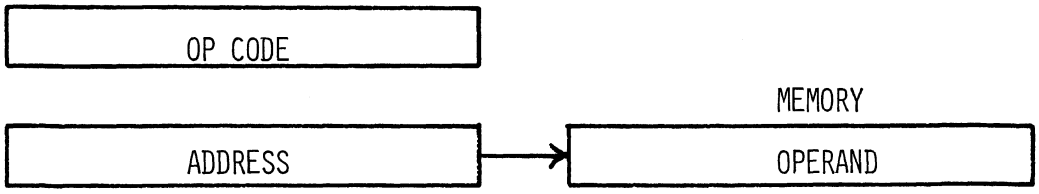
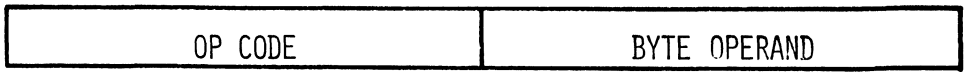
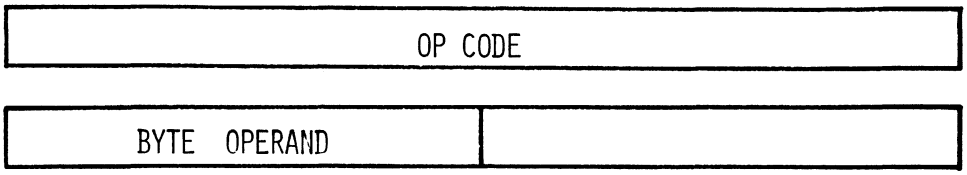


FIGURE 14 AmZ8002 DIRECT ADDRESSING

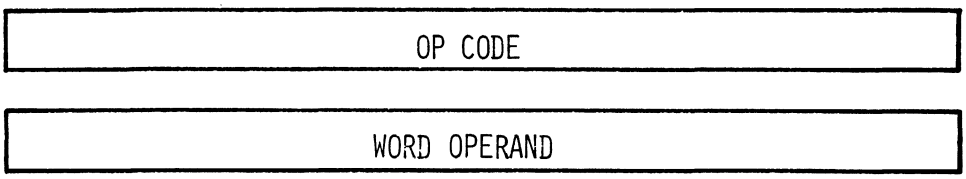
IMMEDIATE 4-BIT OPERAND (DIGIT)



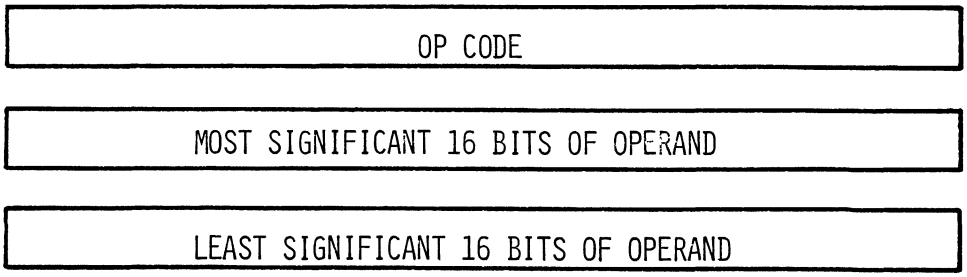
IMMEDIATE 8-BIT OPERAND (BYTE)



ALTERNATIVE IMMEDIATE 8-BIT OPERAND (BYTE)



IMMEDIATE 16-BIT OPERAND (WORD)



IMMEDIATE 32-BIT OPERAND (LONG WORD)

FIGURE 15 AmZ8001 AND AmZ8002 IMMEDIATE ADDRESSING MODE

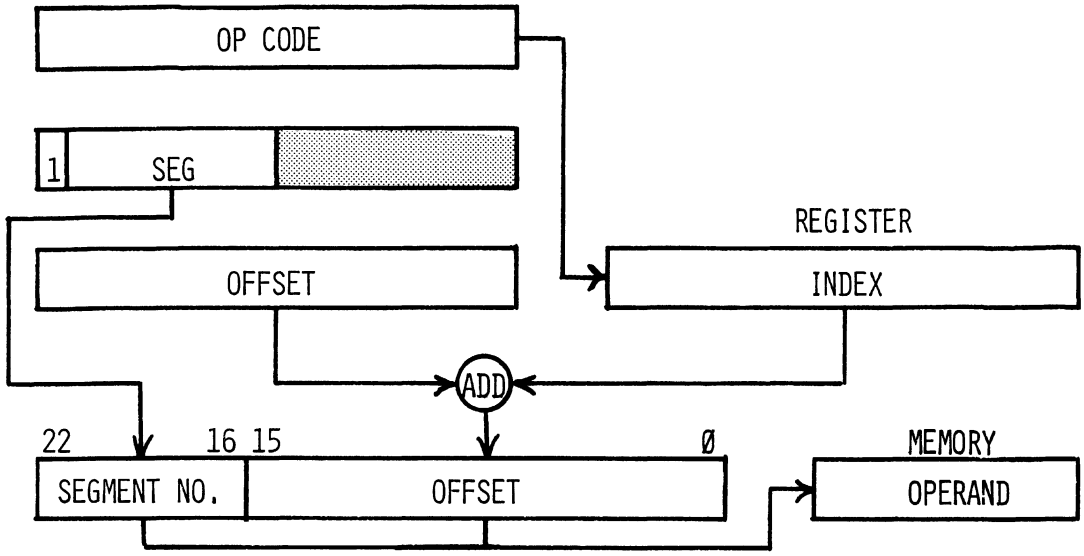


FIGURE 16A AmZ8001 INDEXED ADDRESSING MODE (LONG OFFSET)

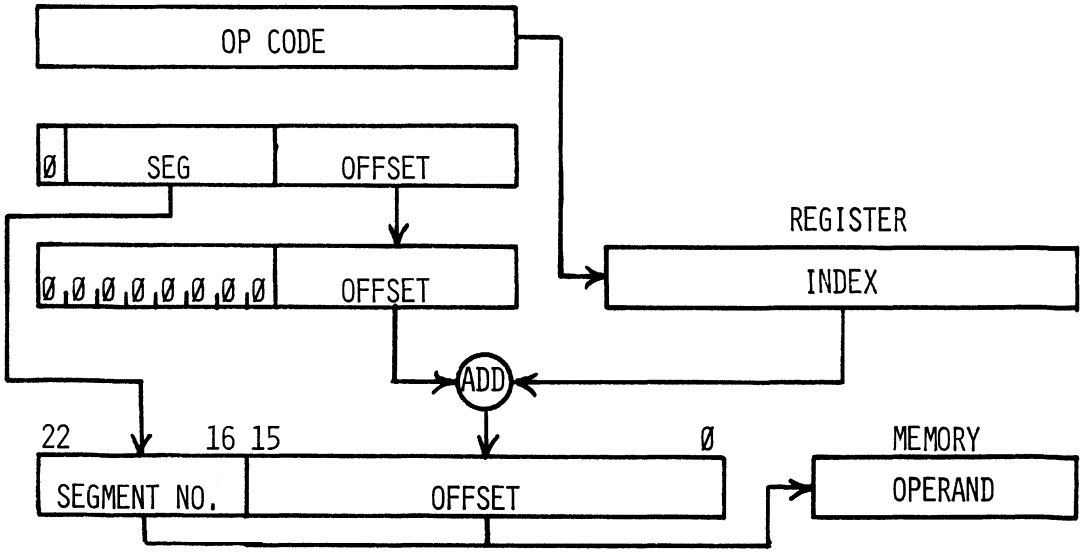


FIGURE 16B AmZ8001 INDEXED ADDRESSING MODE (SHORT OFFSET)

In AmZ8002, the memory word immediately following the instruction opcode word contains a 16-bit address as shown in Figure 17. This unsigned integer is added to the 16-bit unsigned integer located in the designated index register. The carry from the most significant bit position during this addition is ignored. The resulting 16-bit address is where the operand is located in the memory.

Base Address (BA): The instruction designates a general purpose register as the base address register. In case of AmZ8001, the instruction designates a register pair such that the 7-bit segment number is contained in one register and 16-bit offset is contained in the other as shown in Figure 18. In case of AmZ8002, the designated base address register contains 16-bit address as shown in Figure 19. Any general purpose register except RØ or register pair except RRØ can be designated as the base address register. The memory word immediately following the instruction opcode word contains a 16-bit displacement. Both displacement and base address are treated as unsigned binary integers. The 16-bit displacement is added to the 16-bit base address (16-bit offset is AmZ8001) and carry occurring from the most significant position during this addition is ignored. The resulting 16-bit value (together with the segment number of the base address in AmZ8001) is the address of the operand in memory.

Base Indexed (BX): The instruction designates a general purpose register (register pair in AmZ8001) as the base address register. The instruction also designates a 16-bit general purpose register as displacement. Any general purpose register except RØ (AmZ8002) or any register pair except RRØ (AmZ8001) can be used as the base address register. Similarly any general purpose register except RØ can be used as the displacement register. Both base address and displacement are unsigned integers.

The 16-bit displacement is added to the base address (or offset of the base address in AmZ8001) and carry from the most significant bit position during this addition is ignored. The 16-bit result (together with base address segment number) is the address of the operand in memory. Figure 20 and Figure 21 illustrate this addressing mode for AmZ8001 and AmZ8002.

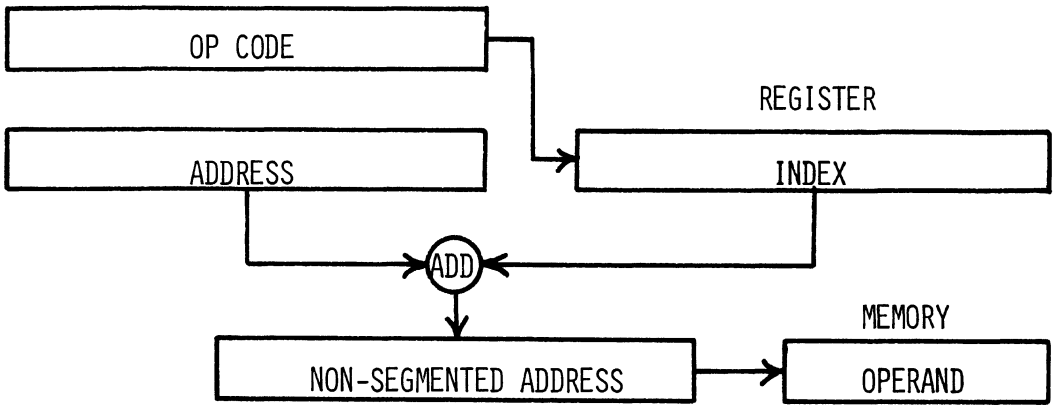


FIGURE 17 AMZ8002 INDEXED ADDRESSING MODE

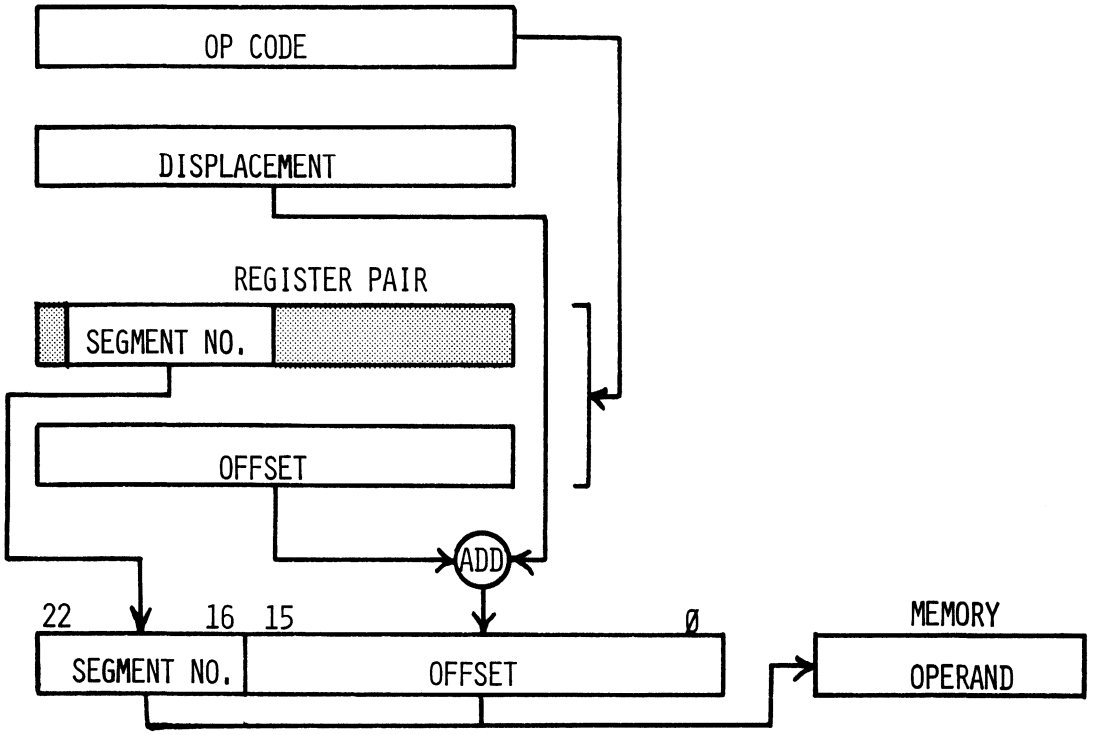


FIGURE 18 AMZ8001 BASE ADDRESS MODE

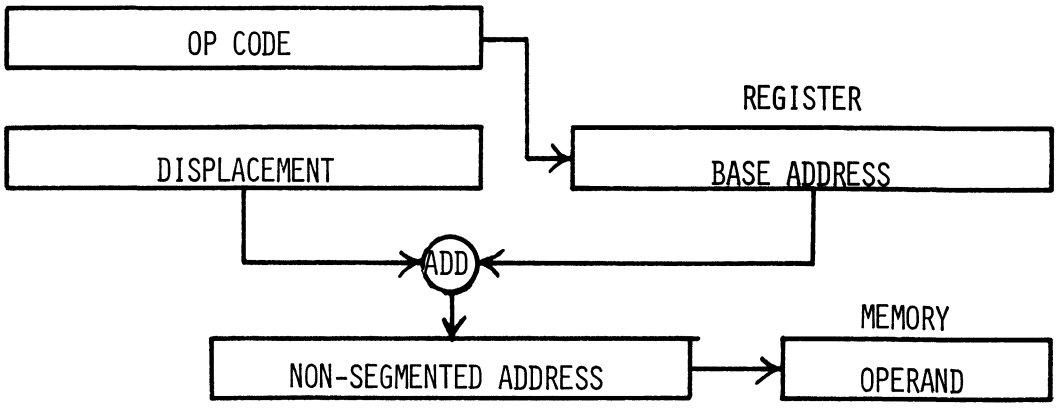


FIGURE 19 AMZ8002 BASE ADDRESS MODE

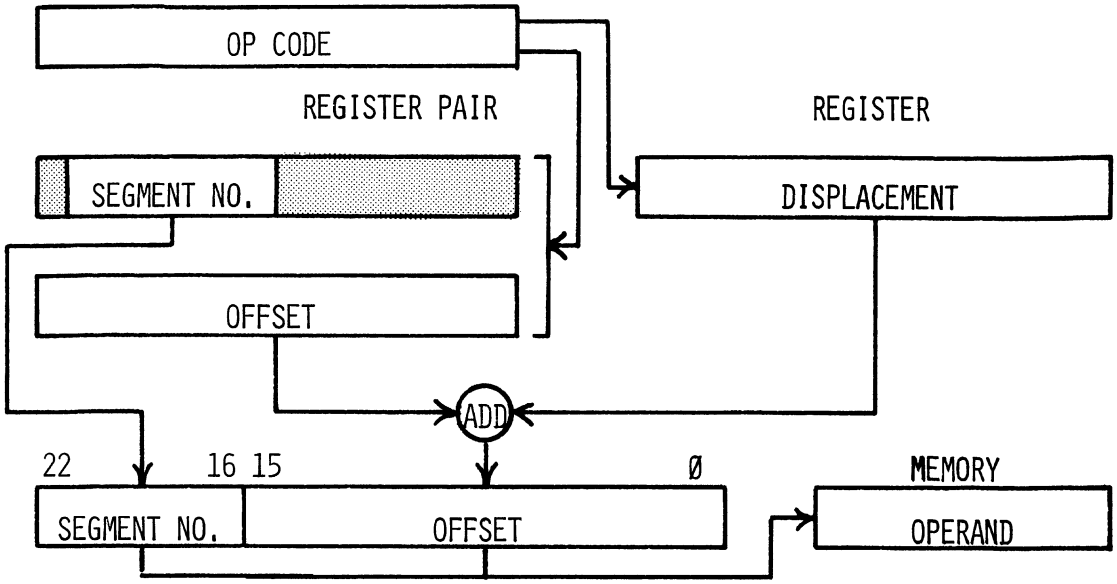


FIGURE 20 AMZ8001 BASE INDEX ADDRESSING MODE

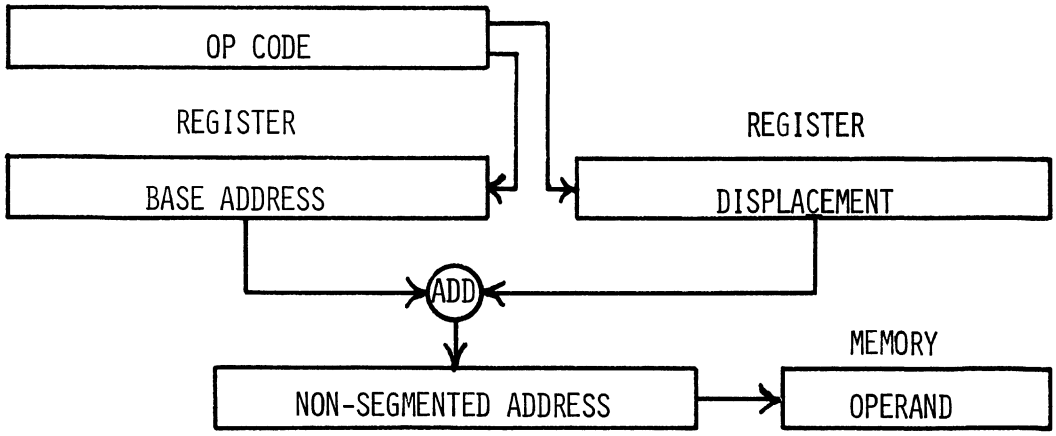


FIGURE 21 AMZ8002 BASE INDEX ADDRESSING MODE

Relative Address (RA): The instruction itself contains a displacement. This displacement is a signed integer using two's complement notation. The number of bits allocated to represent the displacement depend on the instruction where relative addressing mode is available. The displacement is sign extended appropriately to obtain a signed 16-bit displacement. The sign extended displacement is added to the 16-bit program counter (PC OFFSET in AmZ8001). Carry from the most significant bit position during this addition is ignored. As soon as the instruction using the relative address mode is fetched, the PC will be updated. Hence, updated PC value (i.e. address of the following instruction) will be used for address calculation.

The 16-bit value obtained by adding the PC and displacement (together with the segment number in AmZ8001) is the address of the operand in memory. Figure 22 and Figure 23 illustrate the relative addressing mode.

Autoincrement and Autodecrement: These two implied addressing modes are only used in string manipulating instructions. These addressing modes are a variation of the IR addressing mode. The instruction designates a general purpose register (or a register pair in AmZ8001) whose contents are used as the address. After fetching the operand, the contents of the register are incremented or decremented depending on Autoincrement or Autodecrement. In case of AmZ8001, only the register containing the offset is affected and any carry resulting from this operation is ignored. For byte operations incrementing or decrementing by 1 occurs. For word operations incrementing or decrementing by 2 takes place.

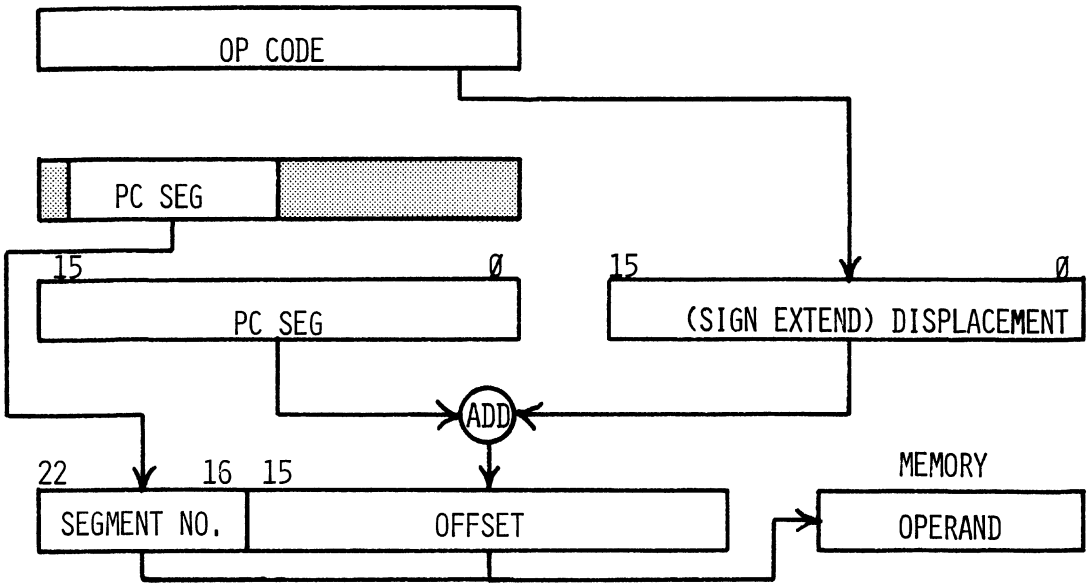


FIGURE 22A AMZ8001 RELATIVE ADDRESSING MODE --(ONE WORD INSTRUCTION)

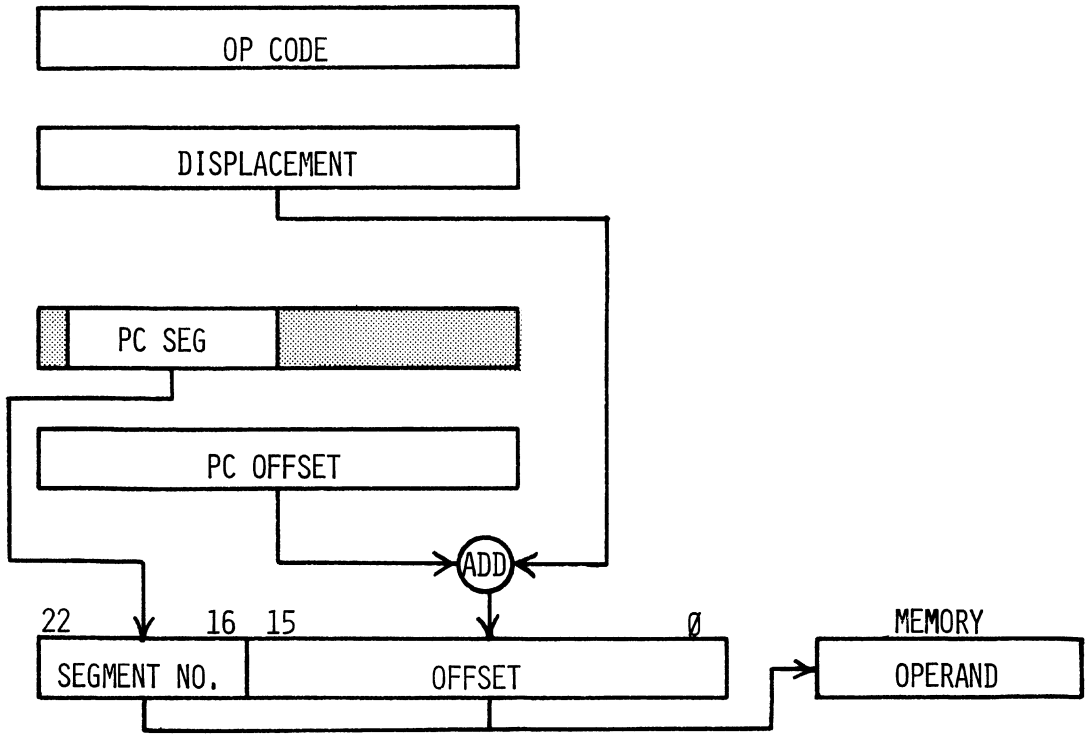


FIGURE 22B AMZ8001 RELATIVE ADDRESSING MODE--(TWO WORD INSTRUCTION)

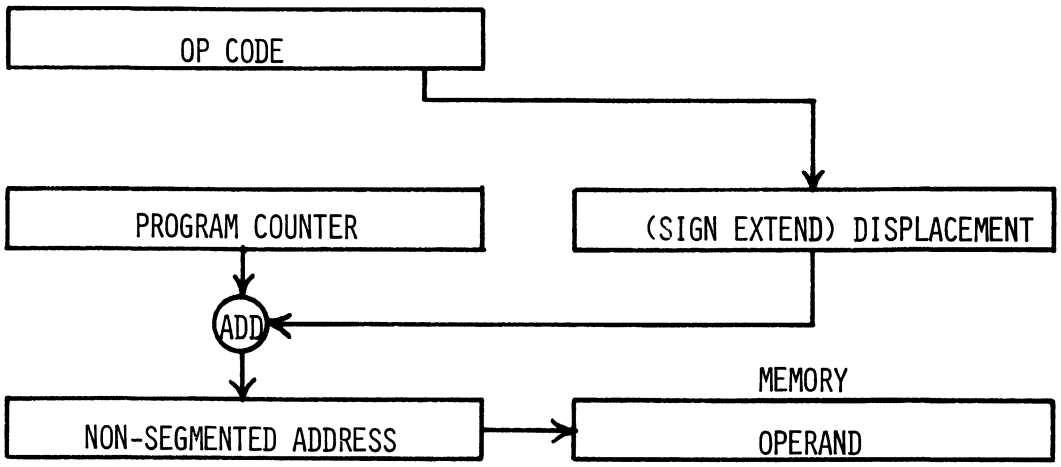


FIGURE 23A AMZ8002 RELATIVE ADDRESSING MODE--(ONE WORD INSTRUCTION)

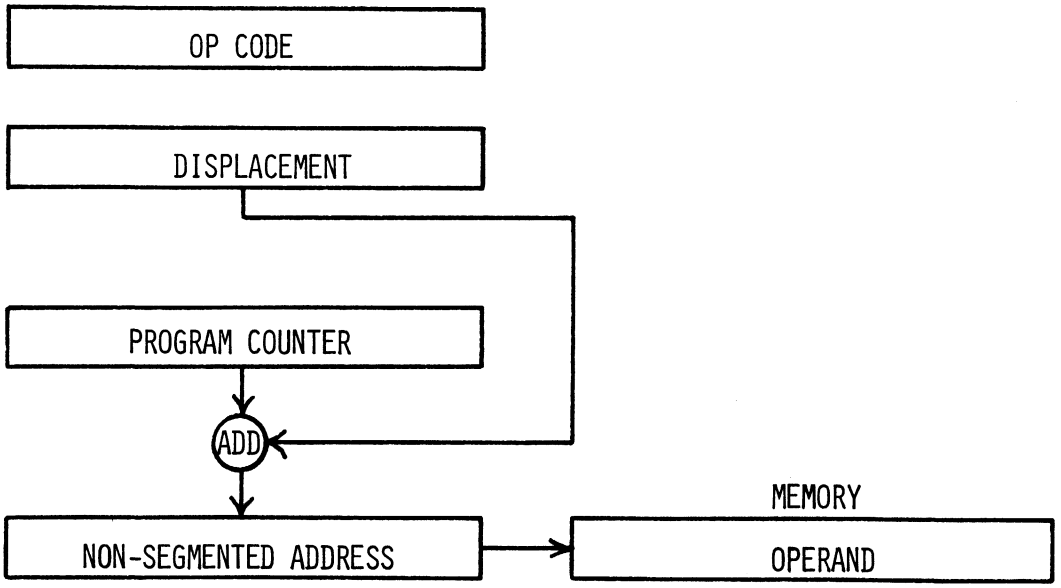


FIGURE 23B AMZ8002 RELATIVE ADDRESSING MODE--(TWO WORD INSTRUCTION)

Operand Addressing

Seven types of operands are handled by the instructions - bits, digits (4-bits), bytes (8-bits), words (16-bits), long words (32-bits), byte strings and word strings. In general, operands may be contained in a general purpose register or located in memory. However, string operands must be located in memory only. The elements of a string reside in consecutive memory locations.

Figure 24 illustrates the conventions used in relating one operand type to another. A byte consists of two digits:; two bytes make a word and two words make a long word. The left most element whether bit, digit or byte is always the most significant. Bits of an operand are numbered from right to left starting with zero for the least significant bit. The bit operand located in a general purpose register is addressed by the designation of the byte register containing the desired bit and specifying the bit number in that byte. Bit operands can also be addressed by designating a 16-bit general purpose register holding the word that contains the desired bit and the bit number in that word. Digits are always addressed by designating the byte register containing the desired digit. Word and long word operands located in registers are addressed by the register or register pair designation. When a long word is specified using a register pair, the even numbered register of the pair contains the most significant 16-bits and the odd numbered register of the pair contains the least significant 16 bits.

Memory Addressing

Memory address space is viewed as a chain of consecutively numbered (in ascending order) bytes as shown in Figure 25. Also note that the numbering starts with zero. The number of each byte is its address. Thus, the byte is the basic addressable element in memory. A word in memory spans two byte addresses. The most significant byte of a word must always be an even address and the least significant byte is the immediately following odd address. This arrangement is called "word aligned". Thus, words are addressed by the address of the most significant byte

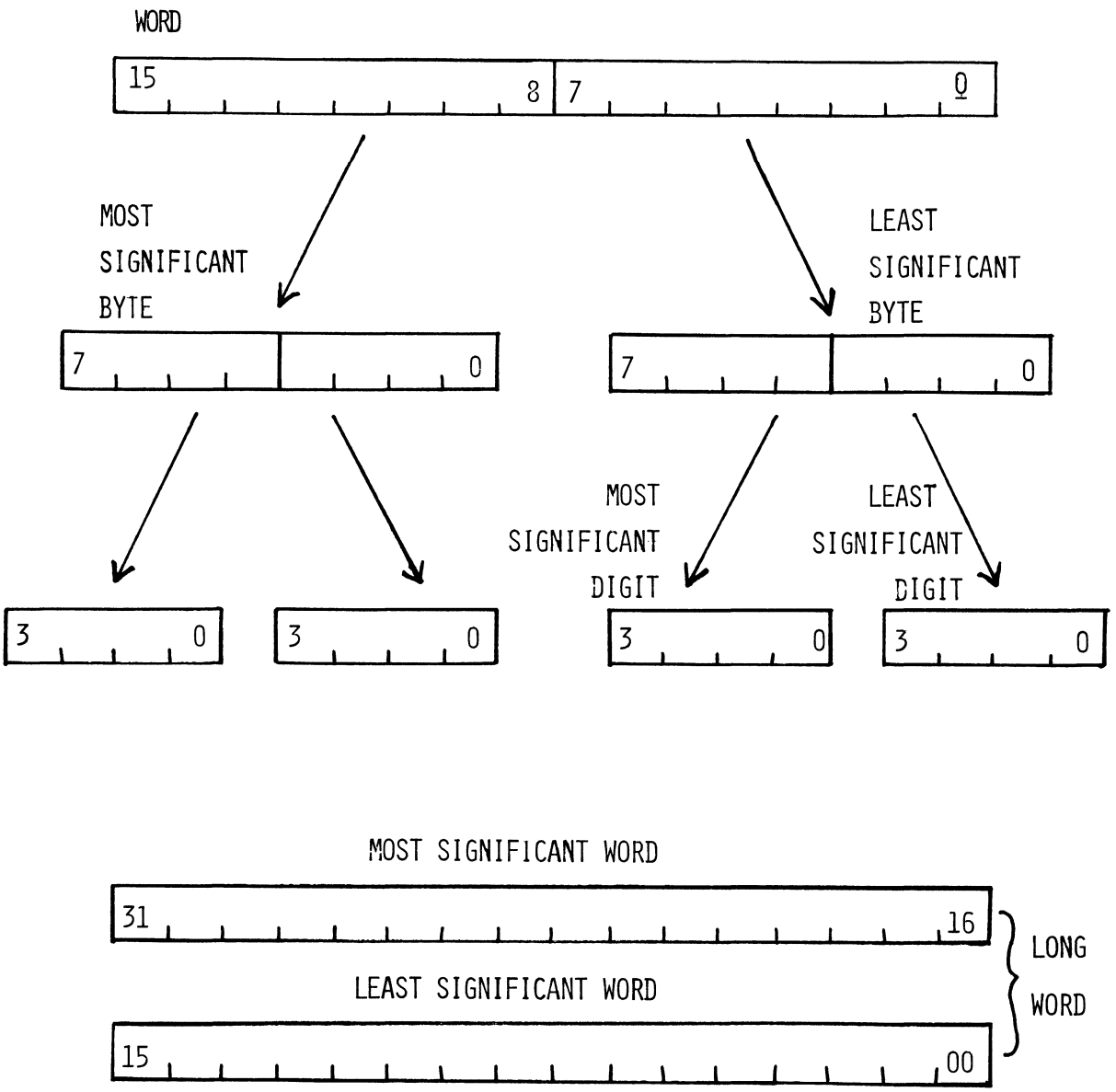


Figure 24. Operand Notation

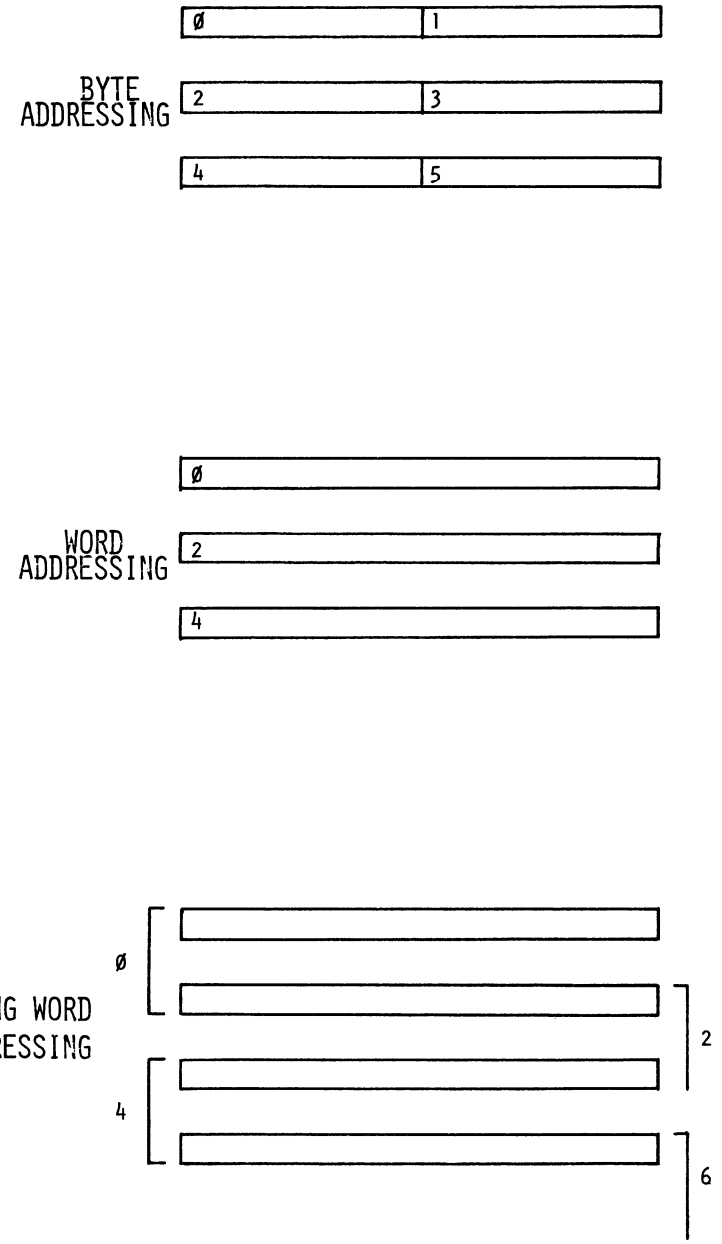


FIGURE 25. MEMORY ADDRESSING

or even address. A long word in memory spans two words or four bytes. The most significant 16-bits are contained at a word address and the least significant 16-bits are contained at the immediately following word. For example a long word is contained in memory addresses Ø and 2, then location Ø contains the most significant 16-bits and location 2 contains the least significant 16 bits. Long word operands in memory are addressed by specifying the address of the most significant byte of the most significant word. For example, address Ø is used for a long word operand located at locations Ø and 2. Instructions are always addressed as words and hence instructions in memory must be word aligned.

The CPU can handle both byte and word string operands. Two parameters are needed to specify a string - starting address (address of the first element) and the length of the string expressed in terms of the number of elements in the string as depicted in Figure 26. For example, a word string starting at address 100 and 25 words long will be characterized by the starting address 100 and length 25. Similarly a byte string which is 15 bytes long and starts at address 157 will be characterized by the starting address 157 and its length 15. By specifying Autoincrement addressing mode in a string manipulating instruction, successive elements of strings specified in the above manner can be accessed for processing. String operands can also be specified by the ending address (address of the last element) instead of the starting address and the length expressed in terms of number of elements in the string. Autodecrement addressing mode is used to access successive elements of the string in this case. It should be noted that when dealing with byte strings, there are no restrictions on whether the starting and ending addresses are odd or even. However, because of the word alignment requirement, word strings can have only even addresses.

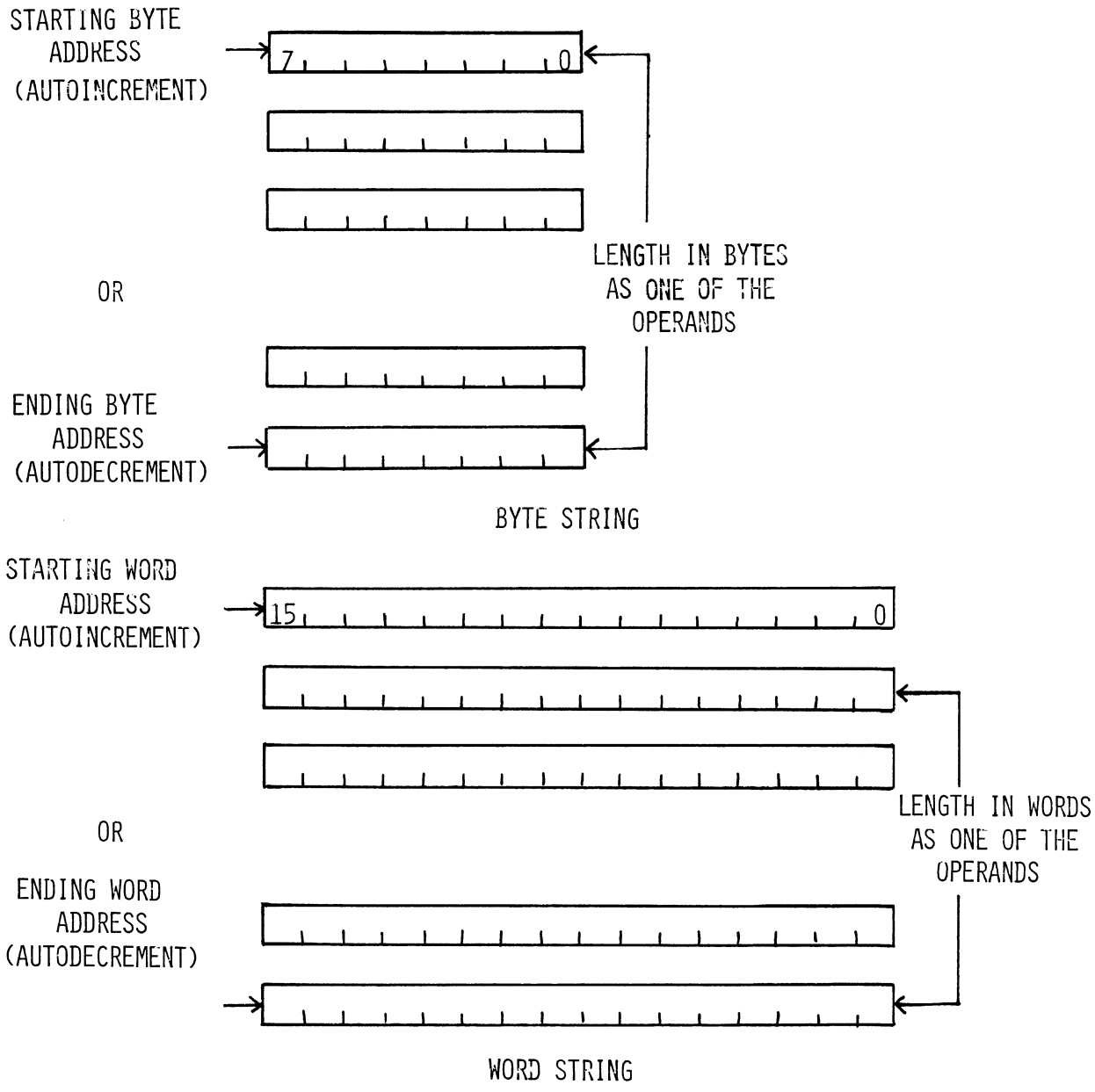


Figure 26. String Operands

Interrupts and Traps

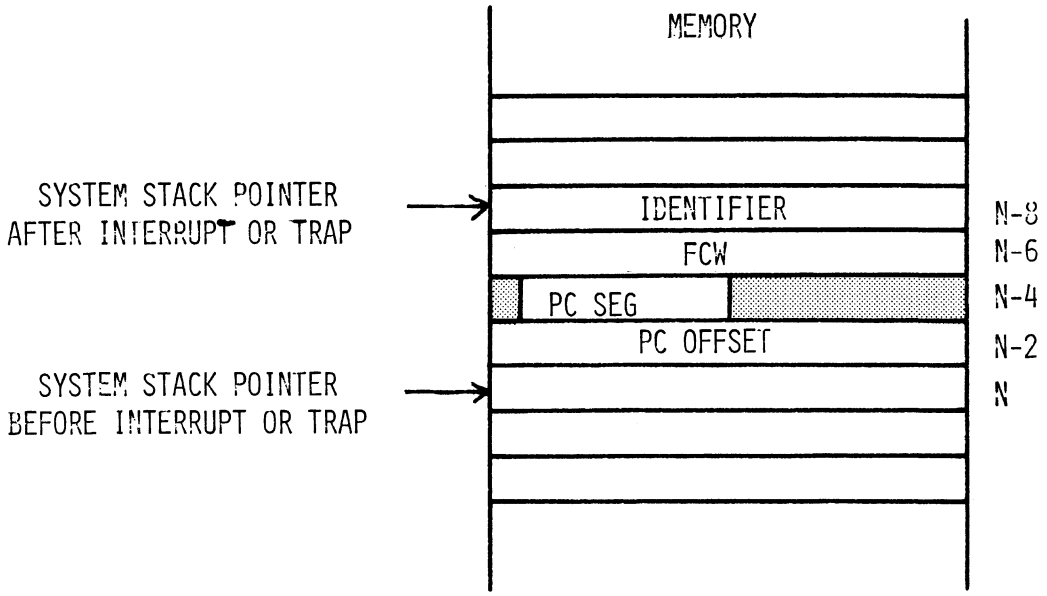
Program interruptions are divided into two groups - interrupts and traps. In general, interrupt is an external asynchronous event needing the CPU's attention. Trap usually is a synchronous event resulting from the execution of certain instructions under some specified condition. Also an interrupt may be disabled in the CPU by an appropriate control bit in the FCW; traps cannot be disabled. Procedures followed by the CPU are essentially the same for interrupts and traps.

When an interruption occurs, the current program status information is automatically pushed on the system stack as shown in Figure 27. As discussed before, program status consists of Processor Status (PC and FCW) and a 16-bit word called Identifier. The Identifier contains information relating to the reason for this interruption.

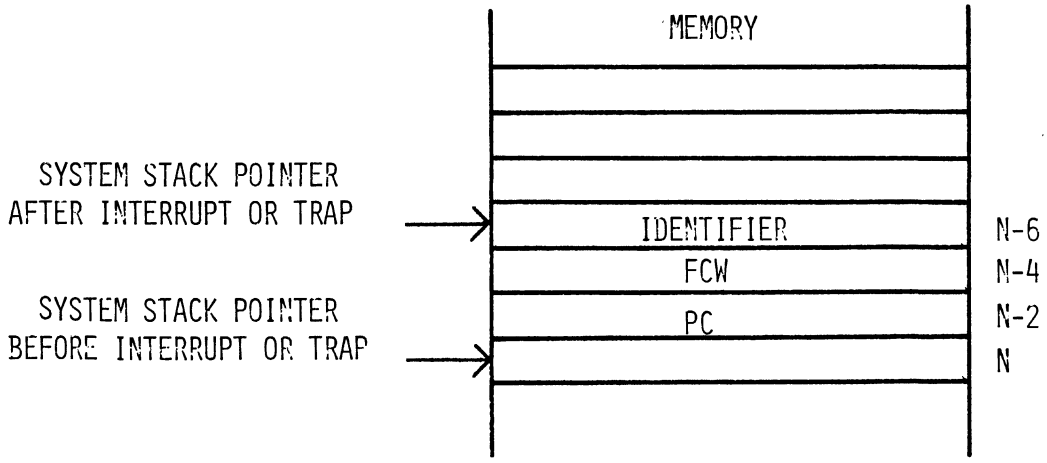
There are three interrupts listed in order of decreasing priority: non-maskable, vectored and non-vectored. There are four traps: system call, unimplemented opcode, privileged instruction in normal mode and segmentation error. For all three interrupts the Identifier is a 16-bit entity supplied by the interrupting device. The Identifier in case of traps (except segmentation error) is the first word of the instruction that caused the trap. This word always contains the instruction opcode.

The segmentation error actually results from several exception conditions that could occur when the Memory Management Unit (AmZ8010) is used in an AmZ8001 system. Detailed discussion of the Memory Management unit is beyond the scope of this document. It is sufficient for the current discussion to know that the Identifier for the segmentation error trap will be supplied by the Memory Management circuitry.

After saving the program status in the system stack, new processor status is automatically obtained from a predetermined area in memory called New Program Status Area. The New Program Status Area Pointer (NPSAP) specifies the area in memory where the New Program Status Area is located. In AmZ8001,



AmZ8001 PROGRAM STATUS-SAVING SEQUENCE

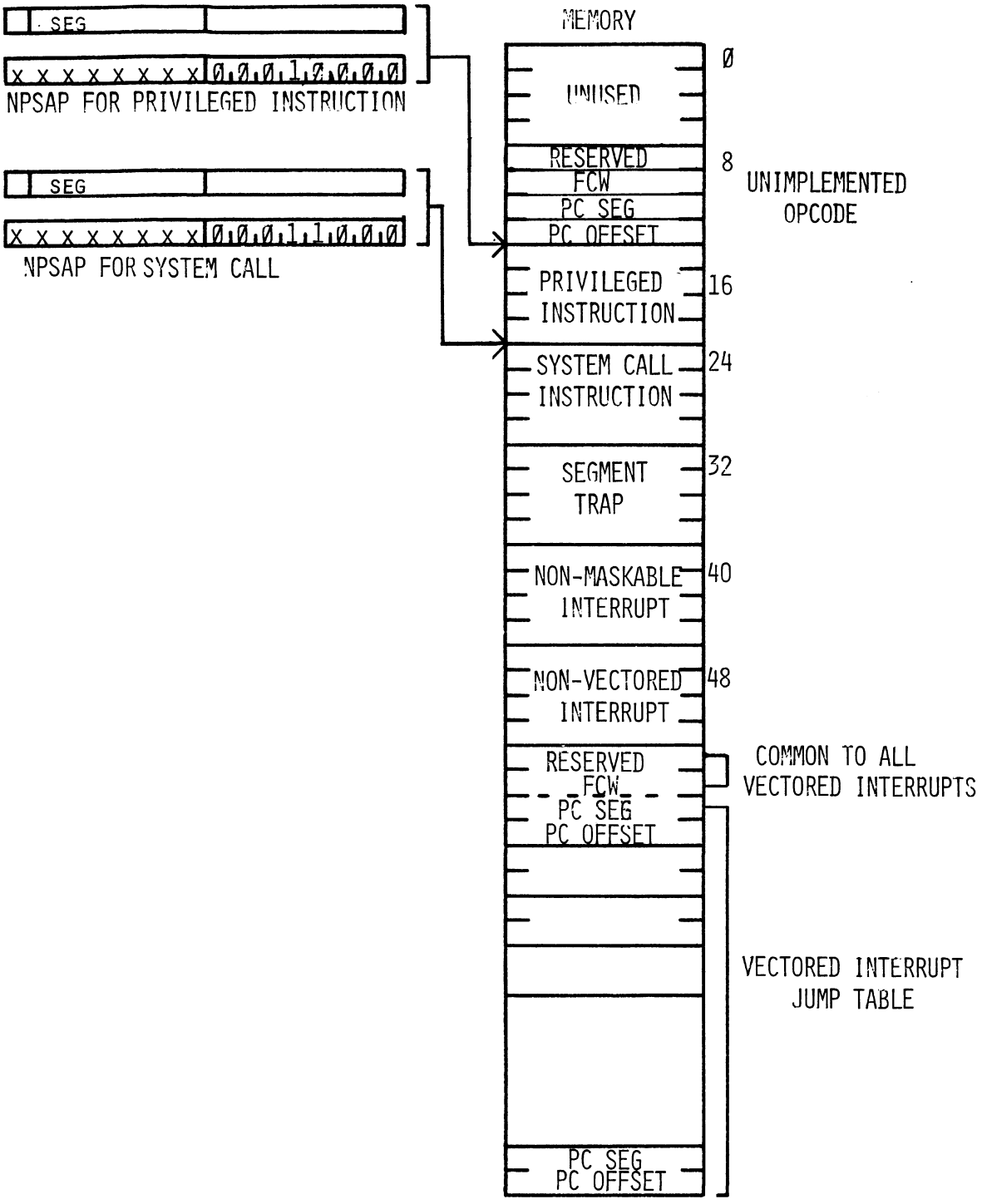


AmZ8002 PROGRAM STATUS-SAVING SEQUENCE

FIGURE 27. PRE-INTERRUPT PROGRAM STATUS IN THE STACK

NPSAP consists of a 7-bit segment number and most significant eight bits of the offset. In AmZ8002 this pointer contains the most significant eight bits of the address. The CPU utilizes a predetermined value in the least significant eight bits of the NPSAP offset. Figure 28 and Figure 29 show the New Program Status Area format. For example, in AmZ8001, the first four locations are used for segmentation error, next four locations for system call trap and so on.

The format of storage for all interruptions is the same. In AmZ8001, New Program Status is contained in four consecutive memory locations. These are in ascending order, Reserved Word, New FCW, new PC SEG and PC OFFSET. The first location for every new processor status area of AmZ8001 is reserved for future CPU expansion and should not be used in the interest of upward software compatibility. In the AmZ8002, only two memory locations are needed for the new processor status information. Two consecutive memory locations in ascending address are used for new FCW and new PC.



FIGRE 28. AMZ8001 NEW PROGRAM STATUS AREA

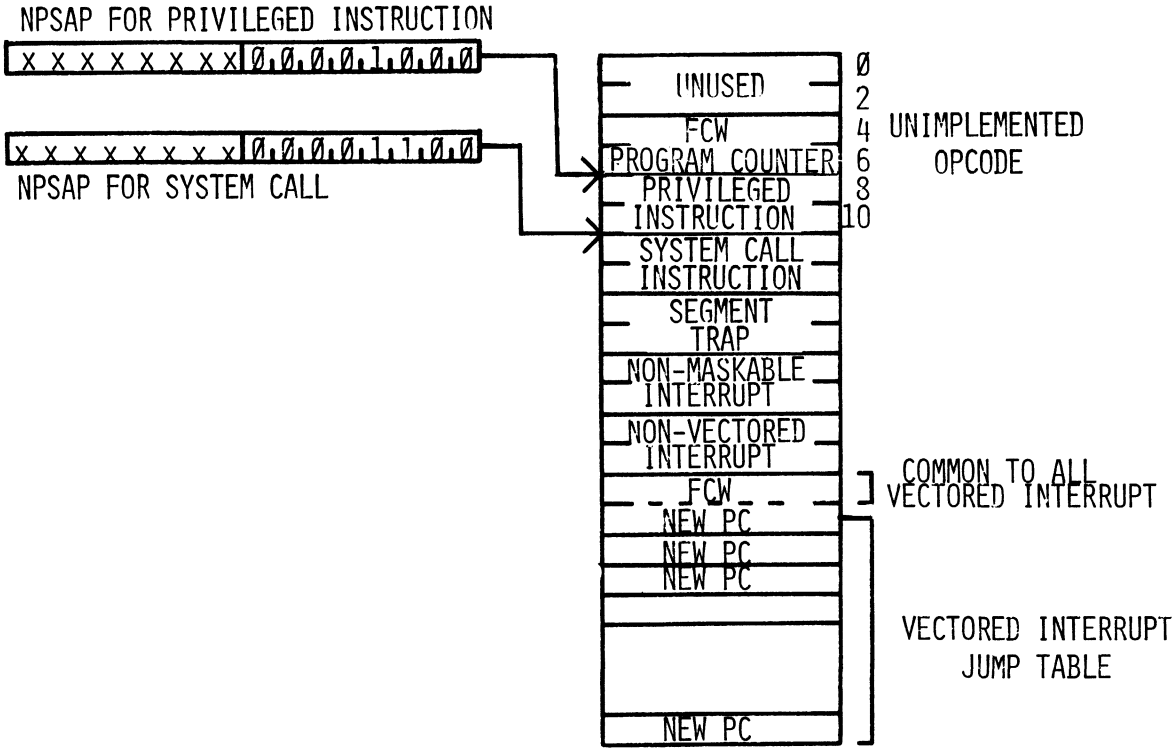


FIGURE 29. AMZ8002 NEW PROGRAM STATUS AREA

Instruction Format

The CPU instructions are one to five words long depending on the type of instruction and addressing mode. Instructions are located in memory and must be word aligned. The first word of an instruction always contains the opcode. Depending on the addressing mode, one or more words will follow the opcode word of an instruction. Figure 30 illustrates the general opcode word format. Some instructions contain fields that differ from the generalized format shown. All such variations can be ascertained by referring to the individual instruction descriptions found in later sections of this document. In Figure 30, the Mode Field (bit 14 and bit 15), together with bit 12 and bit 13 and bits 4, 5, y and 7 determine the applicable addressing mode. Bit 8 of the opcode word specified word or byte operand whenever applicable. Table 3 is a summary of addressing mode decoding. Bits 4, 5, 6 and 7 normally designate a general purpose register. Note that when designating a register pair, bit 4 must be zero and only 5, 6 and 7 are used.

From Table 3 it can be seen that for Register Mode of addressing there are no restrictions on the values of bits 4, 5, 6 and 7. Only the Mode field is needed to specify this addressing mode. This allows designating any general purpose register. However, for IM, RA and DA addressing modes, bits 4, 5, 6 and 7 must all be zero. For these addressing modes zeros in bits 4, 5, 6 and 7 are not interpreted as general purpose register number zero. Similarly, for IR, BA, X and BX addressing modes, bits 4,5, 6 and 7 cannot be zero. In other words, general purpose register number zero cannot be used in these addressing modes. It should be emphasized that if a register pair is needed for these addressing modes, bit 4 is always zero and non-zero comment applies to bits 5, 6 and 7.

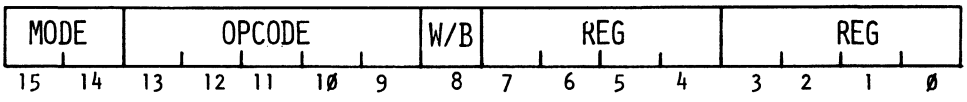


FIGURE 30. GENERAL INSTRUCTION FORMAT

MODE BITS 15, 14	OP CODE BITS 13, 12	REG BITS 7,6,5,4	ADDRESSING MODE
1 0	XX	XXXX	R
0 0	Any Value but 1 1	0	IM
0 0	Any Value but 1 1	Non-Zero	IR
0 0	1 1	0	RA
0 0	1 1	Non-Zero	BA
0 1	XX	0	DA
0 1	Any Value but 1 1	Non-Zero	X
0 1	1 1	Non-Zero	BX

Table 3. Addressing Mode Encoding

Input/Output

A set of input/output (I/O) instructions is provided to perform 16-bit or 8-bit transfers between the CPU and I/O devices. Input/Output devices are addressed using a 16-bit address called port address. Conceptually the port address is very similar to a memory address. Logically, however, port address space is not a part of the memory address space. Although memory and port address information is physically transmitted on the same bus lines in hardware, means are provided to distinguish memory addresses from I/O addresses using status output lines supplied by the CPU. Port address generation uses the same methodology that is used to generate operand addresses in the non-segmented CPU using IR and DA addressing modes.

Two types of I/O instructions are available - standard I/O and special I/O. The address space used by the special I/O is logically separate from the standard I/O. Special I/O address space can be distinguished from the standard I/O space using the status output lines from the CPU. A byte transferred using the standard I/O instruction appears on the least significant 8 bus lines in hardware. However, when transferring a byte using special I/O instruction, the byte will be on the most significant 8 bus lines. This is the only major difference between standard I/O and special I/O operations. Major discussion on the special I/O instructions is beyond the scope of this document. It should be enough to mention that special I/O instructions are intended for communicating with the Memory Management unit. The I/O instructions exist not only to transfer single words or bytes of data, but also blocks of data from contiguous memory locations.

Condition Codes

The Condition Code (CC) is a 4-bit field in some instructions that specifies certain flag settings. The operation performed by the instruction is in most cases determined by the outcome of comparing the actual flag settings with that specified by the CC field. Instructions that specify CC field include conditional jumps, return from subroutine and block/string manipulating instructions. The Condition Code definitions consist of true and

false settings of the C, Z and P/V flags, signed and unsigned comparisons as shown in Table 4. One of the CC values specifies unconditional combination in which flag settings are ignored.

CC FIELD	MEANING			FLAGS
1110	NZ	-	Not zero	Z = Ø
0110	Z	-	Zero	Z = 1
1111	NC	-	No carry	C = Ø
0111	C	-	Carry	C = 1
1100	PO	-	Parity odd	P/V = Ø
0100	PE	-	Parity even	P/V = 1
1101	PL	-	Plus	S = Ø
0101	MI	-	Minus	S = 1
1110	NE	-	Not equal	Z = Ø
0110	EQ	-	Equal	A = 1
1100	NOV	-	Overflow is reset	P/V = Ø
0100	OV	-	Overflow is set	P/V = 1
	SIGNED COMPARISONS:			
1001	GE	-	Greater than or equal	S XOR P/V = Ø
0001	LT	-	Less than	S XOR P/V = 1
1010	GT	-	Greater than	Z OR (S XOR P/V) = Ø
0010	LE	-	Less than or equal	Z OR (S XOR P/V) = 1
	UNSIGNED COMPARISONS:			
1111	LGE	-	Logical greater than or equal	C = Ø
0111	LLT	-	Logical less than	C = 1
1011	LGT	-	Logical greater than	C = Ø AND Z = Ø
0011	LLE	-	Logical less than or equal	C OR Z = 1
1000	UNCONDITIONAL			

TABLE 4. CC - FIELD DECODING

The following pages contain detailed description of the individual instructions. Figure 31 illustrates a sample of the information presented with each instruction.

Top left hand corner shows the title of the instruction and then the mnemonic at the top center in each page. If an instruction is priviliged, this fact will be noted to the right of the mnemonic. The operation performed by the instruction is represented by symbolic notation or a simple diagram whenever possible. In the symbolic notation, the operand lengths are designated by two integers separated by a colon between two angled brackets. For example dst<0:15> means that the destination operand occupies 16-bits. If there is only one integer contained between the brackets then the integer represents the bit number in an operand. For example, src<8> means bit number 8 of the source operand.

A detailed description of the instruction follows the operation. Also shown with each instruction are the applicable addressing modes for that instruction. The instruction format is shown with appropriate fields labelled. The instruction format shows the pre-assigned bit patterns for the fields whenever appropriate. The number of memory locations occupied by the instruction can also be found in the instruction format. For example, in Figure 31, SETB instruction using R addressing mode occupies one memory word.

To the left of the instruction format are the CPU characteristics and offset representation using the following abbreviations: S = Segmented, NS = Non-segmented, SSO = Segmented Short Offset, SLO = Segmented Long Offset. The numbers to the right of the instruction format represent the execution time for the instruction in number of clock cycles. Above each instruction format is the general notation representing the operands needed for the instruction. At the bottom of each page is a description and summary of the flags affected. Any shaded areas in the instruction formats are reserved for future CPU expansion and should not be used.

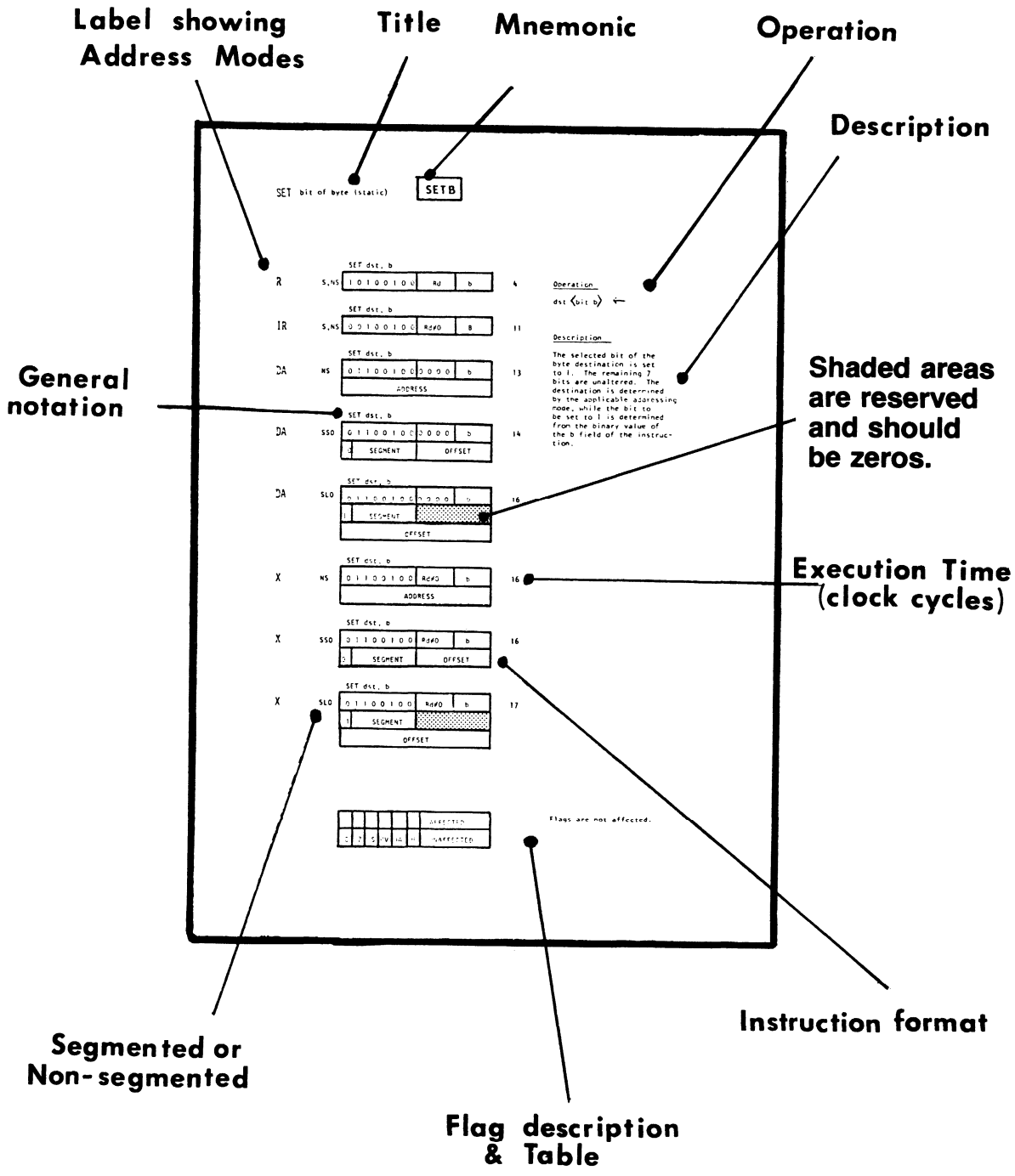


FIGURE 31. SAMPLE INSTRUCTION PAGE

ADC Rd, Rs

R S,NS

5

Operation

dst<0:15> ←src<0:15> +
dst<0:15> + C

Description

The contents of the general purpose registers designated by the Rs (source) and Rd (destination) fields of the instruction are added together along with the carry flag to obtain the result. The 16-bit result is loaded into the destination register, whose original contents are lost. The contents of the source are not altered.

Flags:

C: Set to 1 if there is carry from the most significant bit position of the word. Reset otherwise.
Z: Set to 1 if result is zero. Reset otherwise.
S: Set to 1 if result is negative. Reset otherwise.
P/V: Set to 1 on arithmetic overflow. Reset otherwise.

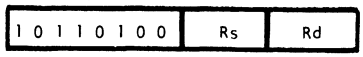
C	Z	S	P/V			AFFECTED
				DA	H	UNAFFECTED

ADD byte with carry

ADCB

ADCB Rd, Rs

R S,NS



5

Operation

Dst<0:7> ← Src<0:7>+ Dst<0:7>+ C

Description

The contents of the general purpose byte registers designated by the Rs (source) and Rd (destination) fields of the instruction are added together along with the carry flag to obtain the result. The 8-bit result is loaded into the destination register, whose original contents are lost. The contents of the source are not altered.

C	Z	S	PV	DA	H	
						AFFECTED
						UNAFFECTED

Flags:

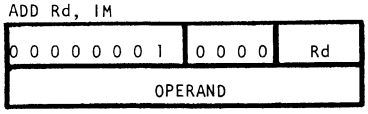
C: Set to 1 if there is a carry from most significant bit position of the byte. Reset otherwise.
Z: Set to 1 if result is zero. Reset otherwise.
S: Set to 1 if result is negative. Reset otherwise.
P/V: Set to 1 on arithmetic overflow. Reset otherwise.
DA: Reset always.
H: Set to 1 on carry from the least significant digit of result. Reset otherwise.

ADD word to register

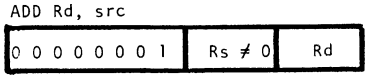
ADD

R S,NS — ADD Rd, Rs — 4

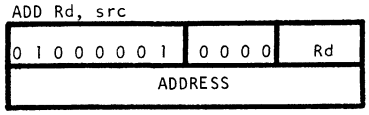
IM S,NS — ADD Rd, IM — 7



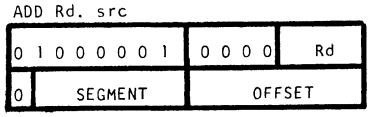
IR S,NS — ADD Rd, src — 7



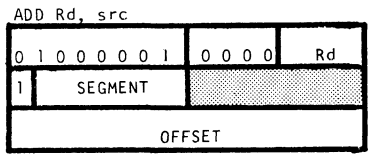
DA NS — ADD Rd, src — 9



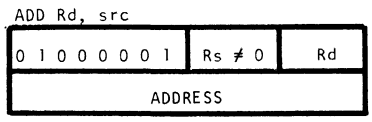
DA SSO — ADD Rd, src — 10



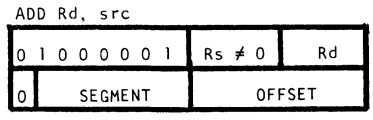
DA SLO — ADD Rd, src — 12



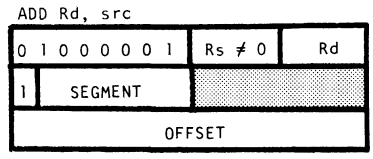
X NS — ADD Rd, src — 10

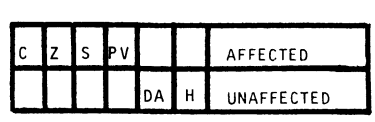


X SSO — ADD Rd, src — 10



X SLO — ADD Rd, src — 13





Operation

dst<0:15> ←src<0:15> + dst<0:15>

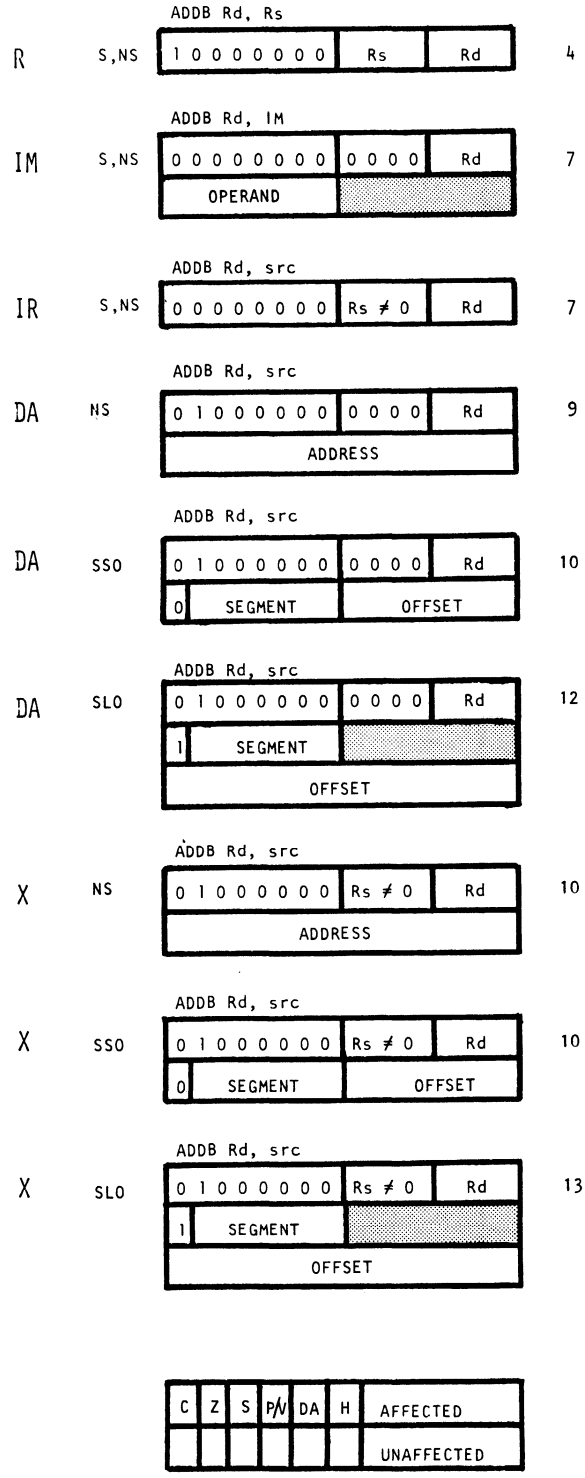
Description

Source operand and destination operand words are added together and the 16-bit result is loaded into the destination. The contents of the source are not altered and the original contents of the destination are lost. The source is determined by the applicable addressing mode and the destination is always a general purpose register designated by the Rd field of the instruction.

Flags:

C: Set to 1 if there is a carry from the most significant bit position of the word.
Z: Set to 1 if result is zero. Reset otherwise.
S: Set to 1 if result is negative. Reset otherwise.
P/V: Set to 1 on arithmetic overflow.

ADD byte to register **ADDB**



Operation

dst<0:7> ← src<0:7> + dst<0:7>

Description

Source operand and destination operand bytes are added together and the 8-bit result is loaded into the destination. The contents of the source are not altered and the original contents of the destination are lost. The source is determined by the applicable addressing mode and the destination is always a general purpose byte register designated by the Rd field of the instruction.

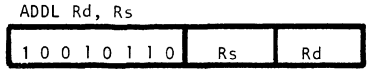
Flags:

C: Set to 1 if there is a carry from the most significant bit position of the byte.
Z: Set to 1 if result is zero. Reset otherwise.
S: Set to 1 if result is negative. Reset otherwise.
P/V: Set to 1 on arithmetic overflow.
DA: Always reset.
H: Set to 1 if there is a carry from the least significant digit.

ADD long word to register — ADDL

R S,NS

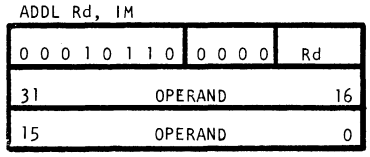
ADDL Rd, Rs



8

IM S,NS

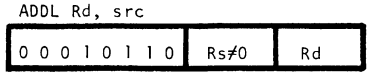
ADDL Rd, IM



14

IR S,NS

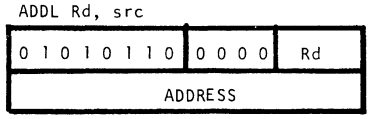
ADDL Rd, src



14

DA NS

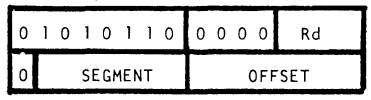
ADDL Rd, src



15

DA SSO

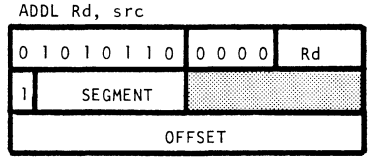
ADDL Rd, src



16

DA SLO

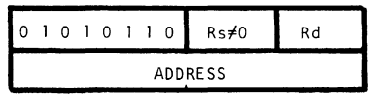
ADDL Rd, src



18

X NS

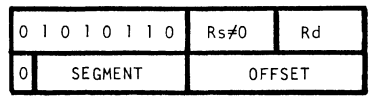
ADDL Rd, src



16

X SSO

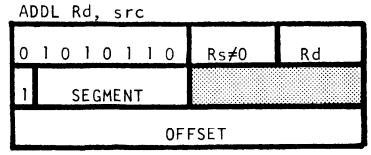
ADDL Rd, src



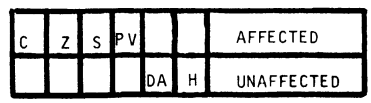
16

X SLO

ADDL Rd, src



19



Operation

dst<0:31> ← src<0:31> + dst<0:31>

Description

Source operand and destination operand long words are added together and the result is loaded into the destination. The contents of the source are not altered and the original contents of the destination are lost. The source is determined by the applicable addressing mode and the destination is always a general purpose register pair designated by the Rd field of the instruction.

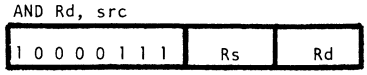
Flags:

C: Set to 1 if there is a carry from the most significant bit position of the long word.
Z: Set to 1 if result is zero. Reset otherwise.
S: Set to 1 if result is negative. Reset otherwise.
P/V: Set to 1 on arithmetic overflow.

AND word with register

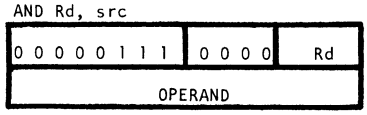
AND

R S,NS — AND Rd, src



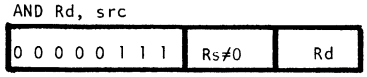
4

IM S,NS — AND Rd, src



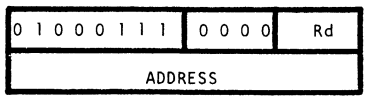
7

IR S,NS — AND Rd, src



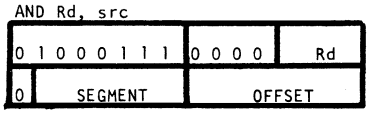
7

DA NS — AND Rd, src



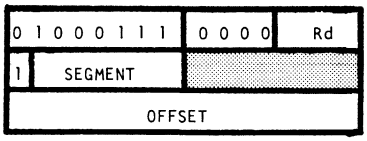
9

DA SSO — AND Rd, src



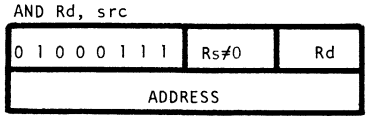
10

DA SLO — AND Rd, src



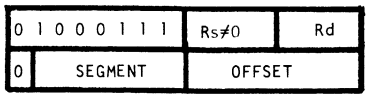
12

X NS — AND Rd, src



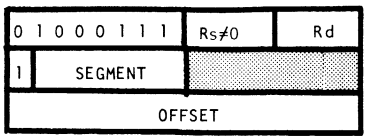
10

X SSO — AND Rd, src

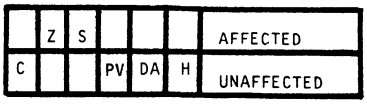


10

X SLO — AND Rd, src



13



Operation

dst<0:15> ← dst<0:15>∧src<0:15>

Description

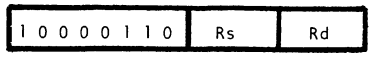
A logical AND operation is performed between the corresponding bits of the source and destination words. The source operand is determined by the applicable addressing mode, while the destination operand is always a general purpose word register, designated by the Rd field of the instruction. The result of the operation is loaded into the destination, whose original contents are lost. The source contents are not altered.

Flags:

Z: Set to 1 if result is zero. Reset otherwise.
S: Set to 1 if result is negative. Reset otherwise.

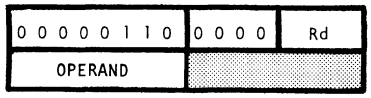
AND byte with register **ANDB**

R S,NS — ANDB Rd, src



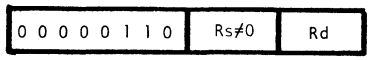
4

IM S,NS — ANDB Rd, src



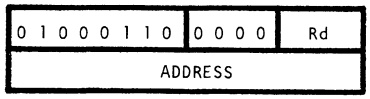
7

IR S,NS — ANDB Rd, src



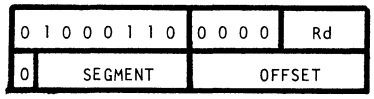
7

DA NS — ANDB Rd, src



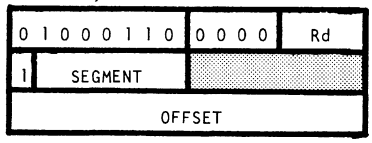
9

DA SSO — ANDB Rd, src



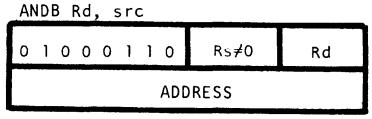
10

DA SLO — ANDB Rd, src



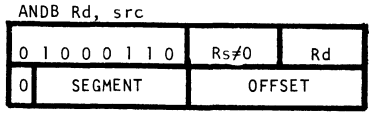
12

X NS — ANDB Rd, src



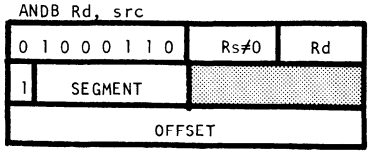
10

X SSO — ANDB Rd, src

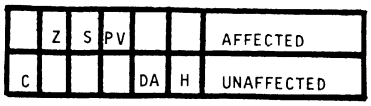


10

X SLO — ANDB Rd, src



13



Operation

dst<0:7> ← dst<0:7>∧src<0:7>

Description

A logical AND operation is performed between the corresponding bits of the source and destination bytes. The source operand is determined by the applicable addressing mode, while the destination operand is always a general purpose byte register, designated by the Rd field of the instruction. The result of the operation is loaded into the destination, whose original contents are lost. The source contents are not altered.

Flags:

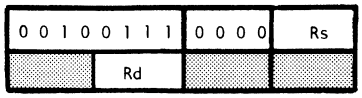
Z: Set to 1 if result is 0. Reset otherwise.
S: Set to 1 if result is negative. Reset otherwise.
P/V: Set to 1 if parity of result is even. Reset otherwise.

BIT test in a word (dynamic)

BIT

BIT dst, Rs

R S,NS

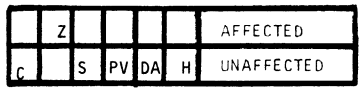


10

Operation

Description

The selected bit of the word destination register is tested and the Z flag is affected. The destination word operand is the general purpose register designated by the Rd field of the instruction. The bit to be tested is determined from a binary decode of the least significant 4 bits of a general purpose word register. This register is designated by the Rs field. The contents of the destination are unaltered.



Flags:

Z: Set to 1 if selected bit of destination operand is zero. Reset otherwise.

BIT test in a word (static)

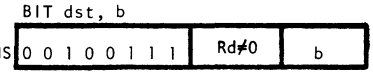
BIT

R — S,NS — 4

BIT dst, b

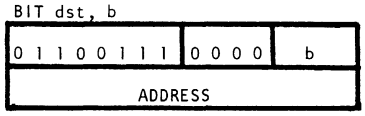
IR — S,NS — 8

BIT dst, b



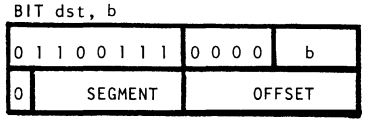
DA — NS — 10

BIT dst, b



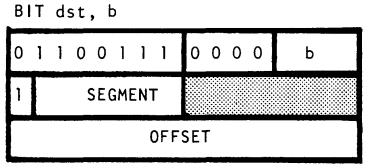
DA — SSO — 11

BIT dst, b



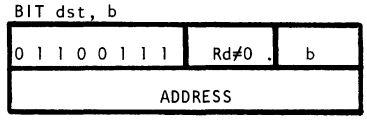
DA — SLO — 13

BIT dst, b



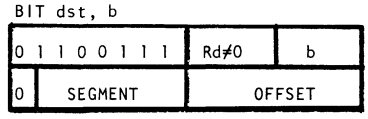
X — NS — 11

BIT dst, b



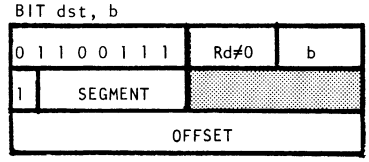
X — SSO — 11

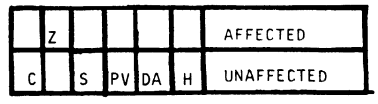
BIT dst, b



X — SLO — 14

BIT dst, b





Operation

Description

A bit in the word destination is tested, and the Z flag is affected as shown below.

The destination is determined by the applicable addressing mode, while the bit to be tested is specified by the b field of the instruction. The contents of the destination are not altered.

Flags:

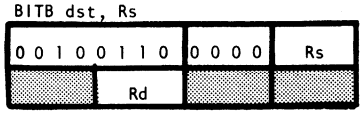
Z: Set to 1 if specified bit of destination word is Ø. Reset otherwise.

BIT test in a byte (dynamic)

BITB

R S,NS

BITB dst, Rs



10

Operation

Description

The selected bit of the byte destination register is tested and the Z flag is affected. The destination byte operand is the general purpose register designated by the Rd field of the instruction. The bit to be tested is determined from a binary decode of the least significant 3 bits of a general purpose word register. This register is designated by the Rs field of the instruction. The contents of the destination are unaltered.

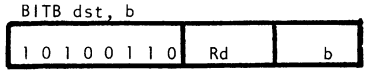
	Z					AFFECTED
C		S	PV	DA	H	UNAFFECTED

Flags:

Z: Set to 1 if selected bit of destination operand is zero. Reset otherwise.

R S,NS

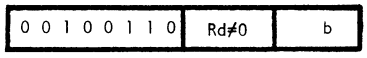
BITB dst, b



4

IR S,NS

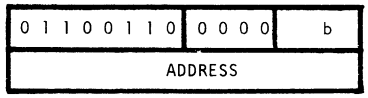
BITB dst, b



8

DA NS

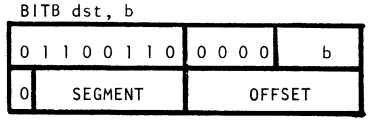
BITB dst, b



10

DA SSO

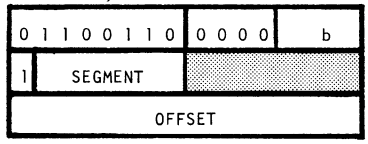
BITB dst, b



11

DA SLO

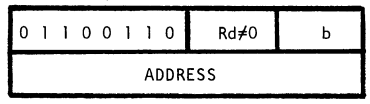
BITB dst, b



13

X SLO

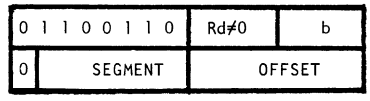
BITB dst, b



11

X NS

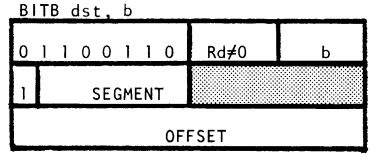
BITB dst, b



11

X SSO

BITB dst, b

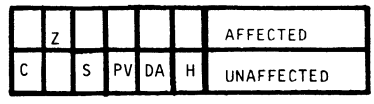


14

Operation

Description

A bit in the byte destination is tested, and the Z flag is affected as shown below. The destination is determined by the applicable addressing mode. The bit to be tested is determined by the **binary value of the least significant three bits of the b field of the instruction. The contents of the destination are not altered.**

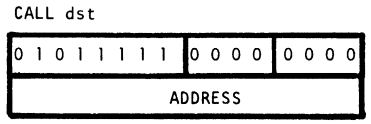
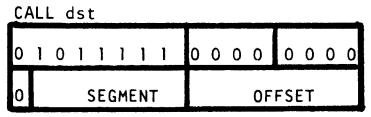
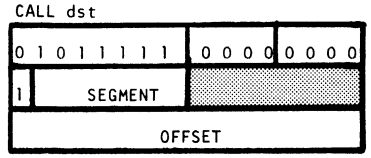
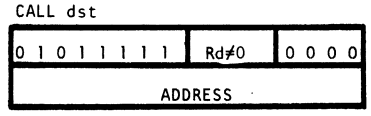
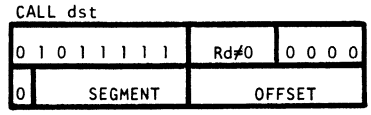
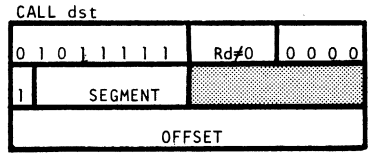


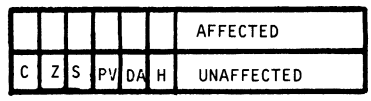
Flags:

Z: Set to 1 if specified bit of destination byte is Ø. Reset otherwise.

CALL subroutine

CALL

IR	S,NS		15,10
DA	NS		12
DA	SSO		18
DA	SLO		20
X	NS		13
X	SSO		18
X	SLO		21



Flags are not affected.

Operation (segmented)

R15<0:15> ← R15<0:15>- 2
(RR14<0:22>) ← Updated PC offset
R15<0:15> ← R15<0:15>- 2
(RR14<0:22>) ← PC SEGMENT
PC SEGMENT ← dst<24:30>
PC OFFSET ← dst<0:15>

Operation (non segmented)

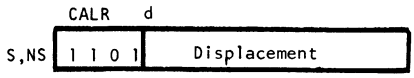
R15<0:15> ← R15<0:15>- 2
(R15<0:15>) ← Updated PC
PC ← dst<0:15>

Description

The program return address (i.e. the updated contents of PC) is pushed onto the stack addressed by the implied stack pointer register (R15 non segmented, RR14 segmented). The new program counter address is then loaded to transfer control to the subroutine. The new address is determined by the applicable addressing mode.

CALL subroutine relative **CALR**

RA



Operation (segmented)

R15<0:15> ← R15<0:15> -2
(RR14<0:22>) ← Updated PC OFFSET
R15<0:15> ← R15<0:15> -2
(RR14<0:22>) ← PC Segment
PC OFFSET ← Updated PC OFFSET +
2x displacement

Operation (non segmented)

R15<0:15> ← R15<0:15> -2
(R15<0:15>) ← Updated PC
PC ← Updated PC + 2 x displacement

Description

The program return address is pushed onto the stack addressed by the implied stack pointer register (R15 non segmented, RR14 segmented). The signed 12 bit displacement field of the instruction is sign extended and left shifted (word aligned) before being added to the return address. The result is then loaded into the program counter to produce a jump address. The program counter segment number remains unaltered. The range of the relative call is +2047 to -2048 words with respect to the updated PC.

						AFFECTED
C	Z	S	PV	DA	H	UNAFFECTED

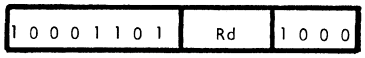
Flags are not affected.

CLEAR word

CLR

R S,NS

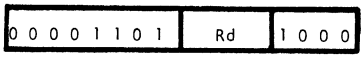
CLR dst



7

IR S,NS

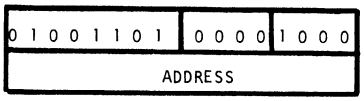
CLR dst



8

DA NS

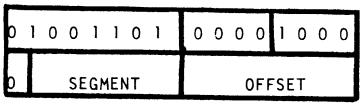
CLR dst



11

DA SSO

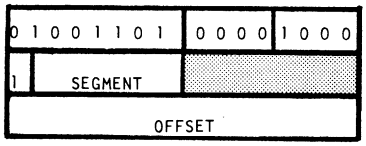
CLR dst



12

DA SLO

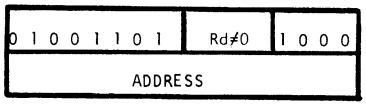
CLR dst



14

X NS

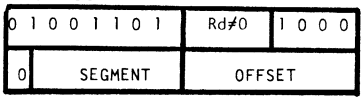
CLR dst



12

X SSO

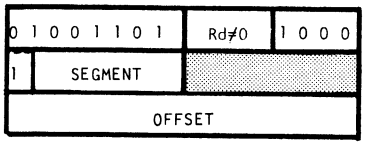
CLR dst



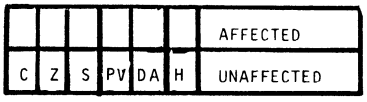
12

X SLO

CLR dst



15



Operation

dst <Ø:15> ← Ø

Description

The 16 bits of the specified destination word are replaced with zeros. The original contents of the destination are lost. The destination is determined by the applicable addressing mode.

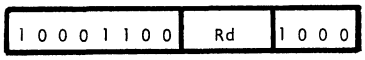
Flags are not affected.

CLEAR byte

CLRB

CLRB dst

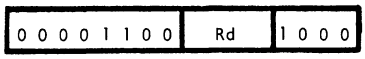
R S,NS



7

CLRB dst

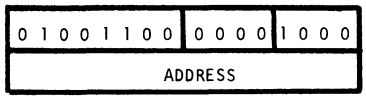
IR S,NS



8

CLRB dst

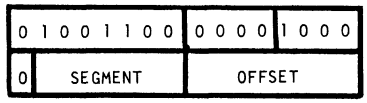
DA NS



11

CLRB dst

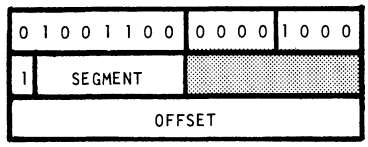
DA SSO



12

CLRB dst

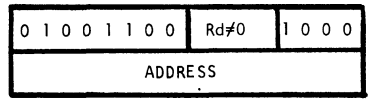
DA SLO



14

CLRB dst

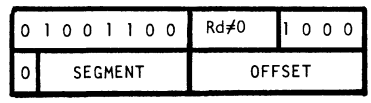
X NS



12

CLRB dst

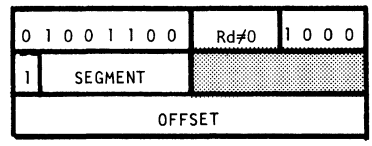
X SSO



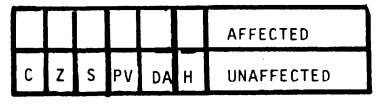
12

CLRB dst

X SLO



15



Flags are not affected.

Operation

dst <Ø:7 > ← Ø

Description

The 8 bits of the specified destination byte are replaced with zeros. The original contents of the destination are lost.

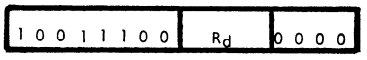
The destination is determined by the applicable addressing mode.

CLEAR long word

CLRL

R S,NS

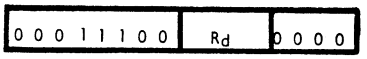
CLRL dst



5

IR S,NS

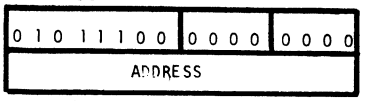
CLRL dst



11

DA NS

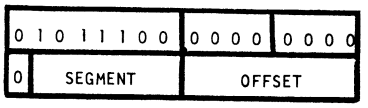
CLRL dst



14

DA SSO

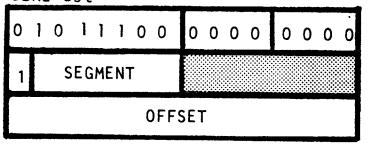
CLRL dst



15

DA SLO

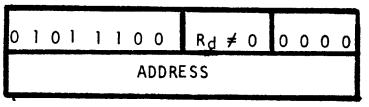
CLRL dst



17

X NS

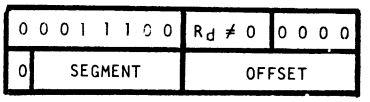
CLRL dst



15

X SSO

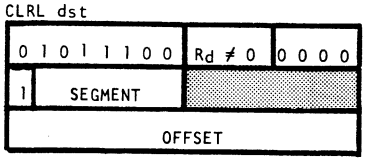
CLRL dst



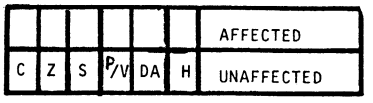
15

X SLO

CLRL dst



18



Flags are not affected.

Operation

dst<0:31> ← Ø

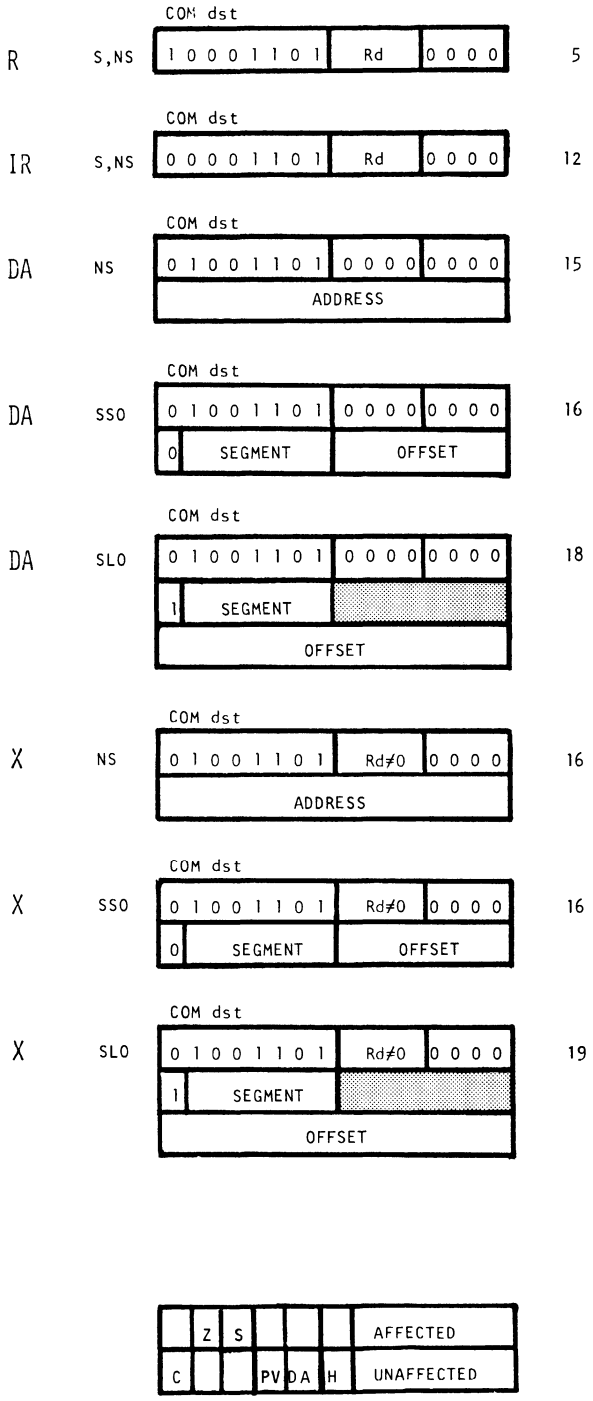
Description

The 32 bits of the specified destination are replaced with zeros. The original contents of the destination are lost.

The destination is determined by the applicable addressing mode.

COMPLEMENT word

COM



Operation

$dst\langle 0:15\rangle \leftarrow \overline{dst\langle 0:15\rangle}$

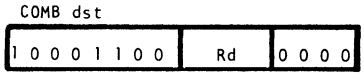
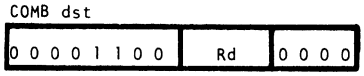
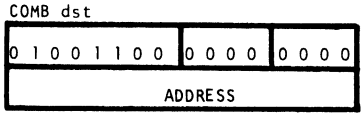
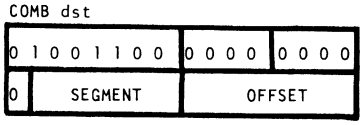
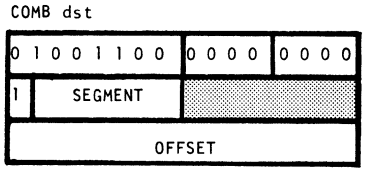
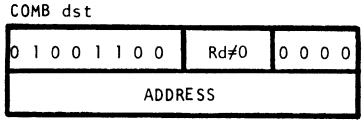
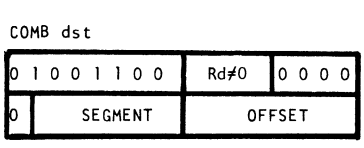
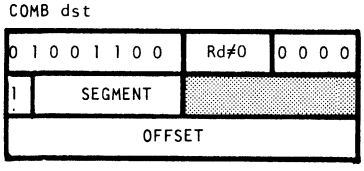
Description

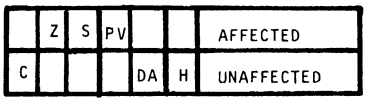
The contents of the destination word operand are complemented. The destination operand is determined by the applicable addressing mode.

Flags:

Z: Set to 1 if result is zero. Reset otherwise.
S: Set to 1 if result is negative. Reset otherwise.

COMPLEMENT byte — COMB

R	S,NS		5
IR	S,NS		12
DA	NS		15
DA	SSO		16
DA	SLO		18
X	NS		16
X	SSO		16
X	SLO		19



Operation

dst<0:7> ← $\overline{\text{dst}}$<0:7>

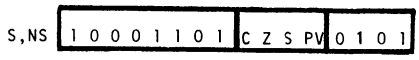
Description

The contents of the destination byte operand are complemented. The destination operand is determined by the applicable addressing mode.

Flags:

Z: Set to 1 if result is zero. Reset otherwise.
S: Set to 1 if result is negative. Reset otherwise.
P/V: Set to 1 if parity of result is even. Reset otherwise.

COMFLG



7

Operation

Description

The CPU flags C,Z,S and P/V are complemented or unaltered, according to the bit settings in the instruction field as described in the table below.

Instruction bit	if = Ø	if 1
7	no effect	complement C flag
6	no effect	complement Z flag
5	no effect	complement S flag
4	no effect	complement P/V flag

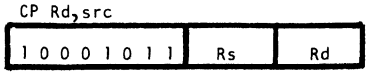
Flags:

See above

C	Z	S	PV			AFFECTED
				DA	H	UNAFFECTED

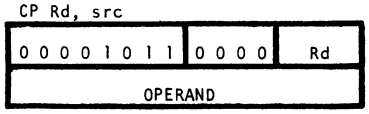
COMPARE register with word CP

R S,NS — CP Rd,src



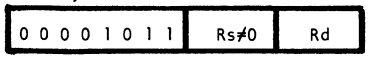
4

IM S,NS — CP Rd, src



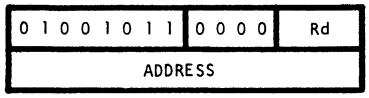
7

IR S,NS — CP Rd, src



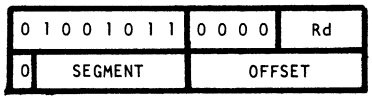
7

DA NS — CP Rd, src



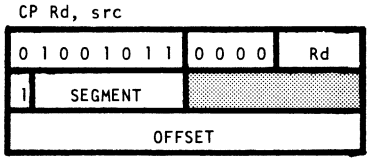
9

DA SSO — CP Rd, src



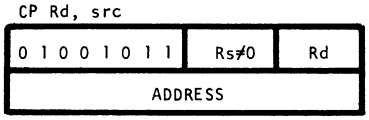
10

DA SLO — CP Rd, src



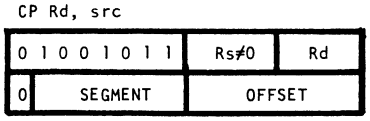
12

X NS — CP Rd, src



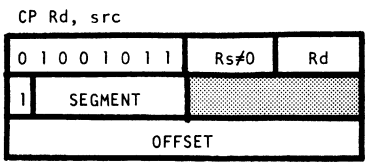
10

X SSO — CP Rd, src

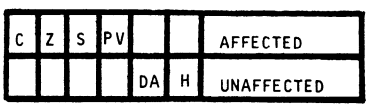


10

X SLO — CP Rd, src



13



Operation

Use the result of Rd<0:15>- src<0:15> to set flags.

Description

The source word operand is compared by subtraction with the contents of a general purpose word register designated by the Rd field of the instruction. The source operand is determined by the applicable addressing mode. Both the source contents and destination contents are unaltered.

Flags:

C: Reset on carry from most significant bit of result. Otherwise set to 1, indicating a borrow.
Z: Set to 1 if result is zero. Reset otherwise.
S: Set to 1 if result is negative. Reset otherwise.
P/V: Set to 1 on arithmetic overflow. Reset otherwise.

COMPARE register with byte **CPB**

CLOCK CYCLES

R	S,NS	CPB Rd, src 1 0 0 0 1 0 1 0 | Rs | Rd	4
IM	S,NS	CPB Rd, src 0 0 0 0 1 0 1 0 | 0 0 0 0 | Rd OPERAND	7
IR	S,NS	CPB Rd, src 0 0 0 0 1 0 1 0 | Rs≠0 | Rd	7
DA	NS	CPB Rd, src 0 1 0 0 1 0 1 0 | 0 0 0 0 | Rd ADDRESS	9
DA	SSO	CPB Rd, src 0 1 0 0 1 0 1 0 | 0 0 0 0 | Rd 0 | SEGMENT | OFFSET	10
DA	SLO	CPB Rd, src 0 1 0 0 1 0 1 0 | 0 0 0 0 | Rd 1 | SEGMENT OFFSET	12
X	NS	CPB Rd, src 0 1 0 0 1 0 1 0 | Rs≠0 | Rd ADDRESS	10
X	SSO	CPB Rd, src 0 1 0 0 1 0 1 0 | Rs≠0 | Rd 0 | SEGMENT | OFFSET	10
X	SLO	CPB Rd, src 0 1 0 0 1 0 1 0 | Rs≠0 | Rd 1 | SEGMENT OFFSET	13

C	Z	S	P/V			AFFECTED
				DA	H	UNAFFECTED

Operation

Use result of Rd<0:7>- src<0:7>

to set flags.

Description

The source byte operand is compared by subtraction with the contents of a general purpose byte register designated by the Rd field of the instruction. The source operand is determined by the applicable addressing mode. Both the source contents and destination contents are unaltered.

Flags:

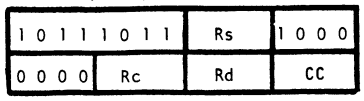
C: Reset on carry from most significant bit of result. Otherwise set to 1, indicating a borrow.
Z: Set to 1 if result is zero. Reset otherwise.
S: Set to 1 if result is negative. Reset otherwise.
P/V: Set to 1 on arithmetic overflow. Reset otherwise.

COMPARE register to memory word, autodecrement

CPD

CPD dst, src, Rc, CC

IR S,NS



20

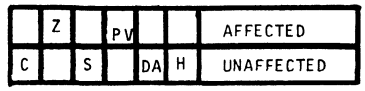
Operation

If result of dst<0:15>- src<0:15> meets CC condition in instruction.
Z flag ← 1

Rs<0:15> ← Rs<0:15>- 2
Rc<0:15> ← Rc<0:15>- 1

Description

The source word operand is compared to the destination word operand by subtraction. The destination operand is the contents of the general purpose word register designated by the Rd field of the instruction. The source operand is a word in memory addressed by the general purpose register designated by the Rs field of the instruction. Both source and destination operands are unaltered, and the only action is to set the flags. The contents of the general purpose register designated by the Rc field of the instruction are decremented by 1. The contents of Rs are decremented by 2.



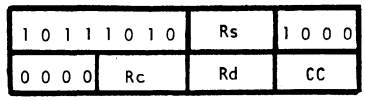
Flags:

Z: Set to 1 if a comparison matches condition specified in CC field. Reset otherwise.
P/V: Set to 1 if result of decrementing Rc is zero. Reset otherwise.

COMPARE register to memory byte, autodecrement — CPDB

CPDB dst, src, Rc, CC

IR S,NS



20

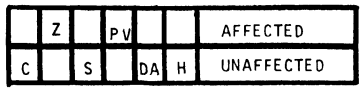
Operation

If result of dst<0:7>- src<0:7>
meets CC condition in instruction
Z flag ← 1

Rs<0:15> ← Rs<0:15>- 1
Rc<0:15> ← Rc<0:15>- 1

Description

The source byte operand is compared to the destination byte operand by subtraction. The destination operand is the contents of the general purpose byte register designated by the Rd field of the instruction. The source operand is a byte in memory addressed by the general purpose register designated by the Rs field of the instruction. Both source and destination operands are unaltered and the only action is to set the flags. The contents of the general purpose register designated by the Rc field of the instruction are decremented by 1. The contents of Rs are decremented by 1.



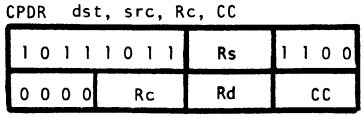
Flags:

Z: Set to 1 if a comparison matches condition specified in CC field. Reset otherwise.
P/V: Set to 1 if result of decrementing Rc is zero. Reset otherwise.

COMPARE register to memory word, autodecrement and repeat — CPDR

CPDR dst, src, Rc, CC

IR S,NS



11 + 9n*

*n is the number of iterations

Operation

If dst<0:15>- src<0:15>meets CC condition in instruction.

Z flag ← 1

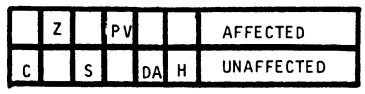
Rs<0:15> ← Rs<0:15>- 2

Rc<0:15> ← Rc<0:15>- 1

repeat until termination

Description

The source word operand is compared to the destination word operand by subtraction. The source operand is a word in memory addressed by the general purpose register designated by the Rs field of the instruction. The destination operand is the contents of the general purpose word register designated by the Rd field of the instruction. Both source and destination operands are unaltered and the only action is to set the flags. The contents of the general purpose register designated by the Rc field of the instruction are decremented by 1. The contents of Rs are decremented by 2, and the operation will repeat until termination. Termination occurs when either the contents of Rc are Ø or CC condition is met. This instruction is interruptible.



Flags:

Z: Set to 1 if a comparison matches condition specified in CC field. Reset otherwise.

P/V: Set to 1 if result of decrementing Rc is zero. Reset otherwise.

CPDRB dst, src, Rc, CC

IR S,NS

1 0 1 1 1 0 1 0	Rs	1 1 0 0
0 0 0 0	Rc	Rd	CC

11 + 9n*

*n is the number of iterations

Operation

If dst<0:7>- src<0:7>meets CC condition in instruction.
Z flag ← 1

Rs<0:15> ← Rs<0:15>- 1
Rc<0:15> ← Rc<0:15>- 1
repeat until termination

Description

The source byte operand is compared to the destination byte operand by subtraction. The source operand is a byte in memory addressed by the general purpose register designated by the Rs field of the instruction. The destination operand is the contents of the general purpose byte register designated by the Rd field of the instruction. Both source and destination operands are unaltered and the only action is to set the flags. The contents of the general purpose register designated by the Rc field of the instruction are decremented by 1. The contents of Rs are decremented by 1 and th operation will repeat until termination. Termination occurs when either the contents of Rc are Ø or CC condition is met. This instruction is interruptible.

	Z		PV			AFFECTED
C		S		DA	H	UNAFFECTED

Flags:

Z: Set to 1 if a comparison matches condition specified in CC field. Reset otherwise.
P/V: Set to 1 if result of decrementing Rc is zero. Reset otherwise.

COMPARE register to memory word, autoincrement — CPI

CPI dst, src, Rc, CC

IR S,NS

1 0 1 1 1 0 1 1	Rs	0 0 0 0
0 0 0 0	Rc	Rd	CC

20

Operation

If result of dst<0:15>- src<0:15> meets CC condition in instruction.

Z flag ← 1

Rs<0:15> ← Rs<0:15>+ 2

Rc<0:15> ← Rc<0:15>- 1

Description

The source word operand is compared to the destination word operand by subtraction. The destination operand is the contents of the general purpose word register designated by the Rd field of the instruction. The source operand is a word in memory addressed by the general purpose register designated by the Rs field of the instruction. Both the source and destination operands are unaltered and the only action is to set the flags. The contents of the general purpose register designated by the Rc field of the instruction are decremented by 1. The contents of Rs are incremented by 2.

	Z		PV			AFFECTED
C		S		DA	H	UNAFFECTED

Flags:

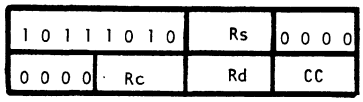
Z: Set to 1 if a comparison matches condition specified in CC field. Reset otherwise.

P/V: Set to 1 if result of decrementing Rc is zero. Reset otherwise.

COMPARE register to memory byte, autoincrement. CPIB

CPIB dst, src, Rc, CC

IR S,NS



20

Operation

If result of dst<0:7>- src<0:7>
meets CC conditon in instruction.
Z flag ← 1

Rs<0:15> ← Rs<0:15>+ 1
Rc<0:15> ← Rc<0:15>- 1

Description

The source byte operand is compared to the destination byte operand by subtraction. The destination operand is the contents of the general purpose byte register designated by the Rd field of the instruction. The source operand is a byte in memory addressed by the general purpose register designated by the Rs field of the instruction. Both the source and destination operands are unaltered, and the only action is to set the flags. The contents of the general purpose register designated by the Rc field of the instruction are decremented by 1. The contents of Rs are incremented by 1.

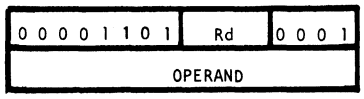
	Z		PV			AFFECTED
C		S		DA	H	UNAFFECTED

Flags:

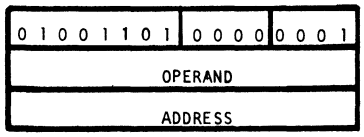
Z: Set to 1 if a comparison matches condition specified in CC field. Reset otherwise.
P/V: Set to 1 if result of decrementing Rc is zero. Reset otherwise.

COMPARE IMMEDIATE word with memory — CP

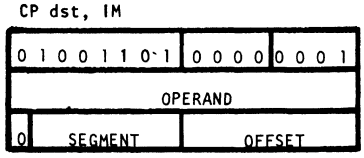
IR S,NS — CP dst, IM — 11



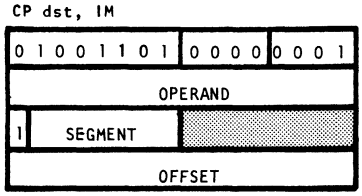
DA NS — CP dst, IM — 14



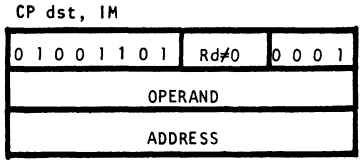
DA SSO — CP dst, IM — 15



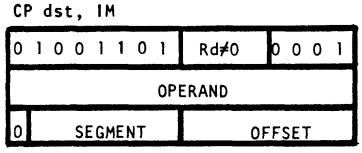
DA SLO — CP dst, IM — 17



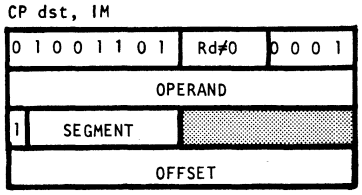
X NS — CP dst, IM — 15

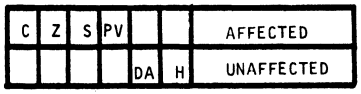


X SSO — CP dst, IM — 15



X SLO — CP dst, IM — 18





Operation

Use result of dst<0:15>- src<0:15> to set flags (see below).

Description

The immediate source word operand is compared with the destination word operand. The comparison is achieved by subtraction. The destination operand is determined by the applicable addressing mode. The contents of the destination operand are unaltered and the only action is to set the flags as described below.

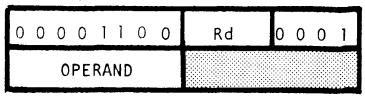
Flags:

C: Reset on carry from most significant bit of result. Otherwise set to 1, indicating a borrow.
Z: Set to 1 if result is zero. Reset otherwise.
S: Set to 1 if result is negative. Reset otherwise.
P/V: Set to 1 on arithmetic overflow. Reset otherwise.

COMPARE IMMEDIATE byte with memory — CP

IR

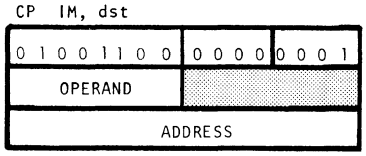
CP IM, dst



11

DA NS

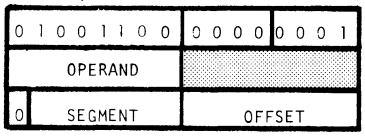
CP IM, dst



14

DA SSO

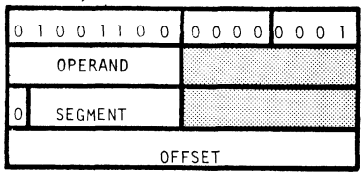
CP IM, dst



15

DA SLO

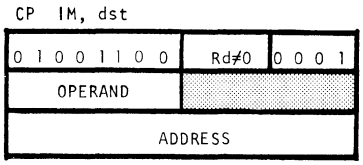
CP IM, dst



17

X NS

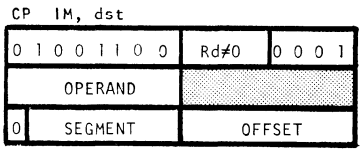
CP IM, dst



15

X SSO

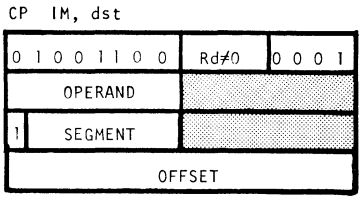
CP IM, dst



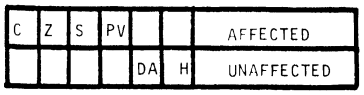
15

X SLO

CP IM, dst



18



Operation

Use result of dst<0:7>- src<0:7> to set flags.

Description

The immediate source byte operand is compared with the destination byte operand. The destination operand is determined by the applicable addressing mode. The contents of the destination operand are unaltered.

Flags:

C: Reset on carry from most significant bit of result. Otherwise set to 1, indicating a borrow.
Z: Set to 1 if result is 0. Reset otherwise.
S: Set to 1 if result is negative. Reset otherwise.
P/V: Set to 1 on arithmetic overflow. Reset otherwise.

COMPARE register to memory word autoincrement and repeat

CPIR

CPIR dst, src, Rc, CC

IR S,NS

1 0 1 1 1 0 1 1	Rs	0 1 0 0
0 0 0 0	Rc	Rd	CC

11 + 9n*

*n is the number of iterations

Operation

If dst<0:15>- src<0:15>meets CC condition in instruction.

Z flag ← 1

Rs<0:15> ← Rs<0:15>+ 2

Rc<0:15> ← Rc<0:15>- 1

repeat until termination

Description

The source word operand is compared to the destination word operand by subtraction. The source operand is a word in memory addressed by the general purpose register designated by the Rs field of the instruction. The destination operand is the content of the general purpose word register designated by the Rd field of the instruction. Both source and destination operands are unaltered and the only action is to set the flags. The contents of the general purpose register designated by the Rc field of the instruction are decremented by 1. The contents of Rs are incremented by 2. The operation will repeat until termination. Termination occurs when either the contents of Rc are Ø or CC condition is met. This instruction is interruptible.

	Z		PV			AFFECTED
C		S		DA	H	UNAFFECTED

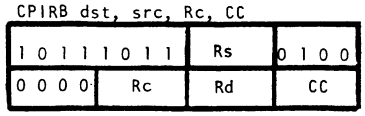
Flags:

Z: Set to 1 if a comparison matches condition specified in CC field. Reset otherwise.

P/V: Set to 1 if result of decrementing Rc is zero. Reset otherwise.

IR S,NS

CPIRB dst, src, Rc, CC



11 + 9n*

*n is the number of iterations

Operation

If dst<0:7>- src<0:7>meets CC condition in instruction.

Z flag ← 1

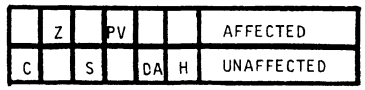
Rs<0:15> ← Rs<0:15>+ 1

Rc<0:15> ← Rc<0:15>- 1

repeat until termination

Description

The source byte operand is compared to the destination byte operand by subtraction The source operand is a byte in memory addressed by the general purpose register designated by the Rs field of the instruction. The destination operand is the contents of the general purpose byte register designated by the Rd field of the instruction. Both the source and destination operands are unaltered and the only action is to set the flags. The contents of the general purpose register designated by the Rc field of the instruction are decremented by 1. The contents of Rs are incremented by 1, and the operation will repeat until termination. Termination occurs when either the contents of Rc are Ø or CC condition is met. This instruction is interruptible.



Flags:

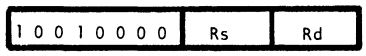
Z: Set to 1 if a comparison matches condition specified in CC field. Reset otherwise.

P/V: Set to 1 if result of decrementing Rc is zero. Reset otherwise.

COMPARE register with long word. CPL

CPL Rd, src

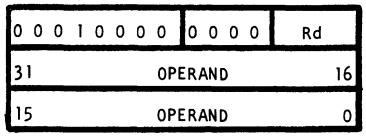
R S,NS



8

CPL Rd, src

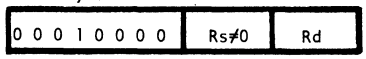
IM S,NS



14

CPL Rd, src

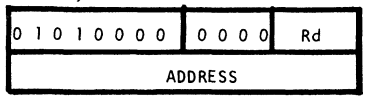
IR S,NS



14

CPL Rd, src

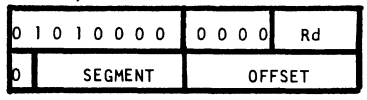
DA NS



15

CPL Rd, src

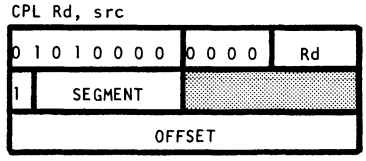
DA SSO



16

CPL Rd, src

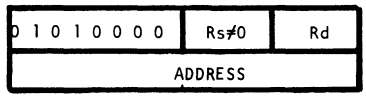
DA SLO



18

CPL Rd, src

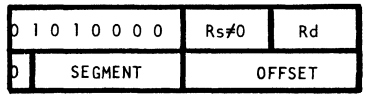
X NS



16

CPL Rd, src

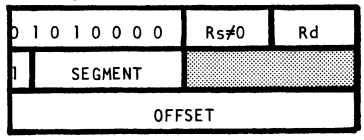
X SSO



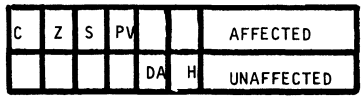
16

CPL Rd, src

X SLO



19



Operation

Use result of Rd<0:31>- src<0:31>

to set flags.

Description

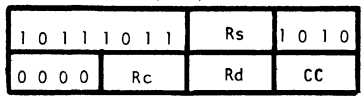
The source long word operand is compared by subtraction with the contents of a general purpose register pair designated by the Rd field of the instruction. The source operand is determined by the applicable addressing mode. Both the source contents and destination contents are unaltered.

Flags:

C: Reset on carry from most significant bit of result. Otherwise set to 1, indicating a borrow.
Z: Set to 1 if result is zero. Reset otherwise.
S: Set to 1 if result is negative. Reset otherwise.
P/V: Set to 1 on arithmetic overflow. Reset otherwise.

CPSD dst, src, Rc, CC

IR S,NS



25

Operation

If result of dst<0:15>- src<0:15>

meets CC condition in instruction,

Z flag ← 1

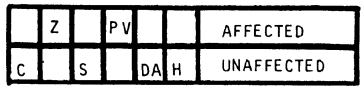
Rs<0:15> ← Rs<0:15>-2

Rd<0:15> ← Rd<0:15>-2

Rc<0:15> ← Rc<0:15>-1

Description

The source word operand is compared to the destination word operand. Both the source and destination operands are words in memory addressed by the general purpose registers designated in the Rd and Rs fields of the instruction. The comparison is achieved by subtraction. The contents of the general purpose register designated by the Rc field of the instruction are decremented by 1. The source and destination operands are unaltered. The contents of the Rs and Rd registers are decremented by 2.



Flags:

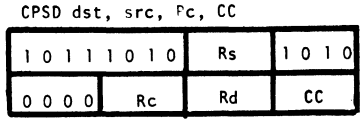
Z: Set to 1 if a comparison matches condition specified in CC field. Reset otherwise.

P/V: Set to 1 if result of decrementing Rc is zero. Reset otherwise.

COMPARE byte strings in memory, autodecrement — CPSDB

CPSD dst, src, Rc, CC

IR S,NS



25

Operation

If result of Dst<0:7> - Src<0:7>

meets CC condition, Z flag ← 1

Rs<0:15> ← Rs<0:15> - 1

Rd<0:15> ← Rd<0:15> - 1

Rc<0:15> ← Rc<0:15> - 1

Description

The source byte operand is compared to the destination byte operand. Both the source and destination operands are bytes in memory addressed by the general purpose registers designated in the Rd and Rs fields of the instruction. The comparison is achieved by subtraction. The contents of the general purpose register designated by the Rc field of the instruction are decremented by 1. The contents of Rs and Rd are both decremented by 1. The source and destination operands are unaltered. The contents of the Rs and Rd registers are decremented by 1.

	Z		PV			AFFECTED
C		S		DA	H	UNAFFECTED

Flags:

Z: Set to 1 if a comparison matches condition specified in CC field. Reset otherwise.

P/V: Set to 1 if result of decrementing Rc is zero. Reset otherwise.

COMPARE word strings in memory, autodecrement and repeat — CPSDR

CPSDR dst, src, Rc, CC

IR S,NS

1 0 1 1 1 0 1 1	Rs	1 1 1 0
0 0 0 0 Rc	Rd	CC

11 + 14n*

*n = number of iterations

Operation

If result of dst<0:15>- src<0:15>

meets CC condition in instruction

Z flag ← 1

Rs<0:15> ← Rs<0:15>- 2

Rd<0:15> ← Rd<0:15>- 2

Rc<0:15> ← Rc<0:15>- 1

repeat until termination

Description

The source word operand is compared to the destination word operand. Both the source and destination operands are words in memory addressed by the general purpose registers designated in the Rs and Rd fields of the instruction. The comparison is achieved by subtraction. Both source and destination operands are unaltered. The contents of the general purpose register designated by the Rc field of the instruction are decremented by 1. The contents of the Rs and Rd registers are both decremented by 2. The operation will repeat until termination. Termination occurs when either the contents of Rc are Ø or CC condition is met. This instruction is interruptible.

	Z		PV			AFFECTED
C		S		DA	H	UNAFFECTED

Flags:

Z: Set to 1 if a comparison matches condition specified in CC field. Reset otherwise.

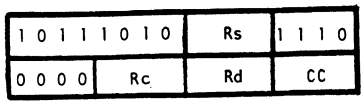
P/V: Set to 1 if result of decrementing Rc is zero. Reset otherwise.

COMPARE byte strings in memory autodecrement and repeat — CPSDRB

CPSDRB dst, src, Rc, CC

IR

S,NS



11 + 14n*

*n = number of iterations

Operation

If result of Dst<0:7> - Src<0:7> meets CC condition, Z flag ← 1

Rs<0:15> ← Rs<0:15> - 1

Rd<0:15> ← Rd<0:15> - 1

Rc<0:15> ← Rc<0:15> - 1

repeat until termination

Description

The source byte operand is compared to the destination byte operand. Both the source and destination operands are bytes in memory addressed by the general purpose registers designated in the Rs and Rd fields of the instruction. The comparison is achieved by subtraction. Both source and destination operands are unaltered and the only action is to set the flags. The contents of the general purpose register designated by the Rc field of the instruction are decremented by 1. The contents of the Rs and Rd registers are both decremented by 1. The operation will repeat until termination. Termination occurs when either: The contents of Rc are Ø (string exhausted) or CC condition is met. This instruction is interruptible.

	Z		PV			AFFECTED
C		S		DA	H	UNAFFECTED

Flags:

Z: Set to 1 if a comparison matches condition specified in CC field. Reset otherwise.

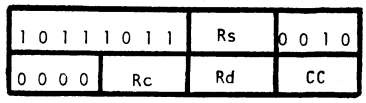
P/V: Set to 1 if result of decrementing Rc is zero. Reset otherwise.

COMPARE word strings in memory, auto-increment

CPSI

CPSI dst, src, Rc, CC

IR S,NS



25

Operation

If result of dst<0:15> - src<0:15>

meets CC condition in instruction.

Z flag ← 1

Rs<0:15> ← Rs<0:15>+ 2

Rd<0:15> ← Rd<0:15>+ 2

Rc<0:15> ← Rc<0:15>-1

Description

The source word operand is compared to the destination word operand. Both the source and destination operands are words in memory addressed by the general purpose registers designated in the Rs and Rd fields of the instruction. The comparison is achieved by subtraction. The contents of the general purpose register designated by the Rc field of the instruction are decremented by 1. The source and destination operands are unaltered. The contents of the Rs and Rd registers are incremented by 2.

	Z		PV			AFFECTED
C		S		DA	H	UNAFFECTED

Flags:

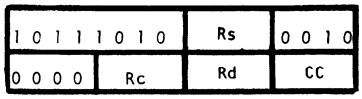
Z: Set to 1 if a comparison matches condition specified in CC field. Reset otherwise.

P/V: Set to 1 if result of decrementing Rc is zero. Reset otherwise.

COMPARE byte strings in memory, autoincrement **CPSIB**

CPSIB dst, src, Rc, CC

IR S,NS



25

Operation

If dst<0:7> - src<0:7>

meets CC condition in instruction

Z flag ← 1

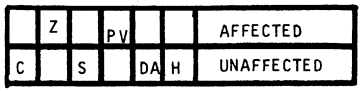
Rs<0:15> ← Rs<0:15>+ 1

Rd<0:15> ← Rd<0:15>+ 1

Rc<0:15> ← Rc<0:15>- 1

Description

The source byte operand is compared to the destination byte operand by subtraction. Both the source and destination operands are bytes in memory addressed by the general purpose registers designated in the Rd and Rs fields of the instruction. The comparison is achieved by subtraction. The contents of the general purpose register designated by the Rc field of the instruction are decremented by 1. Both source and destination operands are unaltered. The contents of the Rs and Rd registers are incremented by 1.



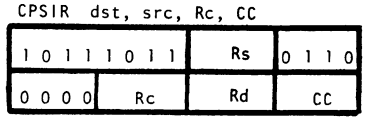
Flags:

Z: Set to 1 if a comparison matches condition specified in CC field. Reset otherwise.

P/V: Set to 1 if result of decrementing Rc is zero. Reset otherwise.

CPSIR dst, src, Rc, CC

IR S,NS



11 + 14n*

*n = number of iterations

Operation

If result of dst<0:15>- src<0:15> meets CC conditon in instruction.

Z flag ← 1

Rs<0:15> ← Rs<0:15>+ 2

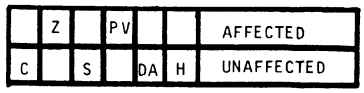
Rd<0:15> ← Rd<0:15>+ 2

Rc<0:15> ← Rc<0:15>- 1

repeat until termination

Description

The source word operand is compared to the destination word operand. Both the source and destination operands are words in memory addressed by the general purpose registers designated in the Rs and Rd fields of the instruction. The comparison is achieved by subtraction. Both source and destination operands are unaltered. The contents of the general purpose register designated by the Rc field of the instruction are decremented by 1. The contents of the Rs and Rd registers are both incremented by 2. The operation will repeat until termination. Termination occurs when either the contents of Rc are Ø or CC condition is met. This instruction is interruptible.



Flags:

Z: Set to 1 if a comparison matches condition specified in CC field. Reset otherwise.

P/V: Set to 1 if result of decrementing Rc is zero. Reset otherwise.

COMPARE byte strings, in memory autoincrement and repeat

CPSIRB

CPSIRB dst, src, Rc, CC

IR S,NS

1 0 1 1 1 0 1 0	Rs	0 1 1 0
0 0 0 0 Rc	Rd	CC

11 + 14n*

*n = number of iterations

Operation

If dst<0:7>- src<0:7> meets CC condition in instruction.

Z flag ← 1

Rs<0:15> ← Rs<0:15>+ 1

Rd<0:15> ← Rd<0:15>+ 1

Rc<0:15> ← Rc<0:15>- 1

repeat until termination

Description

The source byte operand is compared to the destination byte operand. Both the source and destination operands are bytes in memory addressed by the general purpose registers designated in the Rs and Rd fields of the instruction. The comparison is achieved by subtraction. Both source and destination operands are unaltered and the only action is to set the flags. The contents of the general purpose register designated by the Rc field of the instruction are decremented by 1. The contents of the Rs and Rd registers are both incremented by 1. The operation will repeat until termination. Termination occurs when either the contents of Rc are Ø or CC condition is met. This instruction is interruptible.

	Z		PV			AFFECTED
C		S		DA	H	UNAFFECTED

Flags:

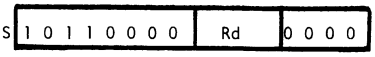
Z: Set to 1 if a comparison matches condition specified in CC field. Reset otherwise.

P/V: Set to 1 if result of decrementing Rc is zero. Reset otherwise.

DECIMAL ADJUST byte — DAB

R

DAB Rd



5

Operation

dst<0:7> ← dst<0:7>+ BCD<0:7>

Description

A destination byte register, designated by the Rd field of the instruction, is adjusted by the addition of the BCD operand given in the table below. This instruction converts a byte (binary representation) into a two digit binary coded decimal representation, following an arithmetic operation.

PRECEDING ARITHMETIC OPERATION	C FLAG BEFORE DAB	dst<4:7> (HEX)	H FLAG BEFORE DAB	dst<0:3> (HEX)	BCD<0:7>	C FLAG AFTER DAB
	0	0-9	0	0-9	00	0
	0	0-8	0	A-F	06	0
ADDB	0	0-9	1	0-3	06	0
ADCB	0	A-F	0	0-9	60	1
	0	9-F	0	A-F	66	1
	0	A-F	1	0-3	66	1
	1	0-2	0	0-9	60	1
	1	0-2	0	A-F	66	1
	1	0-3	1	0-3	66	1
	0	0-9	0	0-9	00	0
SUBB	0	0-8	1	6-F	FA	0
SBCB	1	7-F	0	0-9	A0	1
	1	6-F	1	6-F	9A	1

C	Z	S				AFFECTED
			PV	DA	H	UNAFFECTED

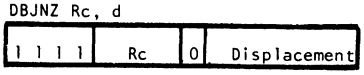
Flags:

C: Set or reset according to table.
Z: Set to 1 if result is zero. Reset otherwise.
S: Set to 1 if the most significant bit of the result is set. Reset otherwise.

DECREMENT byte register and jump on not zero — DBJNZ

DBJNZ Rc, d

RA



11

Operation

Rc<0:7> ← Rc<0:7> - 1

If Rc<0:7> - 1 ≠ 0

Then PC ← Updated Pc-2x displacement

Otherwise PC ← Updated PC

Description

The contents of the general purpose byte register designated by the Rc field of the instruction are decremented, and if this produces a non-zero result, a jump is executed. The jump address is obtained by subtracting the contents of the 7 bit displacement field, which has been left shifted (ie word aligned) from the contents of the updated program counter (ie incremented by 2). The resultant address is loaded into the program counter and is used as the jump destination. The instruction displacement field is interpreted as a 7 bit unsigned integer. Thus the range of the relative jump is Ø to -127 words with respect to the updated PC.

If the register decrementation produces a zero result, then the contents of the program counter are merely updated by incrementing by 2.

						AFFECTED
C	Z	S	PV	DA	H	UNAFFECTED

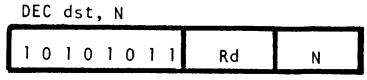
Flags are not affected.

DECREMENT word

DEC

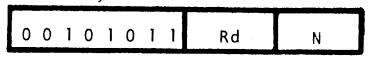
R — S,NS — 4

DEC dst, N



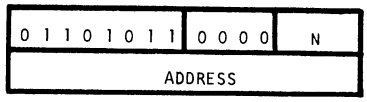
IR — S,NS — 11

DEC dst, N



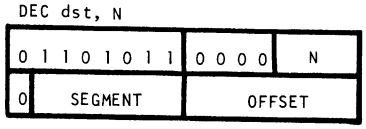
DA — NS — 13

DEC dst, N



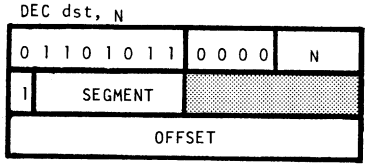
DA — SSO — 14

DEC dst, N



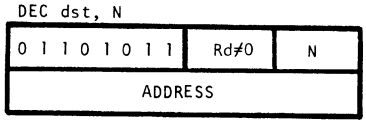
DA — SLO — 16

DEC dst, N



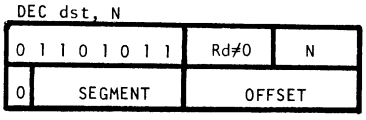
X — NS — 14

DEC dst, N



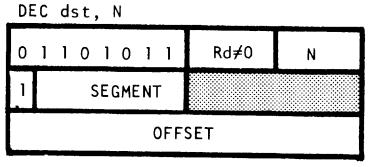
X — SSO — 14

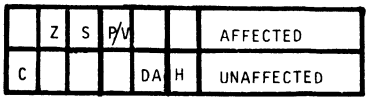
DEC dst, N



X — SLO — 17

DEC dst, N





Operation

dst<Ø:15> ← dst<Ø:15> - N - 1

Description

A value between 1 and 16 is subtracted from the destination operand word and the result is loaded back into the destination. The desired value to be subtracted is specified by the N field. N = Ø corresponds to value 1 and so on and N = F corresponds to value 16. The destination is determined by the applicable addressing mode.

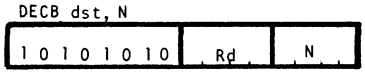
Flags:

Z: Set to 1 if result is zero. Reset otherwise.
S: Set to 1 if result is negative. Reset otherwise.
P/V: Set to 1 on arithmetic overflow. Reset otherwise.

DECREMENT byte **DECB**

R S,NS

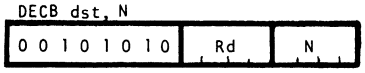
DECB dst, N



4

IR S,NS

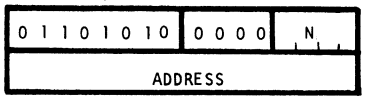
DECB dst, N



11

DA NS

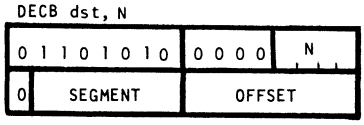
DECB dst, N



13

DA SSO

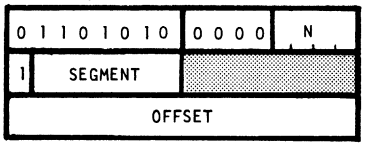
DECB dst, N



14

DA SLO

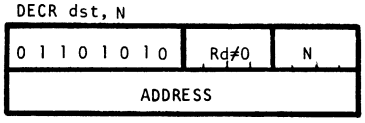
DECB ds, N



16

X NS

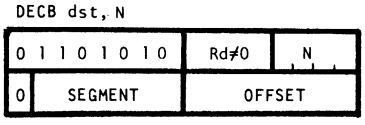
DECR dst, N



16

X SSO

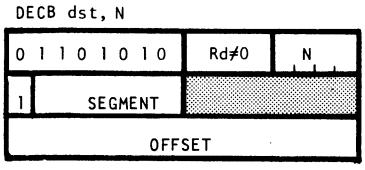
DECB dst, N



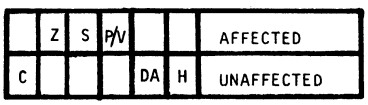
14

X SLO

DECB dst, N



17



Operation

dst<0:7> ← dst<0:7> - N - 1

Description

A value between 1 and 16 is subtracted from the destination byte operand and the result is loaded back into the destination. The desired value to be subtracted is specified by the N field. N = Ø corresponds to value 1 and so on, and N = F corresponds to value 16. The destination is determined by the applicable addressing mode.

Flags:

Z: Set to 1 if result is zero. Reset otherwise.
S: Set to 1 if result negative. Reset otherwise.
P/V: Set to 1 on arithmetic overflow.

DISABLE INTERRUPT

DI

This is a system instruction.

DI

S,NS | 0 1 1 1 1 1 0 0 | 0 0 0 0 0 0 V N | 6

Operation

FCW<11> ← Ø for N=Ø

FCW<11> ← FCW<11> for N=1

FCW<12> ← Ø for V=Ø

FCW<12> ← FCW<12>for V=1

Description

The interrupt enables in the FCW are reset to Ø dependent upon the values of the N & V bit within the instruction. A value of 1 in these bit positions causes the relevant interrupt enable to be unaltered, and a value of Ø causes the relevant interrupt enable to be Reset. The bit designated V in the instruction controls the vectored interrupt enable bit and the bit designated N controls the non-vectored enable interrupt bit.

						AFFECTED
C	Z	S	PV	DA	H	UNAFFECTED

Flags are not affected.

DIVIDE register pair by source word

DIV

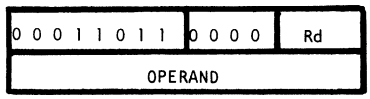
DIV dst, src

R S,NS

95

DIV dst, src

IM S,NS



95

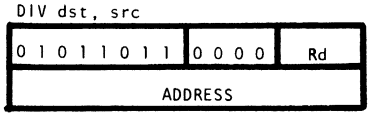
DIV dst, src

IR S,NS

95

DIV dst, src

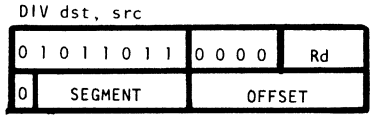
DA NS



96

DIV dst, src

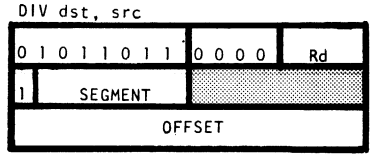
DA SSO



97

DIV dst, src

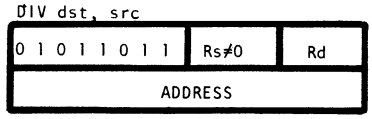
DA SLO



99

DIV dst, src

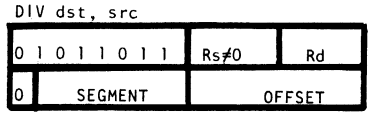
X NS



97

DIV dst, src

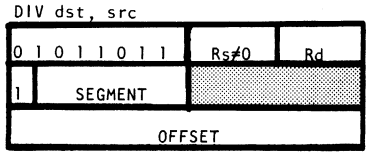
X SSO



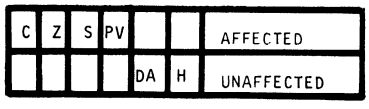
97

DIV dst, src

X SLO



100



Operation

dst<0:15> ← dst<0:31> /src<0:15>

dst<16:31> ← Remainder

Description

A 32-bit signed integer (dividend) is contained in a destination register pair designated by the Rd field of the instruction.

A 16-bit signed integer source operand (divisor) is determined by the applicable addressing mode. Division is performed to obtain a 16-bit quotient and a 16-bit remainder.

The quotient is loaded into the least significant destination register. The remainder is loaded into the most significant destination register. The source operand is not altered.

The original contents of the destination are lost unless the division operation is aborted. This occurs if the divisor is zero or if the magnitude of the divisor is **less than or equal to the magnitude of the high order half of the dividend.**

The aborted instruction takes less than 30 clock cycles.

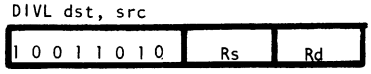
Flags:

C: Set to 1 if the quotient is less than -2^{15} or greater than/equal to 2^{+15}. Reset otherwise.
Z: Set to 1 if either the quotient or divisor is zero. Reset otherwise.
S: Set if quotient is negative. Reset otherwise.
P/V: Set to 1 if division is aborted. Reset otherwise.

DIVIDE register quadruple by source long word — DIVL

R — S,NS

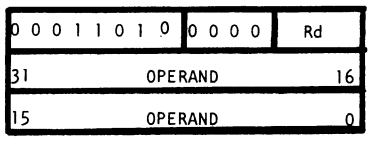
DIVL dst, src



723

IM — S,NS

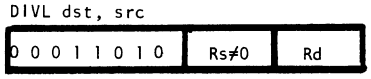
DIVL dst, src



723

IR — S,NS

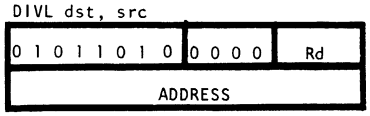
DIVL dst, src



723

DA — NS

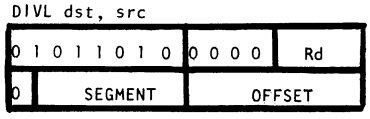
DIVL dst, src



724

DA — SSO

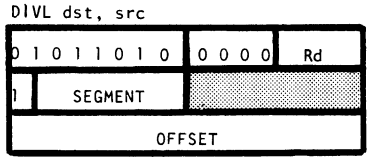
DIVL dst, src



725

DA — SLO

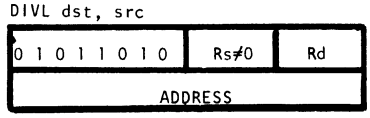
DIVL dst, src



727

X — NS

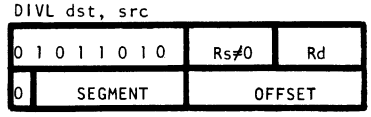
DIVL dst, src



725

X — SSO

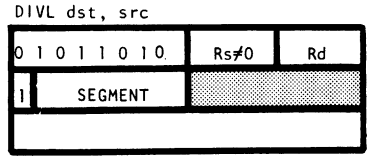
DIVL dst, src



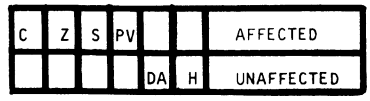
725

X — SLO

DIVL dst, src



728



Operation

dst<0:31> ← dst<0:63> /src<0:31>

dst<32:63> ← Remainder

Description

A 64-bit signed integer (dividend) is contained in a quadruple destination register designated by the Rd field of the instruction.

A 32-bit signed integer source operand (divisor) is determined by the applicable addressing mode. Division is performed to obtain a 32-bit quotient and a 32-bit remainder.

The quotient is loaded into the least significant destination register pair. The remainder is loaded into the most significant destination register pair. The source operand is not altered.

The original contents of the destination are lost unless the division operation is aborted. This occurs if the divisor is zero or if the magnitude of the divisor is less than or equal to the magnitude of the high order half of the dividend.

The aborted instruction takes a maximum of 60 clock cycles.

Flags:

C: Set to 1 if the quotient is less than -2^{31} or greater than/equal to 2^{+31}. Reset otherwise.
Z: Set to 1 if either the quotient or divisor is zero. Reset otherwise.
S: Set if quotient is negative. Reset otherwise.
P/V: Set to 1 if division is aborted. Reset otherwise.

DECREMENT word register & jump on non zero

DJNZ

DJNZ Rc, d

RA

II

Operation

Rc<0:15> ← Rc<0:15> - 1

If Rc<0:15> ≠0

Then PC ← Updated PC-2x displacement

Otherwise PC ← Updated PC

Description

The contents of the general purpose word register designated by the Rc field of the instruction are decremented and if this produces a non-zero result, a jump is executed. The jump address is obtained by subtracting the contents of the 7 bit instruction displacement field which has been left shifted (ie word aligned) from the contents of the updated program counter (ie incremented by 2). The resultant address is loaded into the program counter and is used as the jump destination. The displacement field is interpreted as a 7 bit unsigned integer. Thus the range of the relative jump is Ø to -127 words with respect to the updated PC.

If the register decrementation produces a zero result, then the contents of the program counter are merely updated by incrementing by 2.

						AFFECTED
C	Z	S	PV	DA	H	UNAFFECTED

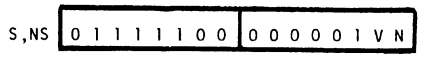
Flags are not affected.

ENABLE INTERRUPT

EI

This is a system instruction.

EI



6

Operation

FCW<11> ← 1 for N=Ø

FCW<11> ← FCW<11>for N=1

FCW<12> ← 1 for V=Ø

FCW<12> ← FCW<12>for V=1

Description

The interrupt enables in the FCW are set to 1 dependent upon the values of the N & V bits within the instruction. A value of 1 in these bit positions causes the relevant interrupt enable to be unaltered, and a value of Ø causes the relevant interrupt enable to be set. The bit designated V in the instruction controls the vectored interrupt enable bit and the bit designated N controls the non-vectored interrupt enable bit.

						AFFECTED
C	Z	S	PV	DA	H	UNAFFECTED

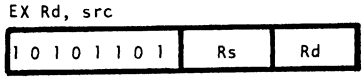
Flags are not affected.

EXCHANGE source word with destination word

EX

R S,NS

EX Rd, src



6

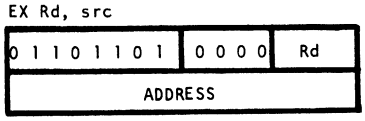
IR S,NS

EX Rd, src

12

DA NS

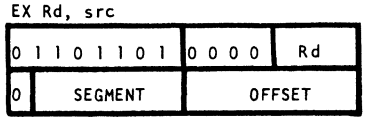
EX Rd, src



15

DA SSO

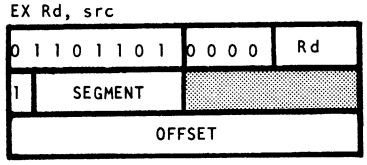
EX Rd, src



16

DA SLO

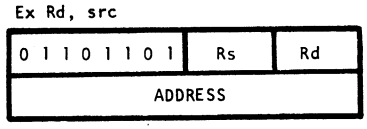
EX Rd, src



18

X NS

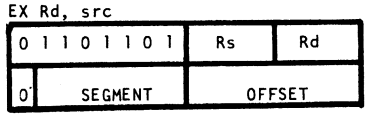
Ex Rd, src



16

X SSO

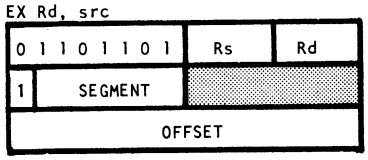
EX Rd, src



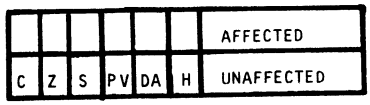
16

X SLO

EX Rd, src



19



Flags are not affected.

Operation

Src<0:15> ⟷ dst<0:15>

Description

The contents of the source operand word are exchanged with the contents of the destination operand word. The destination operand is always a general purpose word register designated by the Rd field of the instruction. The source operand is determined by the appropriate addressing mode.

EXCHANGE Source byte with destination byte — EXB

R	S,NS	EXB Rd, src 1 0 1 0 1 1 0 0 | Rs | Rd	6
IR	S,NS	EXB Rd, src 0 0 1 0 1 1 0 0 | Rs | Rd	12
DA	NS	EXB Rd, src 0 1 1 0 1 1 0 0 | 0 0 0 0 | Rd ADDRESS	15
DA	SSO	EXB Rd, src 0 1 1 0 1 1 0 0 | 0 0 0 0 | Rd 1 | SEGMENT | OFFSET	16
DA	SLO	EXB Rd, src 0 1 1 0 1 1 0 0 | 0 0 0 0 | Rd 1 | SEGMENT OFFSET	18
X	NS	EXB Rd, src 0 1 1 0 1 1 0 0 | Rs≠0 | Rd ADDRESS	16
X	SSO	EXB Rd, src 0 1 1 0 1 1 0 0 | Rs≠0 | Rd 0 | SEGMENT | OFFSET	16
X	SLO	EXB Rd, src 0 1 1 0 1 1 0 0 | Rs≠0 | Rd 1 | SEGMENT OFFSET	19

							AFFECTED
C	Z	S	PV	DA	H		UNAFFECTED

Flags are not affected.

Operation

src<0:7> ↔ dst<0:7>

Description

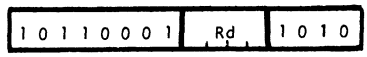
The contents of the source operand byte are exchanged with the contents of the destination operand byte. The destination operand is always a general purpose byte register designated by the Rd field of the instruction. The source operand is determined by the appropriate addressing mode.

EXTEND sign of a word

EXTS

EXTS Rd

R S,NS



11

Operation

If dst<0:15> is negative
dst<16:31> ← 1's
otherwise dst<16:31> ← Ø

Description

The destination is a general purpose register pair, designated by the Rd field of the instruction. The sign bit of the less significant register of the pair is copied into each bit position of the most significant register. In this manner, the sign of the operand is preserved as the operand is extended from 16 to 32 bits in length.

						AFFECTED
C	Z	S	PV	DA	H	UNAFFECTED

Flags are not affected.

EXTEND sign of a byte **EXTSB**

EXTSB Rd

R S,NS

11

Operation

If dst<0:7> is negative
dst<8:15> ← 1's
otherwise dst<8:15> ← Ø

Description

The destination is a general purpose register, designated by the Rd field of the instruction. The sign bit of the the less significant byte of the register is copied into each position of the most significant byte. In this manner, the sign of the operand is preserved as the operand is extended from 8 to 16 bits.

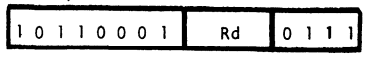
						AFFECTED
C	Z	S	PV	DA	H	UNAFFECTED

Flags are not affected.

EXTEND sign of a long word **EXTSL**

EXTSL, Rd

R S,NS



11

Operation

If dst<0:31> is negative
dst<32:63> ← 1's
Otherwise dst<32:63> ← Ø

Description

The destination is a general purpose register quad designated by the Rd field of the instruction. The sign bit of the less significant register pair of the quad is copied into each bit position of the most significant register pair. In this manner, the sign of the operand is preserved as the operand is extended from 32 to 64 bits.

						AFFECTED
C	Z	S	PV	DA	H	UNAFFECTED

Flags are not affected.

HALT

This is a system instruction.

HALT

0 1 1 1 0 1 0	0 0 0 0	0 0 0 0

8 + 3n*

*Interrupts are recognized at the end of each 3 cycle period.

Description

Instruction execution is suspended and CPU will be in a wait state until an interrupt or reset is received.

While in wait state, bus requests will be acknowledged and memory refresh will continue.

						AFFECTED
C	Z	S	PV	DA	H	UNAFFECTED

Flags are not affected.

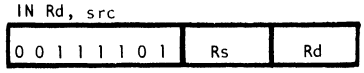
INPUT word to register from I/O port

IN

This is a system instruction.

IR S,NS

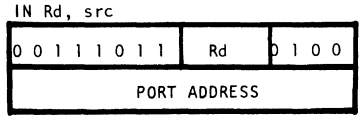
IN Rd, src



10

DA S,NS

IN Rd, src



12

Operation

Rd<0:15> ← src<0:15>

Description

A general purpose word destination register designated by the Rd field of the instruction is loaded from an input port. The port address is determined by the applicable addressing mode. The original contents of the destination are lost.

						AFFECTED
C	Z	S	PV	DA	H	UNAFFECTED

Flags are not affected.

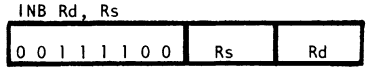
INPUT byte to register from I/O port

INB

This is a system instruction.

IR S,NS

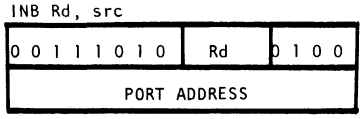
INB Rd, Rs



10

DA S,NS

INB Rd, src



12

Operation

Rd <0:7> ← src<0:7>

Description

A general purpose byte destination register designated by the Rd field of the instruction is loaded from an input port. The port address is determined by the applicable addressing mode. The original contents of the destination are lost.

						AFFECTED
C	Z	S	PV	DA	H	UNAFFECTED

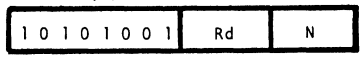
Flags are not affected.

INCREMENT word

INC

R

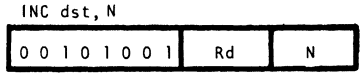
INC dst, N



4

IR

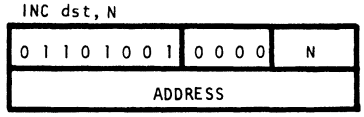
INC dst, N



11

DA

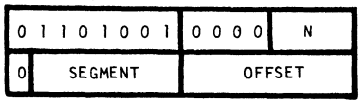
INC dst, N



13

DA

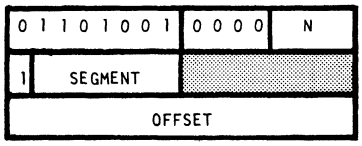
INC dst, N



14

DA

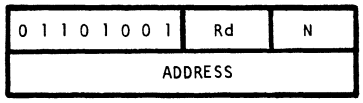
INC dst, N



16

X

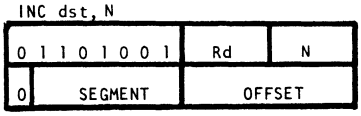
INC dst, N



14

X

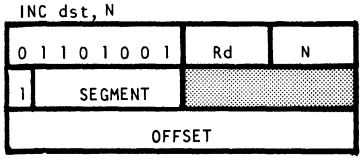
INC dst, N



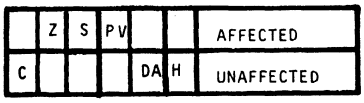
14

X

INC dst, N



17



Operation

dst<0:15> ← dst<0:15> + N + 1

Description

A value between 1 and 16 is added to the destination operand word and the result is loaded back into the destination. The desired value to be added is specified by the N field. N = Ø corresponds to value 1 and so on, and N = F corresponds to value 16. The destination is determined by the applicable addressing mode.

Flags:

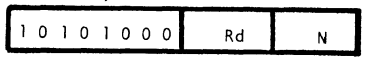
Z: Set to 1 if result is zero. Reset otherwise.
S: Set to 1 if result is negative. Reset otherwise.
P/V: Set to 1 on arithmetic overflow. Reset otherwise.

INCREMENT byte

INCB

R S,NS

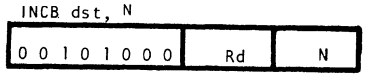
INCB dst, N



4

IR S,NS

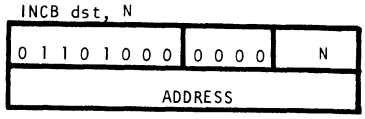
INCB dst, N



11

DA NS

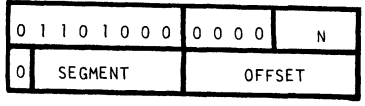
INCB dst, N



13

DA SSO

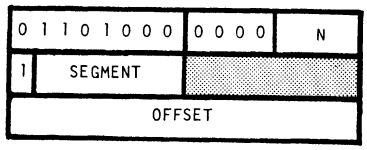
INCB dst, N



14

DA SLO

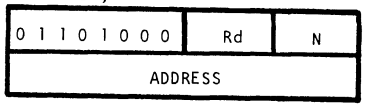
INCB dst, N



16

X NS

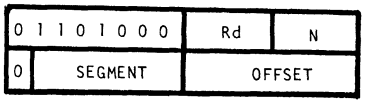
INCB dst, N



14

X SSO

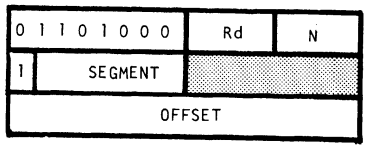
INCB dst, N



14

X SLO

INCB dst, N



17

	Z	S	PV	DA	H	
						AFFECTED
C						UNAFFECTED

Operation

dst<0:7> ← dst<0:7> + N + 1

Description

A value between 1 and 16 is added to the destination operand byte and the result is loaded back into the destination. The desired value to be added is specified by the N field. N = Ø corresponds to 1 and so on, and N = F corresponds to value 16. The destination is determined by the applicable addressing mode.

Flags:

Z: Set to 1 if result is zero. Reset otherwise.
S: Set to 1 if result is negative. Reset otherwise.
P/V: Set to 1 on arithmetic overflow. Reset otherwise.
DA: Set to 0.
H: Set on carry from least significant digit. Reset otherwise.

INPUT word from I/O port to memory, autodecrement **IND**

This is a system instruction.

IR S,NS

IND dst, src, Rc

0 0 1 1 1 0 1 1		Rs	1 0 0 0
0 0 0 0	Rc	Rd	1 0 0 0

21

Operation

dst<0:15> ← src<0:15>

Rd<0:15> ← Rd<0:15> - 2

Rc<0:15> ← Rc<0:15> - 1

Description

Data word from the port addressed by the contents of the general purpose register designated by the Rs field of the instruction is loaded into a memory destination. The destination is addressed by the contents of the general purpose register designated by the Rd field of the instruction. The original contents of the destination are lost. The contents of the general purpose registers designated by Rd are then decremented by 2. The contents of the general purpose register designated by Rc are decremented by 1.

			PV			AFFECTED
C	Z	S		DA	H	UNAFFECTED

Flags:

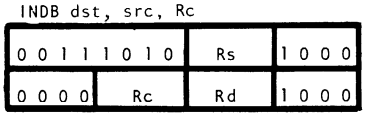
P/V: Set to 1 if result of decrementing Rc register is zero. Reset otherwise.

INPUT byte from I/O port to memory, autodecrement — INDB

This is a system instruction.

INDB dst, src, Rc

IR S,NS



21

Operation

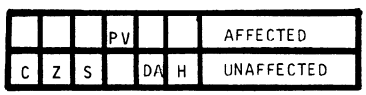
dst<0:7> ← src<0:7>

Rd<0:15> ← Rd<0:15> - 1

Rc<0:15> ← Rc<0:15> - 1

Description

Data byte from the port addressed by the contents of the general purpose register designated by the Rs field of the instruction is loaded into a memory destination. The destination is addressed by the contents of the general purpose register designated by the Rd field of the instruction. The original contents of the destination are lost. The contents of the general purpose registers designated by Rd and Rc are then decremented by 1.



Flags:

P/V: Set to 1 if result of decrementing Rc register is zero. Reset otherwise.

INPUT word from I/O port to memory, autodecrement and repeat

INDR

This is a system instruction.

INDR dst, src, Rc

IR S,NS

0	0	1	1	1	0	1	1	Rs				1	0	0	0
0	0	0	0	Rc				Rd				0	0	0	0

11 + 10n*

*n is the number of iterations

Operation

dst<0:15> ← src<0:15>

Rd<0:15> ← Rd<0:15> - 2

Rc<0:15> ← Rc<0:15> - 1

repeat until termination

Description

Data word from the port addressed by the contents of the general purpose register designated by the Rs field of the instruction is loaded into a memory destination. The destination is addressed by the contents of the general purpose register designated by the Rd field of the instruction. The original contents of the destination are lost. The contents of the general purpose register designated by Rd is then decremented by 2. The contents of the general purpose register designated by Rc are decremented by 1. The instruction is terminated when the result of this decrementation reaches zero. This instruction is interruptible.

			PV			AFFECTED
C	Z	S		DA	H	UNAFFECTED

Flags:

P/V: Set to 1.

INPUT byte from I/O port to memory, autodecrement and repeat

INDRB

This is a system instruction.

INDRB dst, src, Rc

IR S,NS

0	0	1	1	1	0	1	0	Rs				1	0	0	0
0	0	0	0	Rc				Rd				0	0	0	0

11 + 10n*

*n is the number of iterations

Operation

dst<0:7> ← src<0:7>

Rd<0:15> ← Rd<0:15> - 1

Rc<0:15> ← Rc<0:15> - 1

repeat until termination

Description

Data byte from the port addressed by the contents of the general purpose register designated by the Rs field of the instruction is loaded into the memory destination. The destination is addressed by the contents of the general purpose register designated by the Rd field of the instruction. The original contents of the destination are lost. The contents of the general purpose register designated by Rd are then decremented by 1. The contents of the general purpose register designated by Rc are decremented by 1. The instruction is terminated when the result of this decrementation reaches zero. This instruction is interruptible.

			PV			AFFECTED
C	Z	S		DA	H	UNAFFECTED

Flags:

P/V: Set to 1.

INPUT word from I/O port to memory, autoincrement

INI

This is system instruction.

INI dst, src, Rc

IR S,NS

0 0 1 1	1 0 1 1	Rs	0 0 0 0
0 0 0 0	Rc	Rd	1 0 0 0

21

Operation

dst<0:15> ← src<0:15>

Rd<0:15> ← Rd<0:15> + 2

Rc<0:15> ← Rc<0:15> - 1

Description

Data word from the port addressed by the contents of the general purpose register designated by the Rs field of the instruction is loaded into a memory destination. The destination is addressed by the contents of the general purpose register designated by the Rd field of the instruction. The original contents of the destination are lost. The contents of the general purpose register designated by Rd are then incremented by 2. The contents of the general purpose register designated by Rc are decremented by 1.

			PV			AFFECTED
C	Z	S		DA	H	UNAFFECTED

Flags:

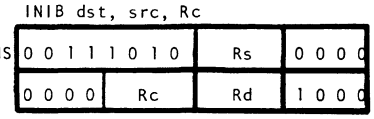
P/V: Set to 1 if result of decrementing Rc register is zero. Reset otherwise.

INPUT byte from I/O port to memory, autoincrement — INIB

This is a system instruction

INIB dst, src, Rc

IR S,NS



21

Operation

dst<0:7> ← src<0:7>

Rd<0:15> ← Rd<0:15> + 1

Rc<0:15> ← Rc<0:15> - 1

Description

Data byte from the port addressed by the contents of the general purpose register designated by the Rs field of the instruction is loaded into a memory destination. The destination is addressed by the contents of the general purpose register designated by the Rd field of the instruction. The original contents of the destination are lost. The contents of the general purpose registers designated by Rd are then incremented by 1. The contents of the general purpose register designated by Rc are decremented by 1.

Flags:

			PV			AFFECTED
C	Z	S		DA	H	UNAFFECTED

P/V: Set to 1 if result of decrementing Rc register is zero. Reset otherwise.

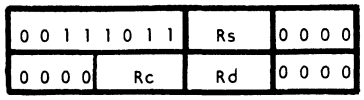
INPUT word from I/O port to memory, autoincrement and repeat

INIR

This is a system instruction

INIR dst, src, Rc

IR S,NS



11 + 10n*

*n is the number of iterations

Operation

dst<0:15> ← src<0:15>

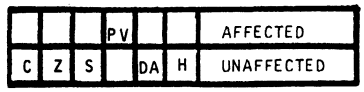
Rd<0:15> ← Rd<0:15> + 2

Rc<0:15> ← Rc<0:15> - 1

repeat until termination

Description

Data word from the port addressed by the contents of the general purpose register designated by the Rs field of the instruction is loaded into a memory destination. The destination is addressed by the contents of the general purpose register designated by the Rd field of the instruction. The original contents of the destination are lost. The contents of the general purpose register designated by Rd are then incremented by 2. The contents of the general purpose register designated by Rc are decremented by 1. This instruction is terminated when the result of this decrementation reaches zero. This instruction is interruptible.



Flags:

P/V: Set to 1.

INIRB dst, src, Rc

IR S,NS

0 0 1 1 1 0 1 0		Rs	0 0 0 0	11 + 10n*
0 0 0 0	Rc	Rd	0 0 0 0	

*n is the number of iterations.

Operation

dst<0:7> ← src<0:7>

Rd<0:15> ← Rd<0:15> + 1

Rc<0:15> ← Rc<0:15> - 1

repeat until termination

Description

Data byte from the port addressed by the contents of the general purpose register designated by the Rs field of the instruction is loaded into a memory destination. The destination is addressed by the contents of the general purpose register designated by the Rd field of the instruction. The original contents of the destination are lost. The contents of the general purpose register designated by Rd are then incremented by 1. The contents of the general purpose register designated by Rc are decremented by 1. This instruction is terminated when the result of this decrementation reaches zero. This instruction is interruptible.

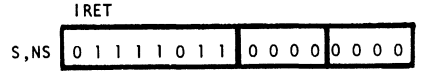
			PV			AFFECTED
C	Z	S		DA	H	UNAFFECTED

Flags:

P/V: Set to 1.

RETURN from interrupt **IRET** This is a system instruction.

IRET



16, 13

Operation

Non Segmented	Segmented
R15<0:15> ← R15<0:15>+ 2	R15<0:15> ← R15<0:15>+ 2
FCW ← (R15<0:15>)	FCW ← (RR14<0:22>)
-	R15<0:15> ← R15<0:15>+ 2
-	PC SEGMENT ← (RR14<0:22>)
R15<0:15> ← R15<0:15>+ 2	R15<0:15> ← R15<0:15>+ 2
PC ← (R15<0:15>)	PC OFFSET ← (RR14<0:22>)
R15<0:15> ← R15<0:15> +2	R15<0:15> ← R15<0:15 + 2>

Description

This instruction causes a return from an interrupt or trap. The program status that was pushed on the system stack is popped to restore the pre-interrupt processor status. The System Stack Pointer contents are modified to reflect the entries that are removed.

C	Z	S	PV	DA	H	
						AFFECTED
						UNAFFECTED

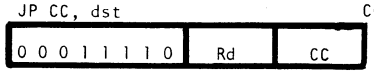
Flags:

The flags will be restored to pre-interrupt values.

JP

JP CC, dst — CC True/CC False

IR — S,NS

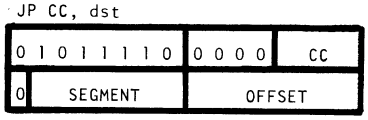


15,10/7,7

DA — NS

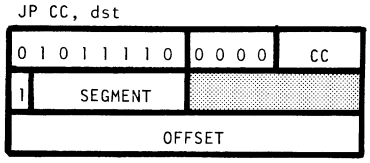
7/7

DA — SSO



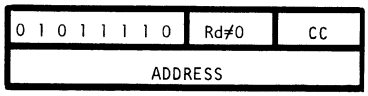
8/8

DA — SLO



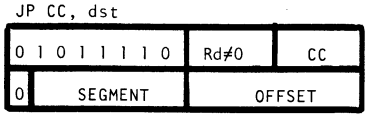
10/10

X — NS



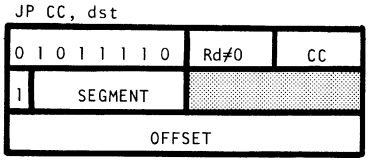
8/8

X — SSO



8/8

X — SLO



11/11

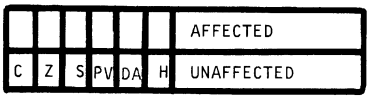
Operation

PC<0:15> ← dst<0:15>
if condition is met.

PC<0:15> ← Updated PC
if condition failed

Description

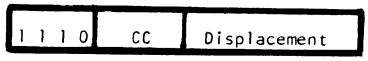
The program executes a jump if the condition set in the condition code field of the instruction is met. A destination address is obtained according to the applicable addressing mode, and is loaded into the program counter. If the condition set in the condition code of the instruction is not met, then the value in the program counter is merely updated.



Flags are not affected.

JR CC, d

RA



6

Operation

PC ← Updated PC + 2x displacement

If condition met

Otherwise PC ← Updated PC

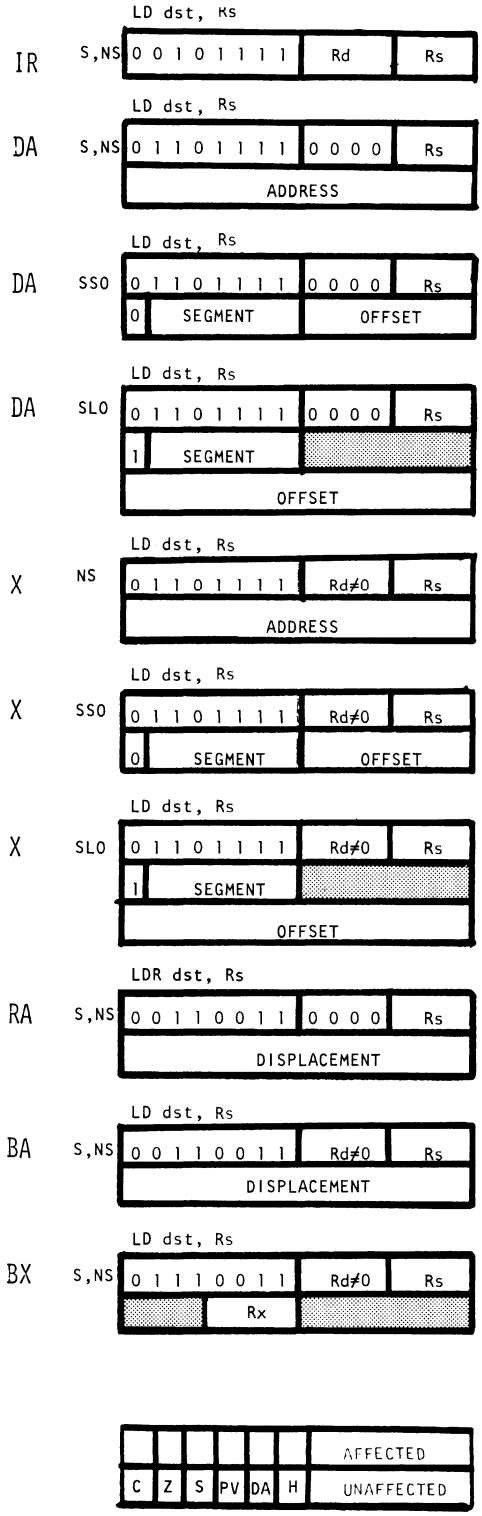
Description

A program jump is taken if the condition code set in the CC field of the instruction is met. If the condition is met, the contents of the program counter are updated (incremented by 2 and added to the contents of the 8 bit displacement field of the instruction, after the latter has been sign extended and left shifted (word aligned). The result is then loaded into the program counter as the jump address. If the condition is failed, the program counter is merely incremented by 2. The range of the jump is +127 to -128 words with respect to the updated PC. The program counter segment number remains unchanged.

						AFFECTED
C	Z	S	PV	DA	H	UNAFFECTED

Flags are not affected.

LOAD word register into memory — LD / LDR



Operation

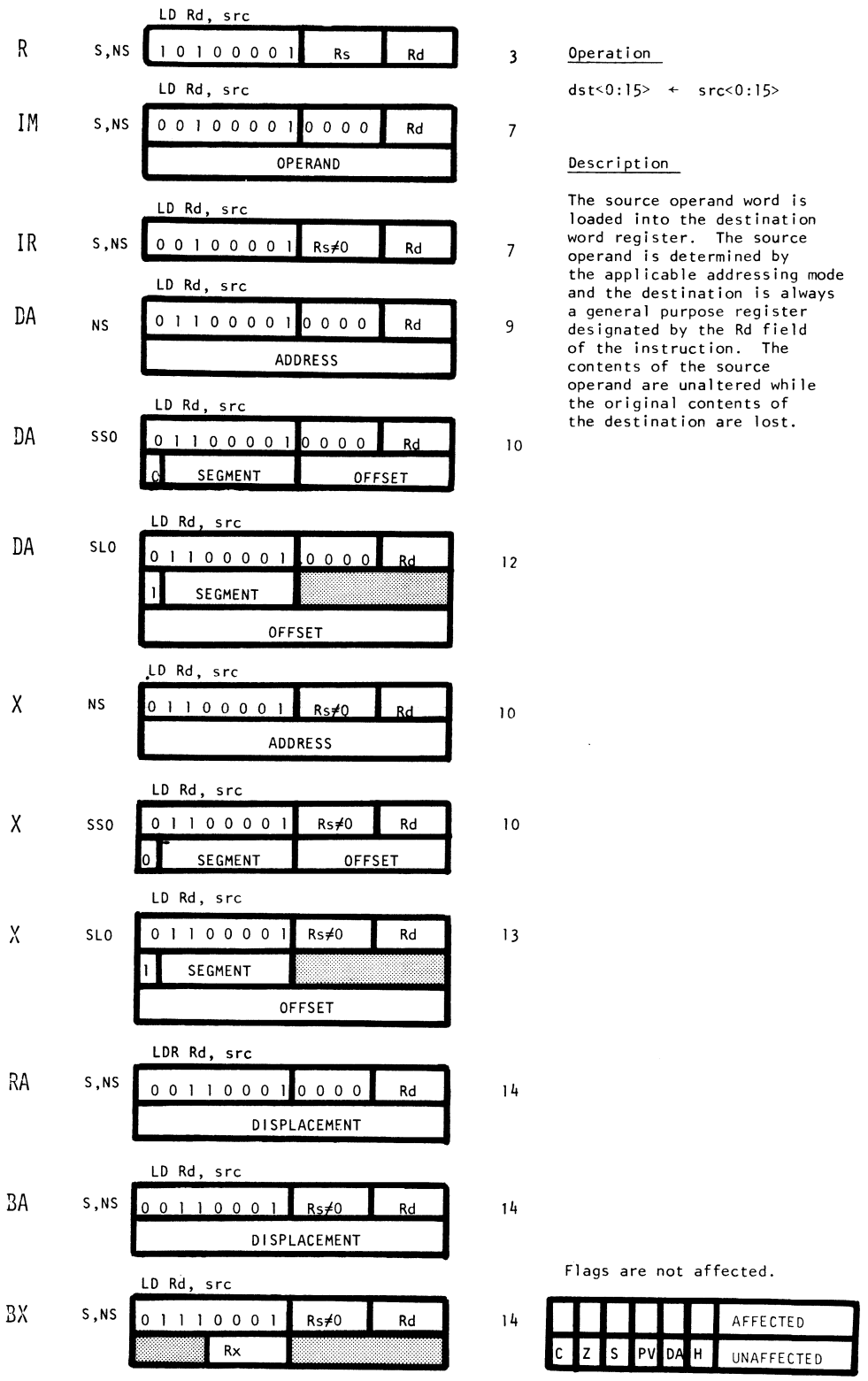
dst<0:15> ← src<0:15>

Description

The word contents of the source register are loaded into the word destination. The source operand is always a general purpose word register designated by the Rs field of the instruction. The destination is determined by the applicable addressing mode. The contents of the source are unaltered, and the original contents of the destination are lost.

Flags are not affected.

LOAD word into register — LD / LDR



Operation

dst<0:15> ← src<0:15>

Description

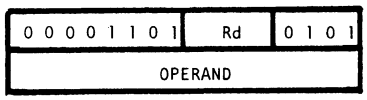
The source operand word is loaded into the destination word register. The source operand is determined by the applicable addressing mode and the destination is always a general purpose register designated by the Rd field of the instruction. The contents of the source operand are unaltered while the original contents of the destination are lost.

Flags are not affected.

LOAD IMMEDIATE word into memory — LD

LD dst, IM

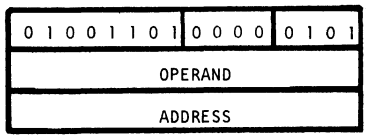
IR S,NS



11

LD dst, IM

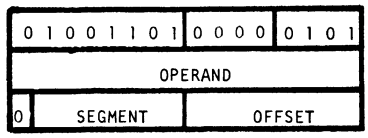
DA NS



14

LD dst, IM

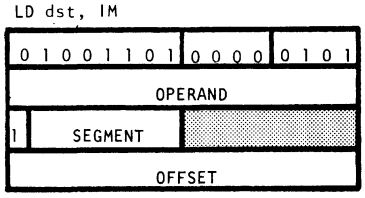
DA SSO



15

LD dst, IM

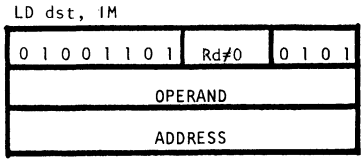
DA SLO



17

LD dst, IM

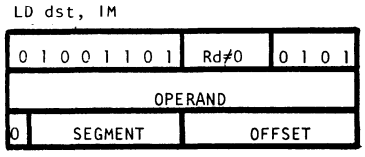
X NS



15

LD dst, IM

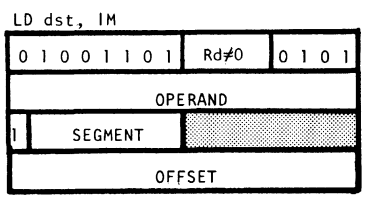
X SSO



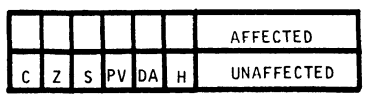
15

LD dst, IM

X SLO



18



Flags are not affected.

Operation

dst<0:15> ← IM <0:15>

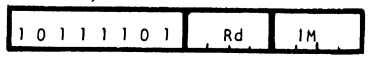
Description

The immediate word operand following the instruction in memory is loaded into the destination. The destination is determined by the applicable addressing mode. The original contents of the destination are lost.

LOAD , constant into a register LDB / LDK

LDK dst, IM

R S,NS



5

LDB dst, IM

R

5

Operation

dst<0:3> ← IM<0:3>
dst<4:15>←Ø
} 4 bits

dst<0:7> ← IM<0:7>
dst<8:15>←Ø
} 8 bits

Description

The immediate value in the instruction field, IM, is loaded into the least significant bits of the destination. The destination is a general purpose word register designated by the Rd field of the instruction. The remaining bits of the destination register are cleared.

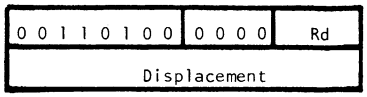
						AFFECTED
C	Z	S	PV	DA	H	UNAFFECTED

Flags are not affected

LOAD address to register — LDA / LDAR

RA S,NS

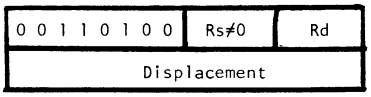
LDAR Rd, D



15

BA S,NS

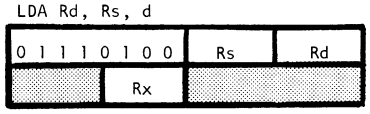
LDA Rd, Rs, d



15

BX S,NS

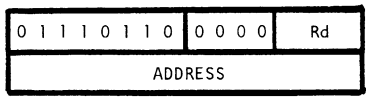
LDA Rd, Rs, d



15

DA NS

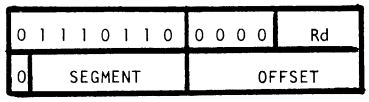
LDA Rd



12

DA SSO

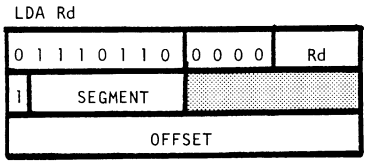
LDA Rd



13

DA SLO

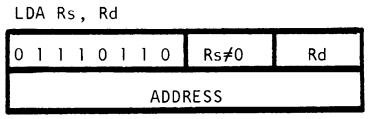
LDA Rd



15

X NS

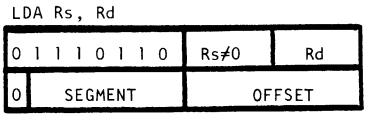
LDA Rs, Rd



13

X SSO

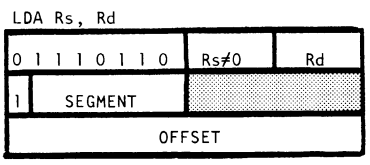
LDA Rs, Rd



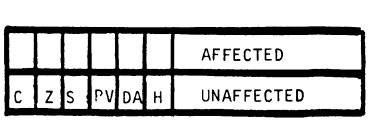
13

X SLO

LDA Rs, Rd



16



Flags are not affected.

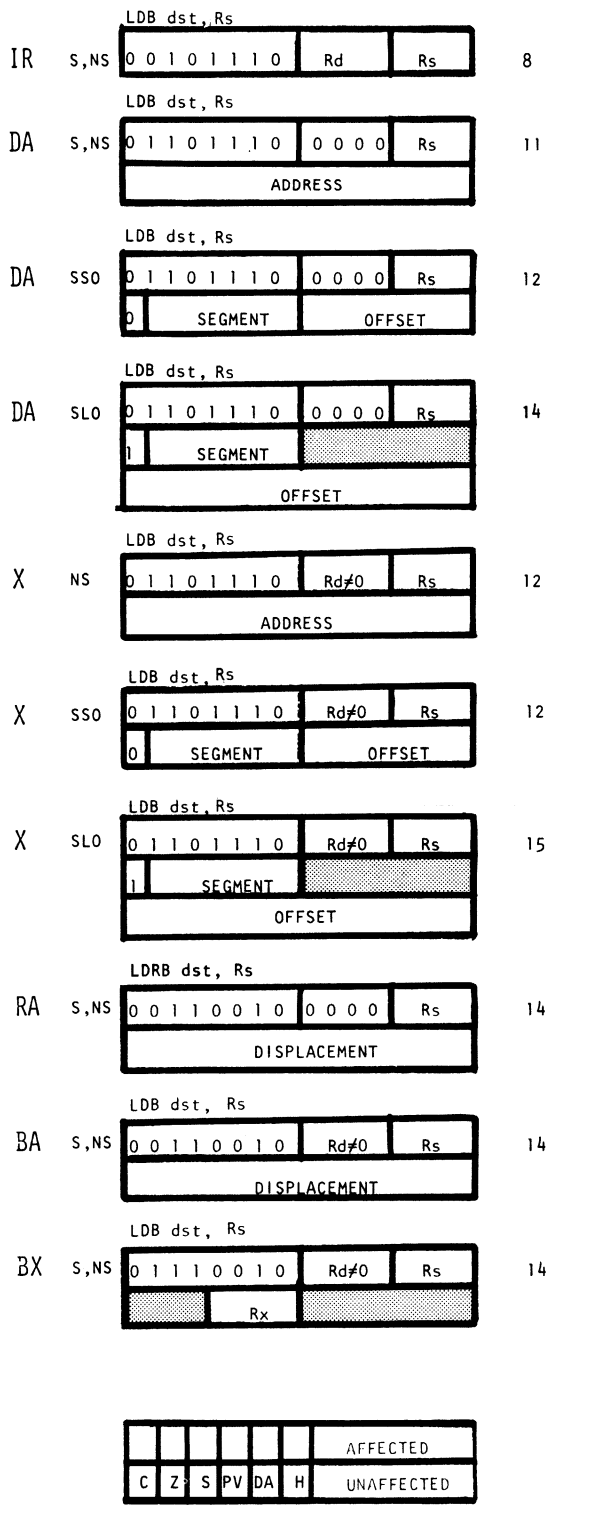
Operation

Described below.

Description

The address of a source operand is determined from the applicable addressing mode. This address is then loaded into the general purpose destination register designated by the Rd field of the instruction. The original contents of the destination are lost. Note that the destination will be a word operand in the non-segmented machine and a long word operand in the segmented machine.

LOAD byte register into memory — LDB / LDRB



Operation

dst<0:7> ← src<0:7>

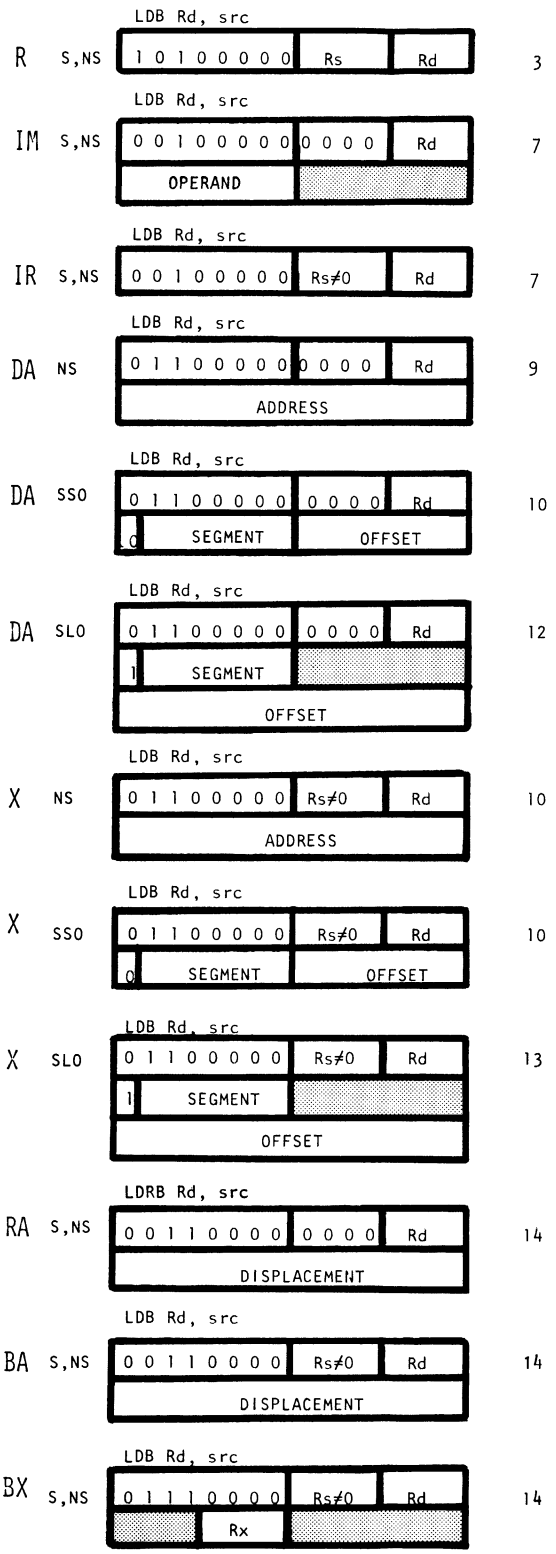
Description

The byte contents of the source register are loaded into the byte destination. The source operand is always a general purpose byte register designated by the Rs field of the instruction. The destination is determined by the applicable addressing mode. The contents of the source are unaltered. The original contents of the destination are lost.

Flags are not affected.

LOAD byte into register

LDB / LDRB



Operation

dst<0:7> ← src<0:7>

Destination

The source operand byte is loaded into the destination byte register. The source operand is determined by the applicable addressing mode and the destination is always a general purpose register designated by the Rd field of the instruction. The contents of the source operand are unaltered while the original contents of the destination are lost.

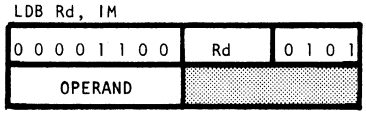
Flags are not affected.

							AFFECTED
C	Z	S	PV	DP	H		UNAFFECTED

LOAD IMMEDIATE byte into memory — LDB

IR S,NS

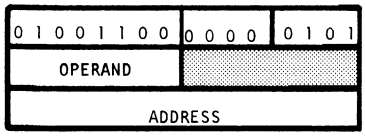
LDB Rd, IM



11

DA NS

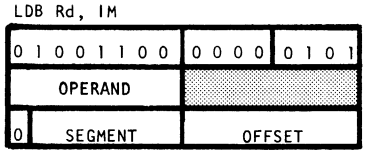
LDB Rd, IM



14

DA SSO

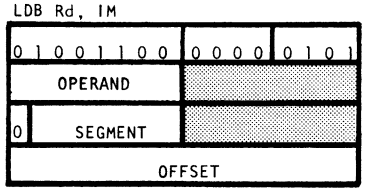
LDB Rd, IM



15

DA SLO

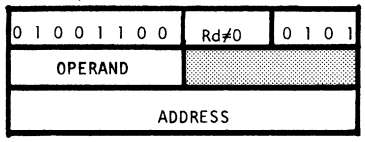
LDB Rd, IM



17

X NS

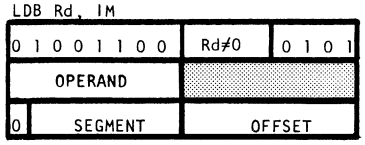
LDB Rd, IM



15

X SSO

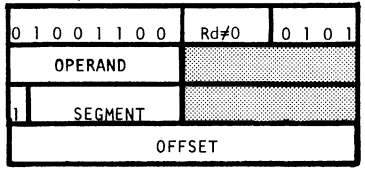
LDB Rd, IM



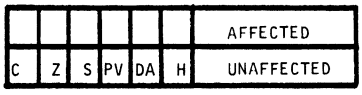
15

X SLO

LDB Rd, IM



18



Operation

dst<0:7> ← IM <0:7>

Description

The immediate byte operand following the instruction in memory is loaded into the destination. The destination is determined by the applicable addressing mode. The original contents of the destination are lost.

Flags are not affected.

LOAD control word into a register.

LDCTL

This is a system instruction.

LDCTL Rd, CW

R S,NS

7

Operation

Rd < 0:15 > ← CW < 0:15 >

Description

The contents of the control word specified in the CW field of the instruction are loaded into the general purpose destination word register specified by the Rd field of the instruction. The original contents of the destination are lost. Where a control word of less than 16 bits is loaded into the destination register, 0's are loaded into the unused bit positions.

The CW field decodes are shown below:

CW Field	Source
0 0 0	—
0 0 1	—
0 1 0	FCW
0 1 1	Refresh register (bits 1 through 8)
1 0 0	NPSAP segment
1 0 1	NPSAP upper offset
1 1 0	R14
1 1 1	R15

						AFFECTED
C	Z	S	PV	DA	H	UNAFFECTED

Flags are not affected.

LOAD control word from a register.

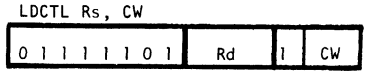
LDCTL

This is a system instruction.

R

LDCTL Rs, CW

S,NS



7

Operation

CW<0:15> ← Rs<0:15>

Description

The control word specified in the CW field of the instruction is loaded from the general purpose source word register specified by the Rs field of the instruction. The original contents of the control word are lost.

The CW field decodes are shown below:

CW Field	Destination
0 0 0	—
0 0 1	—
0 1 0	FCW
0 1 1	Refresh register (bits 1 through 15)
1 0 0	NPSAP segment
1 0 1	NPSAP upper offset
1 1 0	R14
1 1 1	R15

						AFFECTED
C	Z	S	PV	DA	H	UNAFFECTED

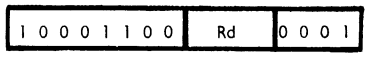
Flags are affected only if the FCW is selected as the destination.

LOAD flag byte into a register

LDCTLB

LDCTLB Rd

R



7

Operation

dst<0:7> ← FCW<0:7>

Description

The flag byte of the FCW is loaded into the general purpose byte destination register designated by the Rd field of the instruction. The previous contents of the destination register are lost.

						AFFECTED
C	Z	S	PV	DA	H	UNAFFECTED

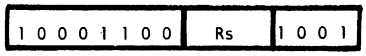
Flags are not affected.

LOAD flag byte from a register

LDCTLB

R

LDCTLB Rs



7

Operation

FCW<0:7> ← src<0:7>

Description

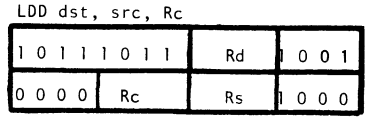
The flag byte of the FCW is loaded from a general purpose byte source register designated by the Rs field of the instruction. The previous contents of the flag register are lost.

C	Z	S	PV	DA	H	
						AFFECTED
						UNAFFECTED

Flags are affected as described above.

LDD dst, src, Rc

IR S,NS



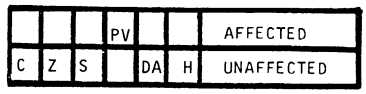
20

Operation

dst<0:15> ← src<0:15>
Rs<0:15> ← Rs<0:15> - 2
Rd<0:15> ← Rd<0:15> - 2
Rc<0:15> ← Rc<0:15> - 1

Description

The source word operand is loaded into the word destination. Both the source and destination operands are addressed by the general purpose registers designated in the Rs and Rd fields of the instruction. The contents of the source are unaltered and the original destination contents are lost. The contents of the general purpose register designated by the Rc field of the instruction are decremented by 1. The contents of Rs and Rd are both decremented by 2.



Flags:

P/V: Set to 1 if result of decrementing Rc is zero. Reset otherwise.

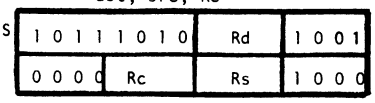
LOAD memory byte to memory, auto decrement

LDDB

CLOCK CYCLES

LDDB dst, src, Rc

IR S,NS



20

Operation

dst<0:7> ← src<0:7>

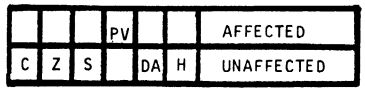
Rs<0:15> ← Rs<0:15>- 1

Rd<0:15> ← Rd<0:15>- 1

Rc<0:15> ← Rc<0:15>- 1

Description

The source byte operand is loaded into the byte destination. Both the source and destination operands are addressed by the general purpose registers designated in the Rs and Rd fields of the instruction. The contents of the source are unaltered, and the original destination contents are lost. The contents of the general purpose register designated by the Rc field of the instruction are decremented by 1. The contents of Rs and Rd are both decremented by 1.



Flags:

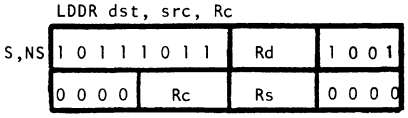
P/V: Set to 1 if result of decrementing Rc is zero. Reset otherwise.

LOAD memory word to memory autodecrement and repeat.

LDDR

LDDR dst, src, Rc

IR



11 + 9n*

*n is the number of iterations.

Operation

dst<0:15> ← src<0:15>
Rs<0:15> ← Rs<0:15>- 2
Rd<0:15> ← Rd<0:15>- 2
Rc<0:15> ← Rc<0:15>- 1
repeat until termination

Description

The source word operand is loaded into the word destination. Both the source and destination operands are addressed by the general purpose registers designated in the Rs and Rd fields of the instruction. The contents of the source are unaltered, and the original destination contents are lost. The contents of the general purpose register designated by the Rc field of the instruction are decremented by 1. The contents of Rs and Rd are both decremented by 2 and the operation will repeat until termination. Termination occurs when the contents of Rc are Ø. This instruction is interruptible.

			PV			AFFECTED
C	Z	S		DA	H	UNAFFECTED

Flags:

P/V: Set to 1.

LDDRB dst, src, Rc

IR S,NS

1 0 1 1	1 0 1 0	Rd	1 0 0 1
0 0 0 0	Rc	Rs	0 0 0 0

11 + 9n*

*n is the number of iterations

Operation

dst<0:7> ← src<0:7>
Rs<0:15> ← Rs<0:15> - 1
Rd<0:15> ← Rd<0:15> - 1
Rc<0:15> ← Rc<0:15> - 1
repeat until termination

Description

The source byte operand is loaded into the byte destination. Both the source and destination operands are addressed by the general purpose registers designated in the Rs and Rd fields of the instruction. The contents of the source are unaltered, and the original destination contents are lost. The contents of the general purpose register designated by the Rc field of the instruction are decremented by 1. The contents of Rs and Rd are both decremented by 1 and the operation will repeat until termination. Termination occurs when the contents of Rc are Ø. This instruction is interruptible.

			PV			AFFECTED
C	Z	S		DA	H	UNAFFECTED

Flags:

P/V: Set to 1.

LOAD memory word to memory, autoincrement. **LDI**

LDI dst, src, Rc

IR S,NS

1 0 1 1 1 0 1 1	Rd	0 0 0 1
0 0 0 0	Rc	Rs	1 0 0 0

20

Operation

dst<0:15> ← src<0:15>
Rs<0:15> ← Rs<0:15> + 2
Rd<0:15> ← Rd<0:15> + 2
Rc<0:15> ← Rc<0:15> - 1

Description

The source word operand is loaded into the word destination. Both the source and destination operands are addressed by the general purpose registers designated in the Rs and Rd fields of the instruction . The contents of the source are unaltered and the original destination contents are lost. The contents of the general purpose register designated by the Rc field of the instruction are decremented by 1. The contents of Rs and Rd are both incremented by 2.

			PV			AFFECTED
C	Z	S		DA	H	UNAFFECTED

Flags:

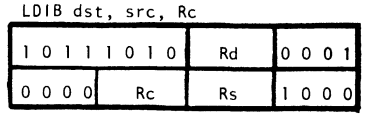
P/V: Set to 1 if result of decrementing Rc is zero. Reset otherwise.

LOAD memory byte to memory, autoincrement.

LDIB

IR

LDIB dst, src, Rc



20

Operation

dst<0:7> ← src<0:7>

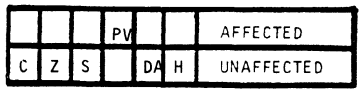
Rs<0:15> ← Rs<0:15>+ 1

Rd<0:15> ← Rd<0:15>+ 1

Rc<0:15> ← Rc<0:15>- 1

Description

The source byte operand is loaded into the byte destination. Both the source and destination operands are addressed by the general purpose registers designated in the Rs and Rd fields of the instruction. The contents of the source are unaltered, and the original destination contents are lost. The contents of the general purpose register designated by the Rc field of the instruction are decremented by 1. The contents of Rs and Rd are both incremented by 1.



Flags:

P/V: Set to 1 if result of decrementing Rc is zero. Reset otherwise.

LOAD memory word to memory, auto-increment and repeat.

LDIR

LDIR dst, src, Rc,

IR S,NS

1	0	1	1	1	0	1	1	Rd				0	0	0	1
0	0	0	0	Rc				Rs				0	0	0	0

11 + 9n*

*n is the number of iterations.

Operation

dst<0:15> ← src<0:15>
Rs<0:15> ← Rs<0:15>+ 2
Rd<0:15> ← Rd<0:15>+ 2
Rc<0:15> ← Rc<0:15>- 1
repeat until termination

Description

The source word operand is loaded into the word destination. Both the source and destination operands are addressed by the general purpose registers designated in the Rs and Rd fields of the instruction. The contents of the source are unaltered, and the original destination contents are lost. The contents of the general purpose register designated by the Rc field of the instruction are decremented by 1. The contents of Rs and Rd are both incremented by 2 and the operation will repeat until termination. Termination occurs when the contents of Rc are Ø. This instruction is interruptable.

			PV			AFFECTED
C	Z	S		DA	H	UNAFFECTED

Flags:

P/V: Set to 1.

LOAD memory byte to memory, autoincrement and repeat. **LDIRB**

LDIRB dst, src, Rc

IR S,NS

1 0 1 1 1 0 1 0	Rd	0 0 0 1
0 0 0 0	Rc	Rs	0 0 0 0

*n is the number of iterations.

11 + 9n* Operation

dst<0:7> ← src<0:7>
Rs<0:15> ← Rs<0:15>+ 1
Rd<0:15> ← Rd<0:15>+ 1
Rc<0:15> ← Rc<0:15>- 1
repeat until termination

Description

The source byte operand is loaded into the byte destination. Both the source and destination operands are addressed by the general purpose registers designated in the Rs and Rd fields of the instruction. The contents of the source are unaltered and the original destination contents are lost. The contents of the general purpose register designated by the Rc field of the instruction are decremented by 1. The contents of Rs and Rd are both incremented by 1 and the operation will repeat until termination. Termination occurs when the contents of Rc are Ø. This instruction is interruptible.

			PV			AFFECTED
C	Z	S		DA	H	UNAFFECTED

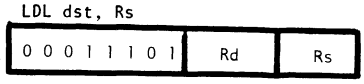
Flags:

P/V: Set to 1.

LOAD long word register to memory — LDL / LDRL

LDL dst, Rs

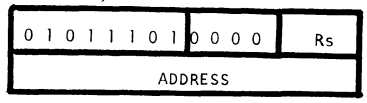
IR S,NS



11

LDL dst, Rs

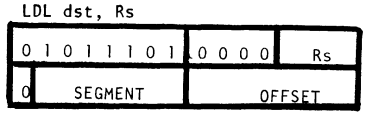
DA NS



14

LDL dst, Rs

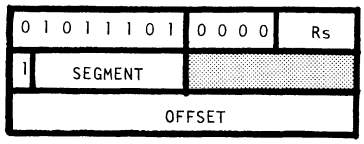
DA SSO



15

LDL dst, Rs

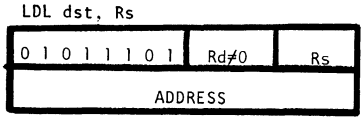
DA SLO



17

LDL dst, Rs

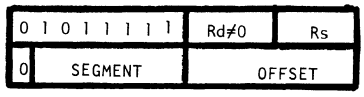
X NS



15

LDL dst, Rs

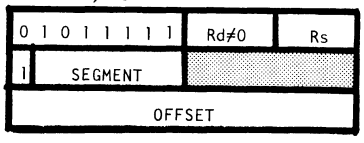
X SSO



15

LDL dst, Rs

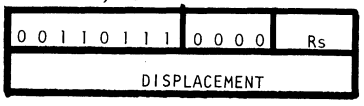
X SLO



18

LDRL dst, Rs

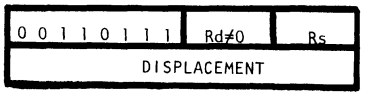
RA S,NS



17

LDL dst, Rs

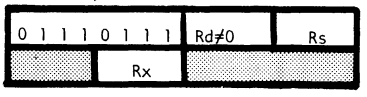
BA S,NS



17

LDL dst, Rs

BX S,NS



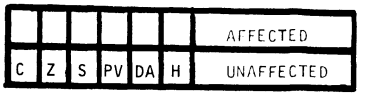
17

Operation

dst<0:31> ← src<0:31>

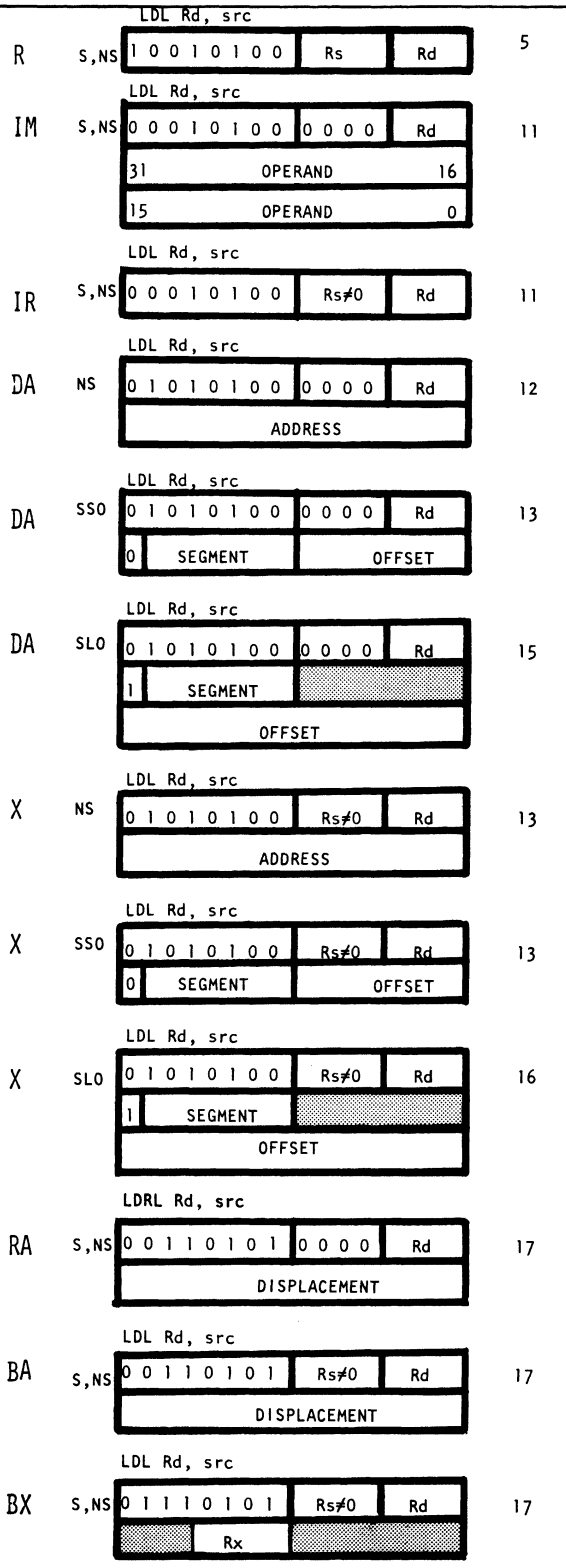
Description

The long word contents of the source register are loaded into the destination. The source operand is always a general purpose long word register designated by the Rs field of the instruction. The long word destination is determined from the applicable addressing mode. The contents of the source are unaffected, and the original contents of the destination are lost.



Flags are not affected.

LOAD long word into register. LDL / LDRL



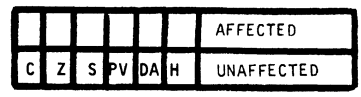
Operation

dst<0:31> ← src<0:31>

Description

The source operand long word is loaded into the destination long word register. The source operand is determined by the applicable addressing mode and the destination is always a general purpose register designated by the Rd field of the instruction. The contents of the source operand are unaltered while the original contents of the destination are lost.

Flags are not affected.

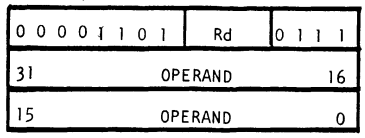


LOAD IMMEDIATE long word to memory

LDL

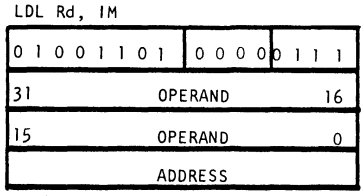
IR S,NS — 17

LDL Rd, IM



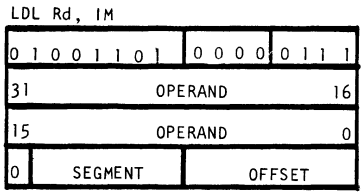
DA NS — 20

LDL Rd, IM



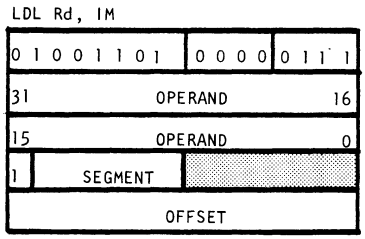
DA SSO — 21

LDL Rd, IM



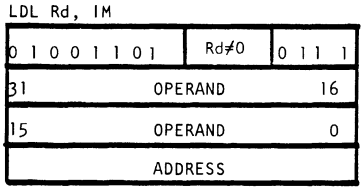
DA SLO — 23

LDL Rd, IM



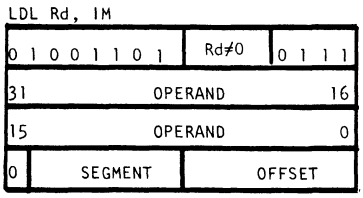
X NS — 21

LDL Rd, IM



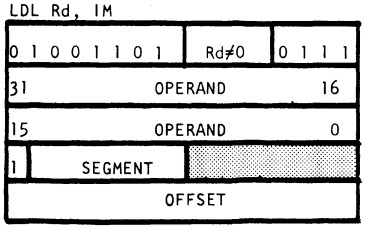
X SSO — 21

LDL Rd, IM



X SLO — 24

LDL Rd, IM



Operation

dst <0:31> ← IM <0:31>

Description

The immediate long word data following the instruction in memory is loaded into the destination long word operand. The destination operand is determined by the applicable addressing mode. The original contents of the destination are lost.

Flags are not affected.

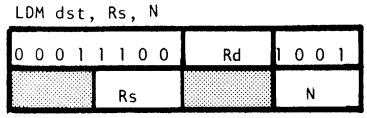
						AFFECTED
C	Z	S	PV	DA	H	UNAFFECTED

LOAD multiple registers into memory

LDM

IR S,NS

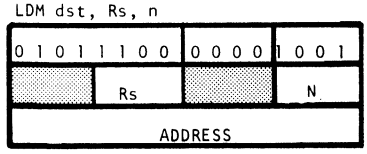
LDM dst, Rs, N



11 + 3n

DA NS

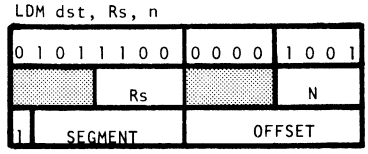
LDM dst, Rs, n



14 + 3n

DA SSO

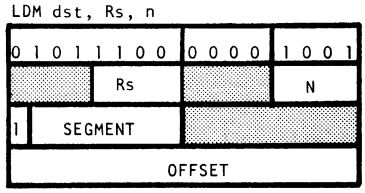
LDM dst, Rs, n



15 + 3n

DA SLO

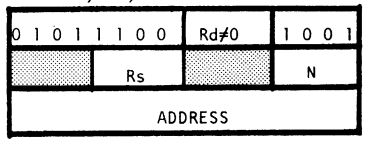
LDM dst, Rs, n



17 + 3n

X NS

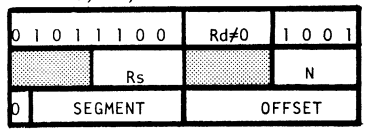
LDM dst, Rs, n



15 + 3n

X SSO

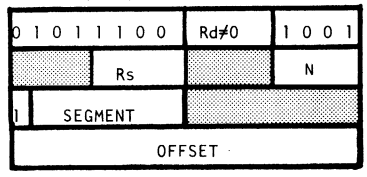
LDM dst, Rs, n



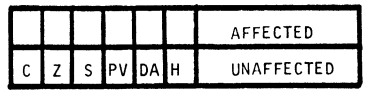
15 + 3n

X SLO

LDM dst, Rs, n



18 + 3n



Flags are not affected.

Operation

Description

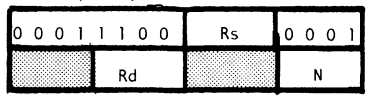
A specified number of 16-bit general purpose registers are loaded into memory. Loading will take place into consecutive memory locations with ascending addresses. The first register to be saved is specified in the Rs field of the instruction and registers will be accessed in ascending order, with RØ following R15. The number of registers to be saved is specified in the N field of the instruction. A Ø in this field represents 1 register, etc. The destination address is determined by the applicable addressing mode using the Rd field of the instruction. The first register will be saved at this address. Succeeding registers will be saved at successive memory locations. The contents of the general purpose registers are not altered. This instruction is not interruptible.

LOAD multiple registers from memory

LDM

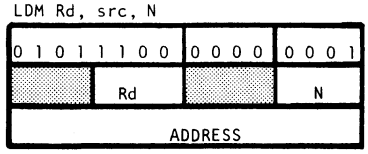
LDM Rd, src, N

IR S,NS



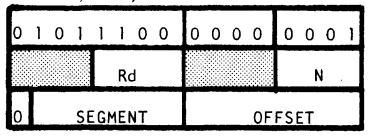
11 + 3n

DA NS



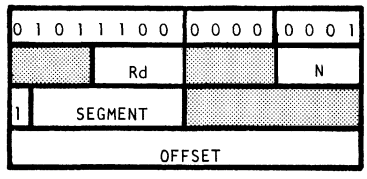
14 + 3n

DA SSO



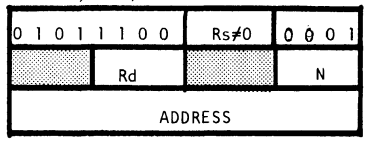
15 + 3n

DA SLO



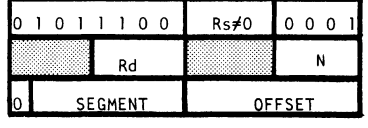
17 + 3n

X NS



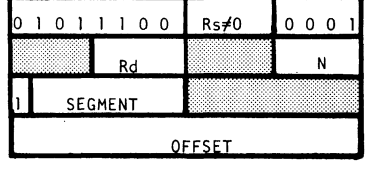
15 + 3n

X SSO

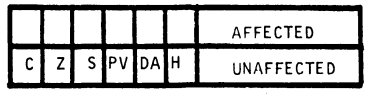


15 + 3n

X SLO



18 + 3n



Flags are not affected.

Operation

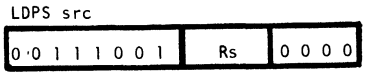
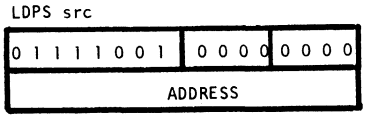
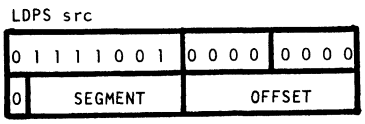
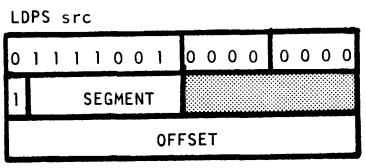
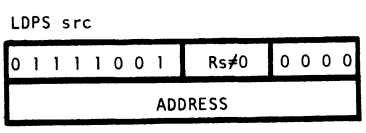
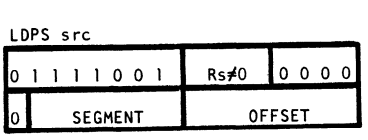
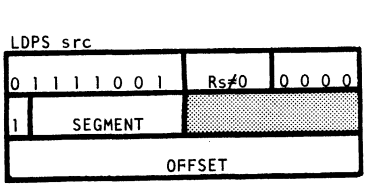
Description

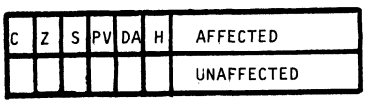
A specified number of general purpose registers are loaded with words from consecutive memory locations with ascending addresses. The first register to be loaded is specified in the Rd field of the instruction. The registers will be addressed in ascending order for loading, with RØ following R15. The number of registers to be loaded is specified in the 'N' field of the instruction. A Ø in this field represents 1 register, etc. A source operand address is generated according to the applicable addressing mode specified by the Rs field of the instruction. The first register will be loaded from this location. Succeeding registers will be loaded from successive memory locations. The memory contents are not altered. This instruction is not interruptible.

LOAD program status

LDPS

This is a system instruction.

IR	S,NS		12,16
DA	NS		16
DA	SSO		20
DA	SLO		22
X	NS		17
X	SSO		20
X	SLO		23



Operation

Description

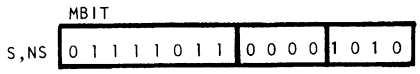
This instruction loads the processor status from consecutive memory locations with ascending address. The starting address of the status is determined by the applicable addressing mode. In AmZ8001 the status is four consecutive words, and in AmZ8002 the status is two words.

The PC segment number is not affected by this instruction in non-segmented mode.

The processor flags are loaded with the contents of the new FCW.

MBIT

This is a system instruction.



7

Operation

S FLAG ← $\overline{\mu 1}$

Z FLAG ← Ø

Description

The multi-micro input line $\mu 1$ is tested.

Flags:

S: Set to 1 if $\mu 1$ is active. Reset otherwise.
Z: Undefined.

		S				AFFECTED
C	Z		PV	DA	H	UNAFFECTED

MULTI MICRO REQUEST **MREQ**

This is a system instruction.

MREQ Rc

S,NS | 0 1 1 1 1 0 1 1 | Rc | 1 1 0 1 | 12 + 7n*

*n is the number of decrementations.

(n = Ø if initial state of μI was 1)

Description

There is an external input called Micro-in (μI) and an output called Micro-out (μO). The MREQ instruction tests the state of the μI input. If the μI input is 1, the instruction terminates If the μI input is zero, the μO output is activated and the general purpose register designated by the Rc field of the instruction is decremented by 1. The state of the μI line is tested, and the contents of Rc are repeatedly decremented until they reach zero. The instruction then terminates with the μO line active if μI is set, or with the μO line inactive if μI is not set.

Flags:

S	Z	
0	0	Instruction terminated after initial test of μI
0	1	Instruction terminated due to contents of Rc reaching zero.
1	1	Instruction terminated due to μI input being 1 after μO input was activated.

	Z	S				AFFECTED
C			PV	DA	H	UNAFFECTED

MULTI MICRO RESET — MRES

This is a system instruction

MRES

S,NS	0	1	1	1	0	1	1	0	0	0	1	0	0	1	7

Operation

$\mu 0 \leftarrow \emptyset$

Description

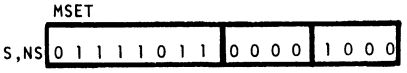
The multi-micro out line $\mu 0$ is reset.

						AFFECTED
C	Z	S	PV	DA	H	UNAFFECTED

Flags are not affected.

MULTI MICRO SET **MSET** This is a system instruction.

MSET



7

Operation

$\mu 0 \leftarrow 1$

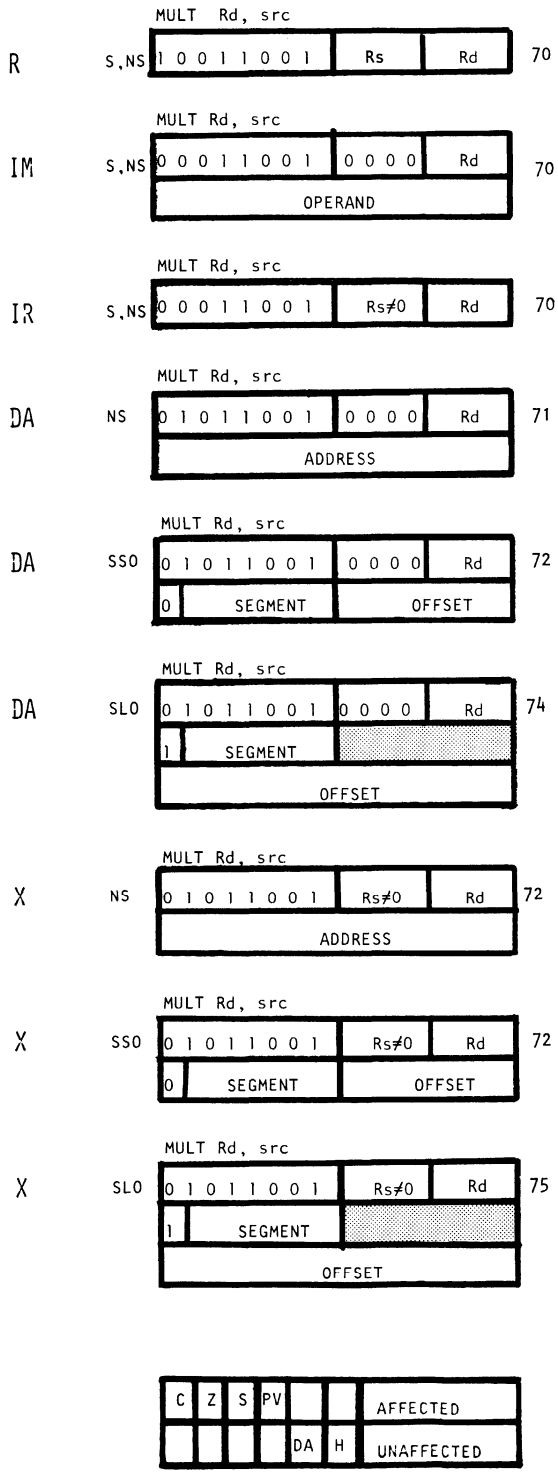
Description

The multi-micro out line $\mu 0$ is set. Note that this operation performs an unconditional setting of the $\mu 0$ line, independent of the state of the multi-micro in line $\mu 1$.

						AFFECTED
C	Z	S	PV	DA	H	UNAFFECTED

Flags are not affected.

MULTIPLY register with word — MULT



Operation

dst<0:31> ← dst<0:15>X src<0:15>

Description

The least significant word of a destination register pair (multiplicand) is multiplied by the contents of a source word operand (multiplier). The result is loaded into the destination, which is a general purpose register pair, designated by the Rd field of the instruction. The source operand is determined from the applicable addressing mode. Both the multiplicand and multiplier are treated as signed two's complement 16 bit integers. The original contents of the destination are lost. The source contents are unaltered.

Flags:

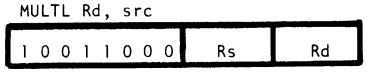
C: Set to 1 if product is less than -2^{15} or greater than/equal to 2^{15}. Reset otherwise.
Z: Set to 1 if product is zero. Reset otherwise.
S: Set to 1 if product is negative. Reset otherwise.
P/V: Reset

MULTL

MULTIPLY register with long word

MULTL Rd, src

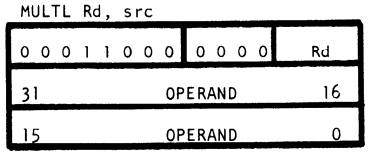
R S,NS



282 + 7n*

MULTL Rd, src

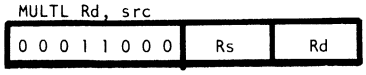
IM S,NS



282 + 7n*

MULTL Rd, src

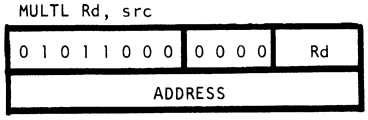
IR S,NS



282 + 7n*

MULTL Rd, src

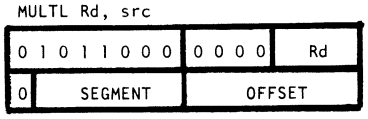
DA NS



283 + 7n*

MULTL Rd, src

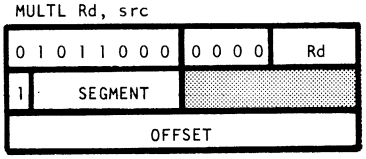
DA SSO



286 + 7n*

MULTL Rd, src

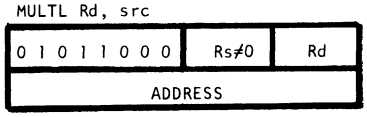
DA SLO



286 + 7n*

MULTL Rd, src

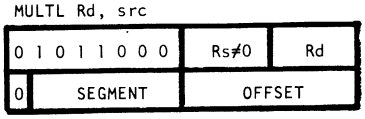
X NS



284 + 7n*

MULTL Rd, src

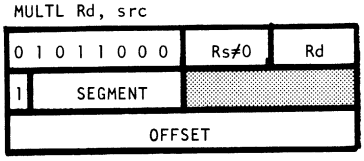
X SSO



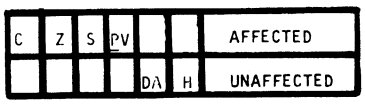
286 + 7n*

MULTL Rd, src

X SLO



287 + 7n*



Operation

dst<0:63> ← dst<0:31>X src<0:31>

Description

The least significant long word of a destination register quadruple (multiplicand) is multiplied by the contents of a source long word operand (multiplier). The result is loaded into the destination, which is a general purpose register quadruple designated by the Rd field of the instruction. The source operand is determined from the applicable addressing mode. Both the multiplicand and multiplier are treated as signed two's complement 32 bit integers. The original contents of the destination are lost. The source contents are unaltered.

*n is the number of bits equal to 1 in the absolute value of the least significant half of the destination operand.

Flags:

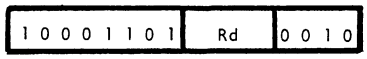
C: Set to 1 if product is less than -2^{31}, or greater than/equal to 2^{31}. Reset otherwise.
Z: Set to 1 if product is zero. Reset otherwise.
S: Set to 1 if product is negative. Reset otherwise.
P/V: Reset.

NEGATE word

NEG

R S,NS — 7

NEG dst

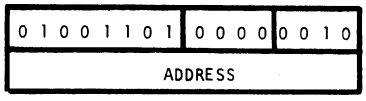


IR S,NS — 12

NEG dst

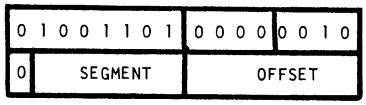
DA NS — 15

NEG dst



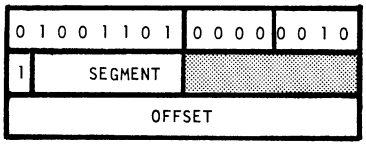
DA SSO — 16

NEG dst



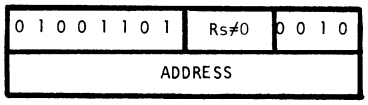
DA SLO — 18

NEG dst



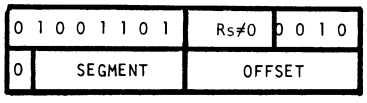
X NS — 16

NEG dst



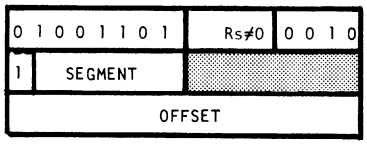
X SSO — 16

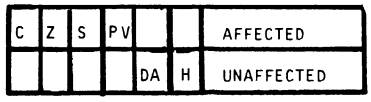
NEG dst



X SLO — 19

NEG dst





Operation

dst<0:15> ← $\overline{\text{dst}}$<0:15> + 1

Description

The contents of the destination word operand are replaced by its two's complement. The destination operand is obtained by using the applicable addressing mode. The negation is achieved by complementing the destination operand and adding 1.

Flags:

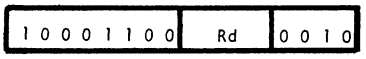
C: Reset on carry from destination. Set to 1 otherwise.
Z: Set to 1 if the result is zero. Reset otherwise.
S: Set to 1 if result is negative. Reset otherwise.
P/V: Set to 1 if operand value is 8000 (HEX) Reset otherwise.

NEGATE byte

NEGB

R S,NS

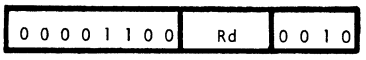
NEGB dst



7

IR S,NS

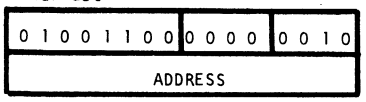
NEGB dst



12

DA NS

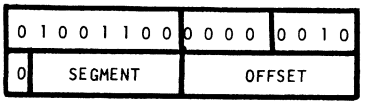
NEGB dst



15

DA SSO

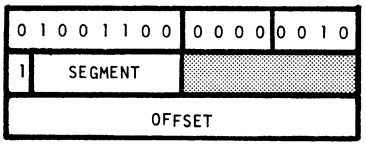
NEGB dst



16

DA SLO

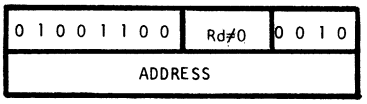
NEGB dst



18

X NS

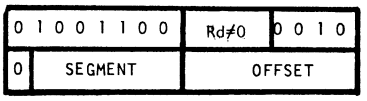
NEGB dst



16

X SSO

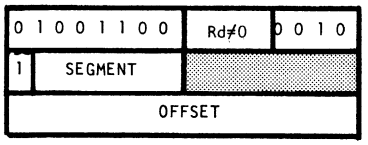
NEGB dst



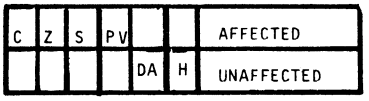
16

X SLO

NEGB dst



19



Operation

dst<0:7> ← $\overline{\text{dst}}$<0:7> + 1

Description

The contents of the destination byte operand are replaced by its two's complement. The destination operand is obtained by using the applicable addressing mode. The negation is achieved by complementing the destination operand and adding 1.

Flags:

C: Reset on carry from destination. Set to 1 otherwise.
Z: Set to 1 if the result is zero. Reset otherwise.
S: Set to 1 if result is negative. Reset otherwise.
P/V: Set to 1 if the operand value is 80 (HEX). Reset otherwise.

NO OPERATION

NOP

NOP

S,NS	1	0	0	0	1	1	0	1	0	0	0	0	0	1	1	1

7

No operation takes place, and PC is incremented by 2.

						AFFECTED
C	Z	S	PV	DA	H	UNAFFECTED

Flags are not affected.

OR word with register

OR

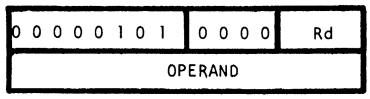
OR Rd, Rs

R S,NS

4

OR Rd, IM

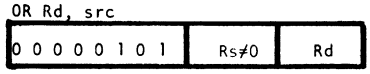
IM S,NS



7

OR Rd, src

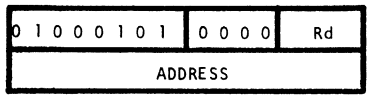
IR S,NS



7

OR Rd, src

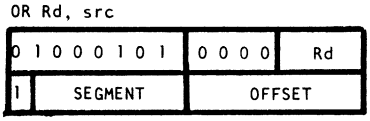
DA NS



9

OR Rd, src

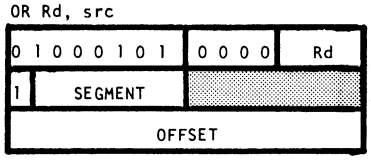
DA SSO



10

OR Rd, src

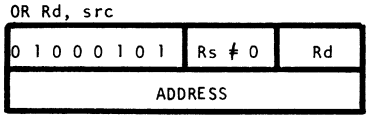
DA SLO



12

OR Rd, src

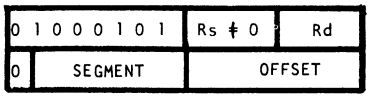
X NS



10

OR Rd, src

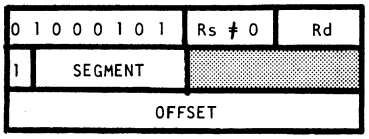
X SSO



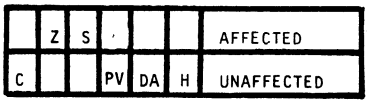
10

OR Rd, src

X SLO



13



Operation

dst<0:15> ←src<0;15>V dst<0:15>

Description

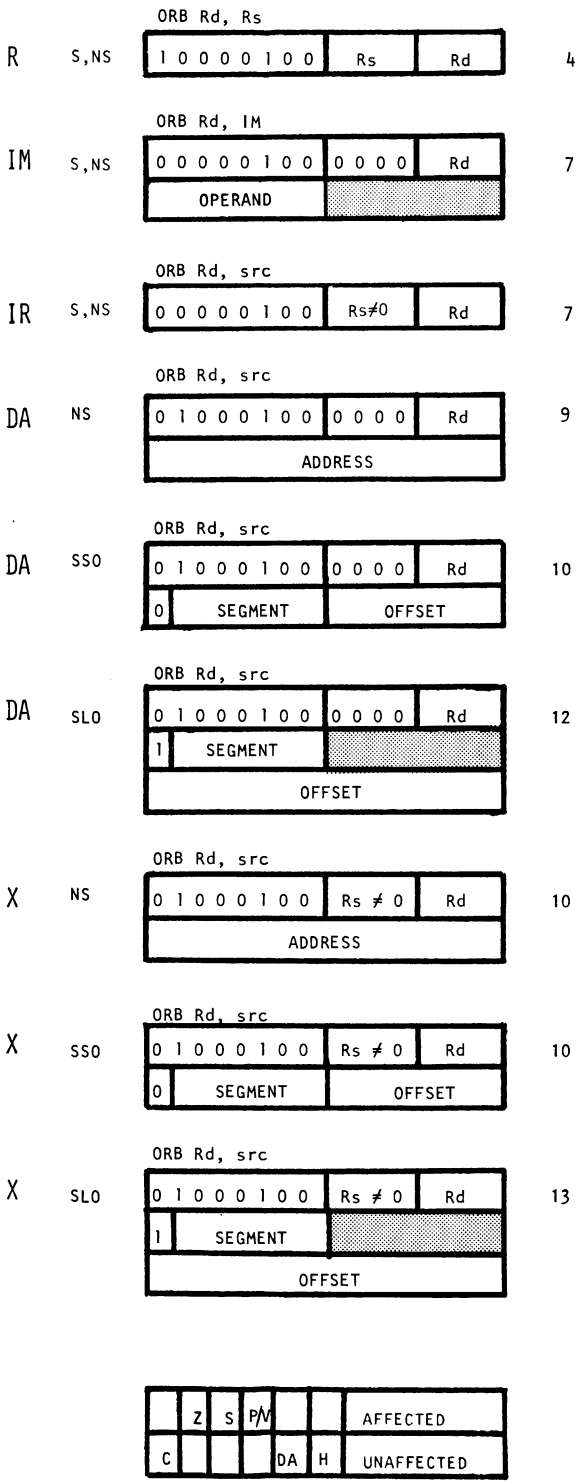
Logical OR operation is performed between corresponding bits of the source and destination words. The source operand is obtained using the applicable addressing mode and the destination is always a general purpose register designated by the Rd field of the instruction. The 16-bit result is loaded into the destination. The source operand is not altered and original destination operand is lost.

Flags:

Z: Set to 1 if result is zero.
Reset otherwise.
S: Set to 1 if result is negative.
Reset otherwise.

OR byte with register

ORB



Operation

dst<0:7> ← src<0:7> ∨ dst<0:7>

Description

A logical OR operation is performed between corresponding bits of the source and destination bytes. The source byte is obtained using the applicable addressing mode and the destination byte is always a general purpose register designated by the Rd field of the instruction. The 8-bit result is loaded into the destination. The source byte is not altered and the original byte in the destination register is lost.

Flags:

Z: Set to 1 if result is zero. Reset otherwise.
S: Set to 1 if result is negative. Reset otherwise.
P/V: Set to 1 if parity of result is even. Reset otherwise.

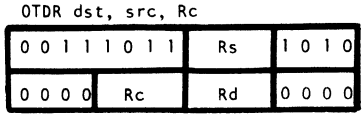
OUTPUT word from memory to I/O port, autodecrement and repeat.

OTDR

This is a system instruction.

OTDR dst, src, Rc

IR S,NS



11 + 10n*

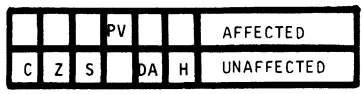
*n is the number of iterations

Operation

dst<0:15> ← src<0:15>
Rs<0:15> ← Rs<0:15>- 2
Rc<0:15> ← Rc<0:15>- 1
repeat until termination

Description

A Data word in memory, addressed by the contents of the general purpose register designated by the Rs field of the instruction, is loaded into the destination port. The destination is addressed by the contents of the general purpose register designated by the Rd field of the instruction. The source contents are unaltered. The contents of the general purpose register designated by Rs are then decremented by 2. The contents of the general purpose register designated by Rc are decremented by 1. The instruction is terminated when the result of this decrementation reaches zero. This instruction is interruptible.



Flags:

P/V: Set to 1.

OUTPUT byte to I/O port, from memory, autodecrement and repeat — OTDRB

This is a system instruction.

IR S,NS

OTDRB dst, src, Rc

0	0	1	1	1	0	1	0	Rs				1	0	1	0
0	0	0	0	Rc				Rd				0	0	0	0

11 + 10n*

*n is the number of iterations

Operation

dst<0:7> ← src<0:7>
Rs<0:15> ← Rs<0:15>- 1
Rc<0:15> ← Rc<0:15>- 1
repeat until termination

Description

A Data byte in memory, addressed by the contents of the general purpose register designated by the Rs field of the instruction, is loaded into the destination port. The destination is addressed by the contents of the general purpose register designated by the Rd field of the instruction. The source contents are unaltered. The contents of the general purpose register designated by Rs are then decremented by 1. The contents of the general purpose register designated by the Rc field is decremented by 1. The instruction is terminated when the result of this decrementation reaches zero. This instruction is interruptible.

			PV			AFFECTED
C	Z	S		DA	H	UNAFFECTED

Flags:

P/V: Set to 1.

OUTPUT word to I/O port, from memory, autoincrement and repeat

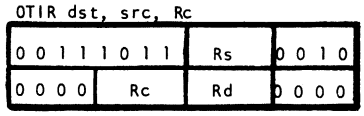
OTIR

This is a system instruction.

IR

OTIR dst, src, Rc

S,NS



11 + 10n*

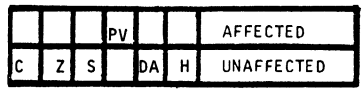
*n is the number of iterations

Operation

dst<0:15> ← src<0:15>
Rs<0:15> ← Rs<0:15>+ 2
Rc<0:15> ← Rc<0:15>- 1

Description

A data word in memory, addressed by the contents of the general purpose register designated by the Rs field of the instruction, is loaded into the destination port. The destination is addressed by the contents of the general purpose register designated by the Rd field of the instruction. The source contents are unaltered. The contents of the general purpose register designated by Rs are then incremented by 2. The contents of the general purpose register designated by Rc are decremented by 1. This instruction terminates when the result of this decrementation reaches zero. This instruction is interruptible.



Flags:

P/V: Set to 1.

OUTPUT byte to I/O Port from memory autoincrement and repeat.

OTIRB

This is a system instruction.

IR S,NS

OTIRB dst, src, Rc

0 0 1 1 1 0 1 0		Rs	0 0 1 0
0 0 0 0	Rc	Rd	0 0 0 0

11 + 10n*

*n is the number of iterations.

Operation

dst<0:7> ← src<0:7>

Rs<0:15> ← Rs<0:15>+ 1

Rc<0:15> ← Rc<0:15>- 1

Description

A data byte in the memory, addressed by the contents of the general purpose register designated by the Rs field of the instruction, is loaded into the destination port. The destination is addressed by the contents of the general purpose register designated by the Rd field of the instruction. The source contents are unaltered. The contents of the general purpose register designated by Rs are then incremented by 1. The contents of the general purpose register designated by Rc are decremented by 1. This instruction terminates when the result of this decrementation reaches zero. This instruction is interruptible.

			PV			AFFECTED
C	Z	S		DA	H	UNAFFECTED

Flags:

P/V: Set to 1.

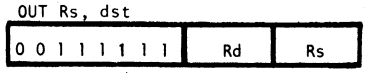
OUTPUT word to I/O port from register.

OUT

This is a system instruction

OUT Rs, dst

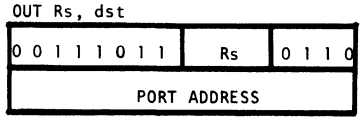
IR S,NS



10

OUT Rs, dst

DA S,NS



12

Operation

dst<0:15> ← Rs<0:15>

Description

The contents of the general purpose word source register designated by the Rs field of the instruction are loaded into an output port. The port address is determined by the applicable addressing mode. The source contents are unaltered.

						AFFECTED
C	Z	S	PV	DA	H	UNAFFECTED

Flags are not affected.

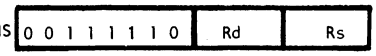
OUTPUT byte to I/O port from register

OUTB

This is a system instruction.

OUTB Rs, dst

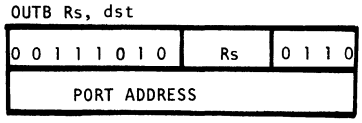
IR S,NS



10

OUTB Rs, dst

DA S,NS



12

Operation

dst<0:7> ← Rs<0:7>

Description

The contents of the general purpose byte source register designated by the Rs field of the instruction are loaded into an output port. The port address is determined by the applicable addressing mode. The source contents are unaltered.

						AFFECTED
C	Z	S	PV	DA	H	UNAFFECTED

Flags are not affected.

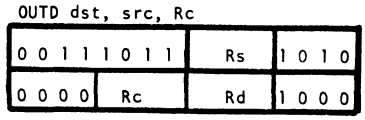
OUTPUT word to I/O Port from memory autodecrement.

OUTD

This is a system instruction.

OUTD dst, src, Rc

IR S,NS



21

Operation

dst<0:15> ← src<0:15>
Rs<0:15> ← Rs<0:15>- 2
Rd<0:15> ← Rd<0:15>- 2
Rc<0:15> ← Rc<0:15>- 1

Description

Data word in memory, addressed by the contents of the general purpose register designated by the Rs field of the instruction, is loaded into the destination port. The destination is addressed by the contents of the general purpose register designated by the Rd field of the instruction. The source contents are unaltered. The contents of the general purpose register designated by Rs are then decremented by 2. The contents of the general purpose register designated by Rc are decremented by 1.

			PV			AFFECTED
C	Z	S		DA	H	UNAFFECTED

Flags:

P/V: Set to 1 if result of decrementing Rc is zero. Reset otherwise.

OUTPUT byte to I/O port from memory, autodecrement

OUTDB

This is a system instruction

OUTDB dst, src, Rc

IR S,NS

0	0	1	1	1	0	1	0	Rs				1	0	1	0
0	0	0	0	Rc				Rd				1	0	0	0

21

Operation

dst<0:7> ← src<0:7>

Rs<0:15> ← Rs<0:15>- 1

Rc<0:15> ← Rc<0:15>- 1

Description

Data byte in memory, addressed by the contents of the general purpose register designated by the Rs field of the instruction, is loaded into the destination port. The destination is addressed by the contents of the general purpose register designated by the Rd field of the instruction. The source contents are unaltered. The contents of the general purpose register designated by Rs are then decremented by 1. The contents of the general purpose register designated by Rc are decremented by 1.

			PV			AFFECTED
C	Z	S		DA	H	UNAFFECTED

Flags:

P/V: Set to 1 if result of decrementing Rc is zero. Reset otherwise.

OUTPUT word to I/O port
from memory autoincrement

OUTI

This is a system instruction

OUTI dst, src, Rc

IR

0	0	1	1	1	0	1	1	Rs				0	0	1	0
0	0	0	0	Rc				Rd				1	0	0	0

21

Operation

dst<0:15> ← src<0:15>

Rs <0:15> ← Rs<0:15>+ 2

Rc <0:15> ← Rc<0:15>- 1

Description

A Data word in memory, addressed by the contents of the general purpose register designated by the Rs field of the instruction, is loaded into the destination port. The destination is addressed by the contents of the general purpose register designated by the Rd field of the instruction. The contents of the general purpose register designated by Rs are then incremented by 2. The contents of the general purpose register designated by Rc are decremented by 1.

			PV			AFFECTED
C	Z	S		DA	H	UNAFFECTED

Flags:

P/V: Set to 1 if result of decrementing Rc is zero. Reset otherwise.

OUTPUT byte to I/O port
from memory, autoincrement

OUTIB

This is a system instruction

OUTIB dst, src, Rc

IR S,NS

0 0 1 1 1 0 1 0	Rs	0 0 1 0	21
0 0 0 0 Rc	Rd	1 0 0 0	

Operation

dst<0:7> ← src<0:7>
Rs<0:15> ← Rs<0:15>+ 1
Rc<0:15> ← Rc<0:15>- 1

Description

Data byte in memory, addressed by the contents of the general purpose register designated by the Rs field of the instruction, is loaded into the destination port. The destination is addressed by the contents of the general purpose register designated by the Rd field of the instruction. The source contents are unaltered. The contents of the general purpose register designated by Rs are then incremented by 1. The contents of the general purpose register designated by Rc are decremented by 1.

			PV			AFFECTED
C	Z	S		DA	H	UNAFFECTED

Flags:

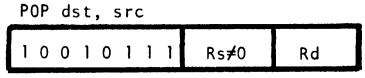
P/V: Set to 1 if result of decrementing Rc is zero. Reset otherwise.

POP word

POP

POP dst, src

R S,NS



8

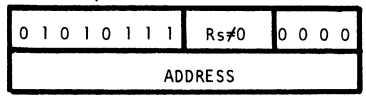
POP dst, src

IR S,NS

12

POP dst, src

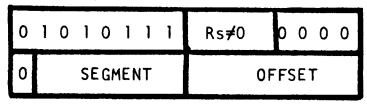
DA NS



15

POP dst, src

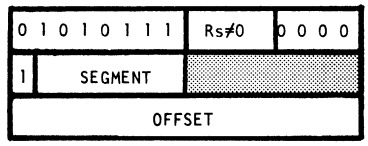
DA SSO



16

POP dst, src

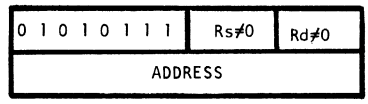
DA SLO



18

POP dst, src

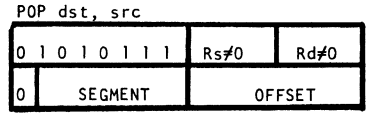
X NS



16

POP dst, src

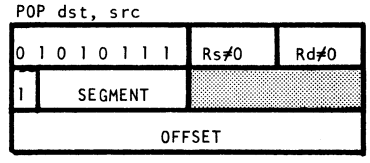
X SSO



16

POP dst, src

X SLO

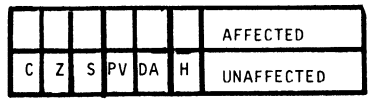


19

Operation

Description

The word from the memory location addressed by the general purpose register designated by Rs, is loaded into the destination. The contents of the register designated by Rs are then automatically incremented by 2. Thus, if the general purpose register designated by Rs is regarded as a stack pointer, then the operation described above can be regarded as a POP. Any general purpose register except RØ may be utilized as a stack pointer. The destination is determined by the applicable addressing mode.



Flags are not affected.

POPL

R S,NS

POPL dst, src

12

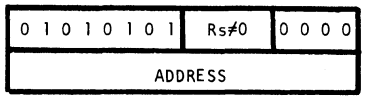
IR S,NS

POPL dst, src

19

DA NS

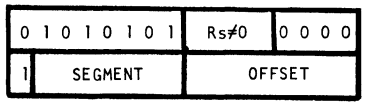
POPL dst, src



22

DA SSO

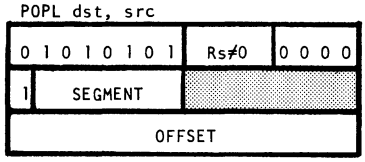
POPL dst, src



23

DA SLO

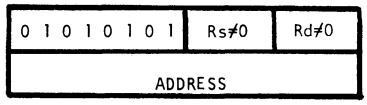
POPL dst, src



25

X NS

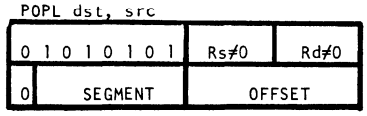
POPL dst, src



23

X SSO

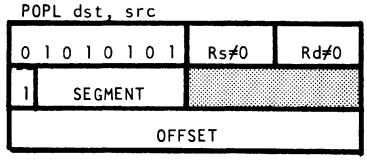
POPL dst, src



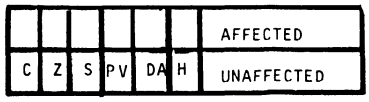
23

X SLO

POPL dst, src



26



Flags are not affected.

Operation

Description

The long word from the memory location addressed by the general purpose register designated by Rs, is loaded into the destination. The contents of the register designated by Rs are then automatically incremented by 4. Thus, if the general purpose register designated by Rs is regarded as a stack pointer, then the operation described above can be regarded as a POP. Any general purpose register except RØ may be utilized as a stack-pointer. The destination operand is determined by the applicable addressing mode.

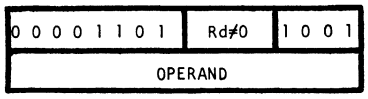
PUSH word

PUSH

R S,NS — PUSH IR, src

9

IM S,NS — PUSH IR, src

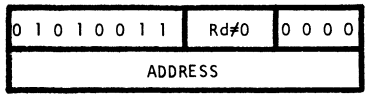


12

IR S,NS — PUSH IR, src

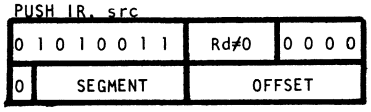
13

DA NS — PUSH IR, src



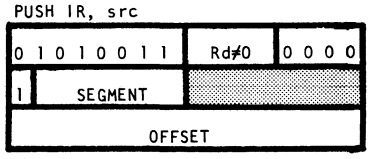
13

DA SSO — PUSH IR, src



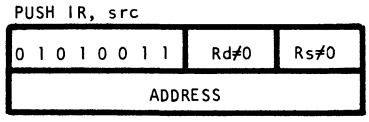
14

DA SLO — PUSH IR, src



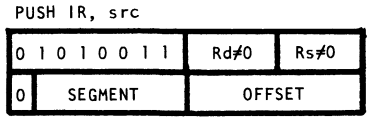
16

X NS — PUSH IR, src



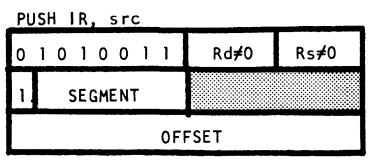
14

X SSO — PUSH IR, src

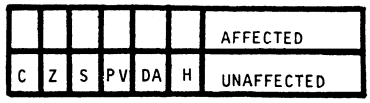


14

X SLO — PUSH IR, src



17



Operation

Description

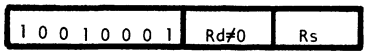
The contents of the register designated by the Rd field of the instruction are decremented by 2. The source word operand is then loaded into the the memory location addressed by the general purpose register designated in the Rd field of the instruction. Thus, if the general purpose register designated by Rd is regarded as a stack pointer, then the operation described above can be regarded as a PUSH. Any general purpose register except R0 can be utilized as a stack pointer. The source operand is determined by the applicable addressing mode.

Flags are not affected.

PUSHL

PUSHL IR, src

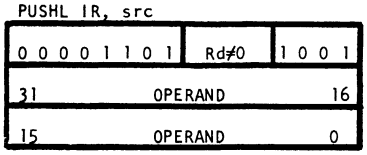
R S,NS



12

PUSHL IR, src

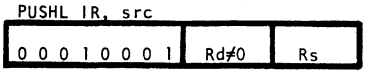
IM S,NS



19

PUSHL IR, src

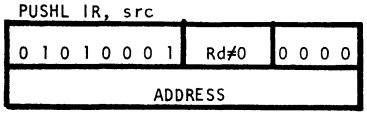
IR S,NS



20

PUSHL IR, src

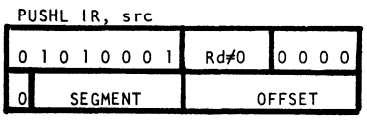
DA NS



20

PUSHL IR, src

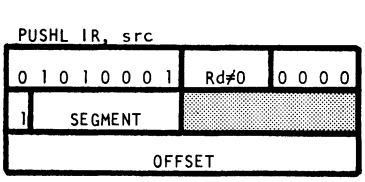
DA SSO



21

PUSHL IR, src

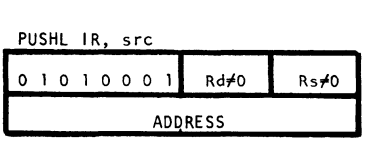
DA SLO



23

PUSHL IR, src

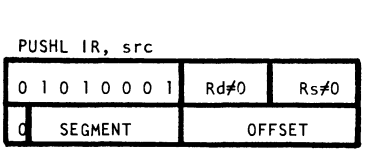
X NS



21

PUSHL IR, src

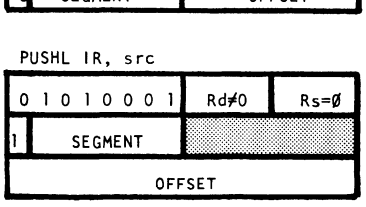
X SSO



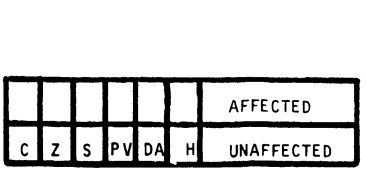
21

PUSHL IR, src

X SLO



24



Operation

Description

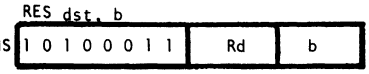
The contents of the register designated by the Rd field of the instruction are decremented by 4. The source long word operand is then loaded into the memory location addressed by the general purpose register designated in the Rd field of the instruction. Thus, if the general purpose register designated by Rd is regarded as a stack pointer, then the operation described above can be regarded as a PUSH. Any general purpose register except R0 can be utilized as a stack pointer. The source operand is determined by the applicable addressing mode.

Flags are not affected.

RESET bit in word

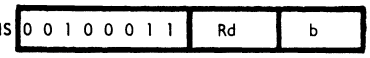
RES

R — S,NS — RES dst, b



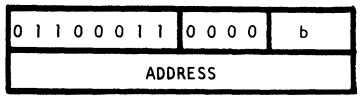
4

IR — S,NS — RES dst, b



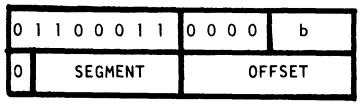
11

DA — NS — RES dst, b



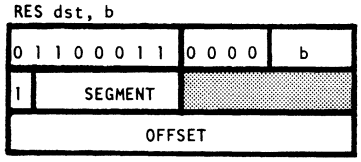
13

DA — SSO — RES dst, b



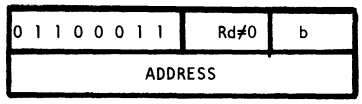
14

DA — SLO — RES dst, b



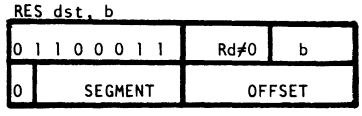
16

X — NS — RES dst, b



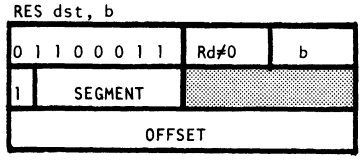
14

X — SSO — RES dst, b

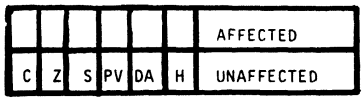


14

X — SLO — RES dst, b



17



Flags are not affected.

Operation

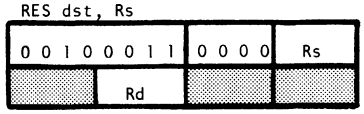
Description

The selected bit of the word destination is reset to Ø. The remaining 15 bits are unaltered. The destination is determined by the applicable addressing mode, while the bit to be reset is determined by the binary value of the b field of the instruction.

RESET bit in word, (dynamic) **RES**

RES dst, Rs

R S,NS



10

Operation

Description

The selected bit of the word destination is reset to Ø. The destination word operand is the general purpose register designated by the Rd field of the instruction. The bit of the destination register to be reset is determined by binary decode of the least significant 4 bits of a general purpose word register. This register is designated by the Rs field of the instruction. The remaining 15 bits of the destination are unaltered.

						AFFECTED
C	Z	S	PV	DA	H	UNAFFECTED

Flags are not affected.

RESET bit in byte (static) RESB

R S,NS

RESB dst, b

1 0 1 0 0 0 1 0	Rd	b

4

IR S,NS

RESB dst, b

0 0 1 0 0 0 1 0	Rd	b

11

DA NS

RESB dst, b

0 1 1 0 0 0 1 0	0 0 0 0	b
ADDRESS		

13

DA SSO

RESB dst, b

0 1 1 0 0 0 1 0	0 0 0 0	b
0 SEGMENT	OFFSET	

14

DA SLO

RESB dst, b

0 1 1 0 0 0 1 0	0 0 0 0	b
1 SEGMENT		
OFFSET		

16

X NS

RESB dst b

0 1 1 0 0 0 1 0	Rd≠0	b
ADDRESS		

14

X SSO

RESB dst, b

0 1 1 0 0 0 1 0	Rd≠0	b
0 SEGMENT	OFFSET	

14

X SLO

RESB dst, b

0 1 1 0 0 0 1 0	Rd≠0	b
1 SEGMENT		
OFFSET		

17

						AFFECTED
C	Z	S	PV	DA	H	UNAFFECTED

Operation

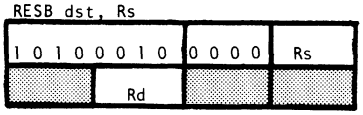
Description

The selected bit of the byte destination is reset to Ø. The remaining 7 bits are unaltered. The destination is determined by the applicable addressing mode, while the bit to be reset is determined by the binary value of the b field of the instruction.

Flags are not affected.

R S,NS

RESB dst, Rs



10

Operation

Description

The selected bit of the byte destination is reset to Ø. The destination byte operand is the general purpose register designated by the Rd field of the instruction. The bit of the destination register to be reset is determined by binary decode of the least significant 3 bits of a general purpose word register. This register is designated by the Rs field of the instruction. The remaining 7 bits of the destination are unaltered.

						AFFECTED
C	Z	S	PV	DA	H	UNAFFECTED

Flags are not affected.

RESET FLAGS

RESFLG

RESFLG

S,NS	1 0 0 0 1 1 0 1	C Z S PV	0 0 1 1	7

Operation

Description

The CPU flags C,Z,S, and P/V are reset or unaltered, according to the bit settings in the instruction field as described in the table below.

Instruction bit	if = Ø	if 1
7	no effect	reset C flag
6	no effect	reset Z flag
5	no effect	reset S flag
4	no effect	reset P/V flag

C	Z	S	PV			AFFECTED
				DA	H	UNAFFECTED

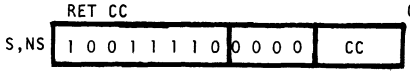
Flags:

See above

RETURN conditional from subroutine

RET

RET CC



CC True/CC false

10,13 / 7,7

Operation	
Non segmented	Segmented
if CC condition met	if CC condition met
PC ← (R15 <0:15>)	PC segment ← (RR14<0:22>)
R15<0:15> ← R15<0:15> + 2	R15<0:15> ← R15<0:15> + 2
	PC OFFSET ← (RR14<0:22>)
	R15<0:15> ← R15<0:15> + 2
otherwise	otherwise
PC ← PC + 2	PC OFFSET ← PC OFFSET + 2

Description

This instruction conditionally returns the CPU to the calling program. During a subroutine call the return address was automatically stacked. This return address is popped from the stack into the PC to effect the return. If the flags do not satisfy the conditions specified by the CC field, the PC is not loaded with the return address but merely updated to the following instruction . The stack pointer remains unaltered from its original value if there is no return.

						AFFECTED
C	Z	S	PV	DA	H	UNAFFECTED

Flags are not affected.

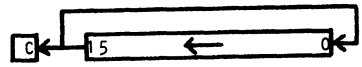
ROTATE word left. **RL**

RL Rd, n

R S,NS

6 (one place)
7 (two places)

Operation



Description

The contents of the general purpose word register designated by the Rd field of the instructions are rotated left. The number of places to be rotated is specified by bit 1 of the instruction; zero corresponds to one place and one corresponds to two places.

Flags:

C: loaded from last bit rotated out of destination register.
Z: **Set to 1 if result is zero.** Reset otherwise.
S: Set to 1 if result is negative. Reset otherwise.
P/V: Set to 1 if sign of destination register changed during rotation. Reset otherwise.

C	Z	S	P/V			AFFECTED
				DA	H	UNAFFECTED

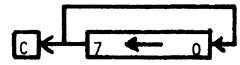
RLB Rd, n

R S,NS

1	0	1	1	0	0	1	0	Rd	1	0	b	0

6 (one place)
7 (two places)

Operation



Description

The contents of the general purpose byte register designated by the Rd field of the instruction are rotated left. The number of places to be rotated is specified by bit 1 of the instruction; zero corresponds to one place and one corresponds to two places.

Flags:

C: Loaded from last bit rotated out of destination register.
Z: Set to 1 if result is zero. Reset otherwise.
S: Set to 1 if result is negative. Reset otherwise.
P/V: Set to 1 if sign of register changed during rotation. Reset otherwise.

C	Z	S	P/V			AFFECTED
				DA	H	UNAFFECTED

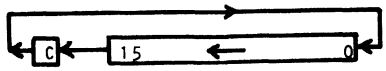
ROTATE word left through carry — RLC

RLC Rd, n

R S,NS | 1 0 1 1 0 0 1 1 | Rd | 0 0 b 0 |

6 (one place)
7 (two places)

Operation



Description

The contents of the destination word register, designated by the Rd field of the instruction, are rotated one or two places left. The most significant bit shifted out of the destination word is loaded into the carry flag, while the previous contents of the carry flag are shifted into the least significant bit of the destination word. The number of places to be rotated is specified by bit 1 of the instruction; zero corresponds to one place and one corresponds to two places.

Flags:

C: Loaded from most significant bit rotated out of destination register.
Z: Set to 1 if result is zero. Reset otherwise.
S: Set to 1 if result is negative. Reset otherwise.
P/V: Set to 1 if sign of destination contents changed during rotation. Reset otherwise.

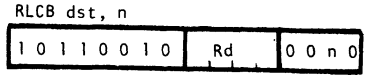
C	Z	S	P/V			AFFECTED
				DA	H	UNAFFECTED

ROTATE byte left through carry — RLCB

R

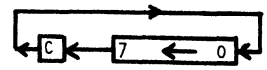
RLCB dst, n

S,NS



6 (one place)
7 (two places)

Operation



Description

The contents of the destination byte register, designated by the Rd field of the instruction are rotated one or two places left. The most significant bit out of the destination byte is loaded into the carry flag, while the previous contents of the carry flag are rotated into the least significant bit of the destination byte. The number of places to be rotated is specified by bit 1 of the instruction; zero corresponds to one place and one corresponds to two places.

Flags:

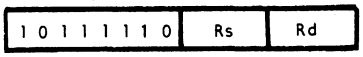
C: Loaded from most significant bit rotated out of destination.
Z: Set to 1 if result is zero. Reset otherwise.
S: Set to 1 if result is negative. Reset otherwise.
P/V: Set to 1 if sign of destination changed during rotation. Reset otherwise.

C	Z	S	P/V			AFFECTED
				DA	H	UNAFFECTED

ROTATE DIGIT LEFT, byte **RLDB**

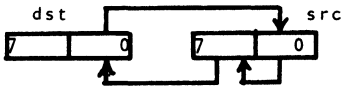
RLDB Rs, Rd

R S,NS



9

Operation



Description

The contents of the source and destination byte registers are exchanged as shown in the operation. Both the source and destination are general purpose byte registers designated by the Rs and Rd fields of the instruction respectively. The most significant 4 bits of the destination remain unchanged.

Flags:

Z: Set to 1 if destination result is zero. Reset otherwise.
S: Set to 1 if most significant bit of destination result is 1. Reset otherwise.

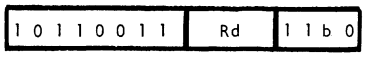
	Z	S				AFFECTED
C			PV	DA	H	UNAFFECTED

ROTATE word right.

RR

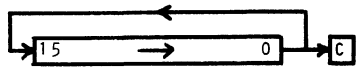
RR Rd, n

R S,NS



6 (one place)
7 (two places)

Operation



Description

The contents of the general purpose word register designated by the Rd field of the instructions are rotated right. The number of places to be rotated is specified by bit 1 of the instruction; zero corresponds to one place and one corresponds to two places.

C	Z	S	P/V			AFFECTED
				DA	H	UNAFFECTED

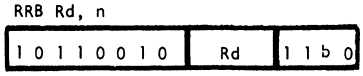
Flags:

C: loaded from last bit rotated out of destination register.
Z: Set to 1 if result is zero. Reset otherwise.
S: Set to 1 if result is negative. Reset otherwise.
P/V: Set to 1 if sign of destination register changed during rotation. Reset otherwise.

ROTATE BYTE right — RRB

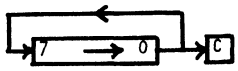
RRB Rd, n

R S,NS



6 (one place)
7 (two places)

Operation



Description

The contents of the general purpose byte register designated by the Rd field of the instructions are rotated right. The number of places to be rotated is specified by bit 1 of the instructions; zero corresponds to one place and one corresponds to two places.

Flags:

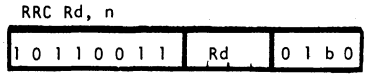
C: Loaded from least significant bit rotated out of destination register.
Z: Set to 1 if result is 0. Reset otherwise.
S: Set to 1 if result is negative. Reset otherwise.
P/V: Set to 1 if sign of destination register changed during rotation. Reset otherwise.

C	Z	S	P/V			AFFECTED
				DA	H	UNAFFECTED

ROTATE word right through carry — RRC

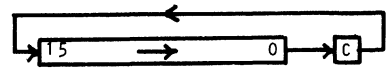
RRC Rd, n

R S,NS



6 (one place)
7 (two places)

Operation



Description

The contents of the destination word register, designated by the Rd field of the instruction are rotated one or two places right. The least significant bit rotated out of the destination word is loaded into the carry flag, while the previous contents of the carry flag are shifted into the most significant bit of the destination word. The number of places to be rotated is specified by by bit 1 of the instruction; zero corresponds to one place and one corresponds to two places.

Flags:

C: Loaded from least significant bit rotated out of destination.
Z: Set to 1 if result is zero. Reset otherwise.
S: Set to 1 if result is negative. Reset otherwise.
P/V: Set to 1 if sign of register changed during rotation. Reset otherwise.

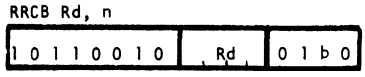
C	Z	S	P/V			AFFECTED
				DA	H	UNAFFECTED

ROTATE byte right through carry

RRCB

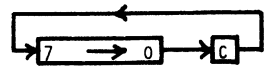
RRCB Rd, n

R S,NS



6 (one place)
7 (two places)

Operation



Description

The contents of the destination byte register, designated by the Rd field of the instruction are rotated one or two places right.

The least significant bit shifted out of the destination byte is loaded into the carry flag, while the previous contents of the carry flag are shifted into the most significant bit of the destination byte.

The number of rotated places to be rotated is specified by bit 1 of the instruction; zero corresponds to one place and one corresponds to two places.

C	Z	S	P/V			AFFECTED
				DA	H	UNAFFECTED

Flags:

C: Loaded from least significant bit shifted out of destination register.
Z: Set to 1 if result is zero. Reset otherwise.
S: Set to 1 if result is negative. Reset otherwise.
P/V: Set to 1 if sign of destination contents changes during rotation. Reset otherwise.

ROTATE DIGIT RIGHT , byte **RRDB**

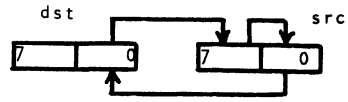
RRDB Rs, Rd

R S,NS

1 0 1 1 1 1 0 0	Rs	Rd

9

Operation



Description

The contents of the source and destination byte register are exchanged as shown in the operation. Both the source and destination are general purpose byte registers designated by the Rs and Rd fields of the instruction respectively. The most significant four bits of the destination remain unchanged.

	Z	S				AFFECTED
C			PV	DA	H	UNAFFECTED

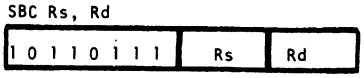
Flags:

Z: Set to 1 if destination result is zero. Reset otherwise.
S: Set to 1 if most significant bit of destination result is 1. Reset otherwise.

SUBTRACT word with carry **SBC**

SBC Rs, Rd

R S,NS



5

Operation

dst<0:15> ←dst<0:15> -src<0:15>-C

Description

The source operand word is subtracted from the destination operand word, along with carry, to obtain the result. The subtraction is achieved by adding the two's complement of the source operand to the destination operand.

Both the source and destination are general purpose word registers designated by the Rs and Rd fields of the instruction respectively. The 16 bit result is loaded into the destination register, whose original contents are lost. The contents of the source are not affected.

Flags:

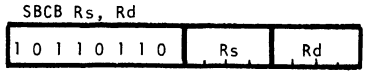
C: Reset to Ø on carry from most significant bit of result. Set otherwise.
Z: Set to 1 if result is zero. Reset otherwise.
S: Set to 1 if result is negative. Reset otherwise.
P/V: Set to 1 if there is arithmetic overflow. Reset otherwise.

C	Z	S	PV			AFFECTED
				DA	H	UNAFFECTED

SUBTRACT byte with carry **SBCB**

SBCB Rs, Rd

R S,NS



5

Operation

dst<0:7> ←dst<0:7>- src<0:7>- C

Description

The source operand byte is subtracted from the destination operand byte along with carry, to obtain the result. The subtraction is achieved by adding the two's complement of the source operand to the destination operand.

Both the source and destination are general purpose byte registers designated by the Rs and Rd fields of the instruction respectively. The 8-bit result is loaded into the destination register, whose original contents are lost. The contents of the source are not altered.

Flags:

C: Reset to Ø on carry from most significant bit of result. Set otherwise.
Z: Set to 1 if result is zero. Reset otherwise.
S: Set to 1 if result is negative. Reset otherwise.
P/V: Set to 1 if there is arithmetic overflow. Reset otherwise.
DA: Set to 1 always
H: Reset to Ø if there is a carry from the most significant bit of the lower 4 bits of the result. Set otherwise.

C	Z	S	P/V	DA	H	
						AFFECTED
						UNAFFECTED

SYSTEM CALL

SC

SC N

S,NS | 0 1 1 1 1 1 1 1 | N | 39, 33

Operation

Non Segmented

R15<0:15> ← R15<0:15>- 2
(R15<0:15>) ← PC<0:15>+ 2
-
-
R15<0:15> ← R15<0:15>- 2
(R15<0:15>)← FCW
R15<0:15> ← R15<0:15>- 2
(R15<0:15>) ← Identifier
FCW ← (NPSAP<0:15>+ 4)
PC ← (NPSAP<0:15>+ 6)

Segmented

R15<0:15> ← R15<0:15>- 2
(RR14<0:22>) ← PC OFFSET + 2
R15<0:15> ← R15<0:15>- 2
(RR14<0:22>) ← PC SEGMENT
R15<0:15> ← R15<0:15>- 2
(RR14<0:22>) ← FCW
R15<0:15> ← R15<0:15>- 2
(RR14<0:22>) ← Identifier
FCW ← (NPSAP<0:22>+ 10)
PC SEGMENT ← (NPSAP<0:22> + 12)
PC OFFSET ← (NPSAP<0:22> + 14)

Description

This instruction produces a system call trap. The system call causes the program status to be pushed into the system stack and then loads the new processor status using NPSAP.

The status stored on the stack comprises the program counter return address, and the flag control word (FCW) as well as the system call instruction itself, as the Identifier.

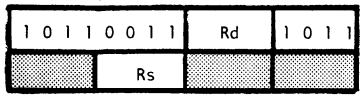
The new program counter and FCW are obtained from the NPSAP and are loaded into the relevant CPU registers to cause the transfer of control. The 8 bit N field of the instruction is user definable, and thus allows up to 256 identifiers.

C	Z	S	PV	DA	H	
						AFFECTED
						UNAFFECTED

Flags:

As specified by the new FCW.

SDA dst, Rs



15 + 3n*

Operation

dst<0:15> ← dst<0:15> shifted

*n is the number of places shifted.

Description

The contents of a general purpose word register designated by the Rd field of the instruction are shifted. The magnitude and direction of the shift are determined from the contents of the general purpose word register designated by the Rs field of the instruction. The register contains a signed 2's complement integer, which is used to determine the shift value. A positive number indicates a left shift, and a negative number indicates a right shift. The magnitude of the shift must be in the range -16 to +16.

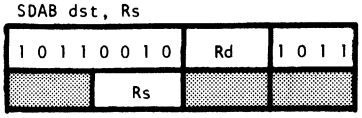
This operation is identical to the operation SDL apart from the treatment of the most significant bit of the word, bit 15. This bit is unaltered during right shifts, and shifts into the adjacent bit, bit 14. For left shifts, the bit is treated in an identical manner to other bits of the register. Thus a signed operand has the sign preserved during right shifts.

C	Z	S	PV			AFFECTED
				DA	H	UNAFFECTED

Flags:

C: Loaded from bit 15 shifted out of destination register (left shift) or from bit Ø shifted out of the destination register (right shift).
Z: Set to 1 if the result is zero. Reset otherwise.
S: Set to 1 if most significant bit of resultant destination register is 1. Reset otherwise.
P/V: Set to 1 if sign of destination register is changed during shift. Reset otherwise.

SDAB dst, Rs



15 + 3n*

*n is the number of places shifted.

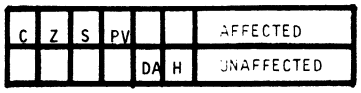
Operation

dst<0:7> ← dst<0:7> shifted

Description

The contents of a general purpose byte register designated by the Rd field of the instruction are shifted. The magnitude and direction of the shift are determined from the contents of the general purpose word register designated by the Rs field of the instruction. The register contains a signed 2's complement integer, which is used to determine the shift value. A positive number indicates a left shift and a negative number indicates a right shift. The magnitude of the shift must be in the range -8 to +8.

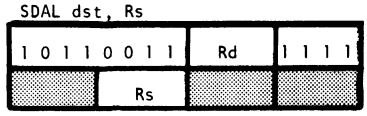
This operation is identical to the operation SDLB apart from the treatment of the most significant bit of the byte, bit 7. This bit is unaltered during right shifts, and shifts into the adjacent bit, bit 6. For left shifts, the bit is treated in an identical manner to other bits of the register. Thus a signed operand has the sign preserved during right shifts.



Flags

C: Loaded from bit 7 shifted out of destination register (left shift) or from bit Ø shifted out of the destination register (right shift).
Z: Set to 1 if the result is zero. Reset otherwise.
S: Set to 1 if most significant bit of resultant destination register is 1. Reset otherwise.
P/V: Set to 1 if sign of destination register is changed during shift. Reset otherwise.

SDAL dst, Rs



15 + 3n*

*n is the number of places shifted.

Operation

dst<0:31> ← dst<0:31> shifted

Description

The contents of a general purpose long word register designated by the Rd field of the instruction are shifted. The magnitude and direction of the shift are determined from the contents of the general purpose register designated by the Rs field of the instruction. The register contains a signed 2's complement integer, which is used to determine the shift value. A positive number indicates a left shift, and a negative number indicates a right shift. The magnitude of the shift must be in the range -32 to +32.

This operation is identical to the operation SDLL apart from the treatment of the most significant bit of the long word, bit 31. This bit is unaltered into the adjacent bit, bit 30. For left shifts, the bit is treated to other bits of the register. Thus a signed operand has the sign preserved during right shifts.

C	Z	S	PV			AFFECTED
				DA	H	UNAFFECTED

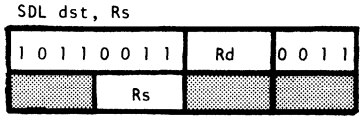
Flags :

C: Loaded from bit 31 shifted out of destination register (left shift) or from bit Ø shifted out of the destination register (right shift).
Z: Set to 1 if the result is zero. Reset otherwise.
S: Set to 1 if most significant bit of resultant destination register is 1. Reset otherwise.
P/V: Set to 1 if sign of destination register is changed during shift. Reset otherwise.

SDL

SDL dst, Rs

R



15 + 3n*

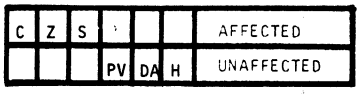
*n is the number of places shifted

Operation

dst<0:15> ← dst<0:15>shifted

Description

The contents of a general purpose word register designated by the Rd field of the instruction are shifted. The magnitude and direction of the shift are determined from the contents of the general purpose word register designated by the Rs field of the instruction. The register contains a signed 2's complement integer, which is used to determine the shift value. A positive number indicates a left shift, and a negative number indicates a right shift. The magnitude of the shift must be in the range -16 to +16.

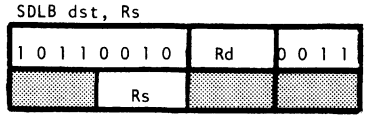


Flags:

C: Loaded from the last bit shifted out of the destination register.
Z: Set to 1 if the result is zero. Reset otherwise.
S: Set to 1 if result is negative. Reset otherwise.
P/V: Undefined.

SDLB

SDLB dst, Rs



15 + 3n*

*n is the number of places shifted

Operation

dst <0:7> ← dst <0:7> (shifted)

Description

The contents of a general purpose byte register designated by the Rd field of the instruction are shifted. The magnitude and direction of the shift are determined from the contents of the general purpose byte register designated by the Rs field of the instruction. The register contains a signed 2's complement integer, which is used to determine the shift value. A positive number indicates a left shift, and a negative number indicates a right shift. The magnitude of the shift must be in the range -8 to +8.

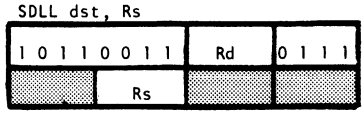
C	Z	S				AFFECTED
			PV	DA	H	UNAFFECTED

Flags:

C: Loaded from the last bit shifted out of the destination register
Z: Set to 1 if the result is Ø. Reset otherwise.
S: Set to 1 if the result is negative. Reset otherwise.
P/V: Undefined.

SDLL

SDLL dst, Rs



15 + 3n*

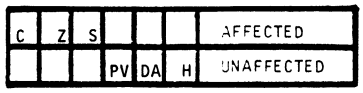
*n is the number of places shifted.

Operation

dst<0:31> ← dst<0:31>(shifted)

Description

The contents of a general purpose register pair designated by the Rd field of the instruction are shifted. The magnitude and direction of the shift are determined from the contents of the general purpose word register designated by the Rs field of the instruction. The register contains a signed 2's complement integer, which is used to determine the shift value. A positive number indicates a left shift, and a negative number indicates a right shift. The magnitude of the shift must be in the range -32 to +32.

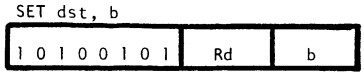


Flags:

C: Loaded from the last bit shifted out of the destination register.
Z: Set to 1 if the result is zero. Reset otherwise.
S: Set to 1 if result is negative. Reset otherwise.
P/V: Undefined.

R S,NS

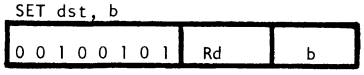
SET dst, b



4

IR S,NS

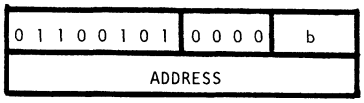
SET dst, b



11

DA NS

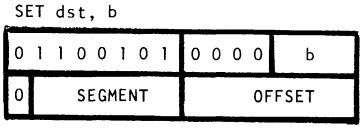
SET dst, b



13

DA SSO

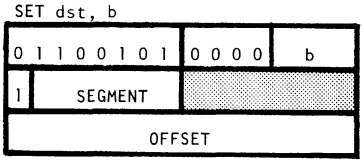
SET dst, b



14

DA SLO

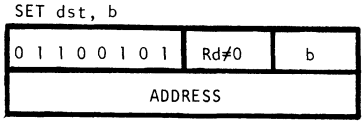
SET dst, b



16

X NS

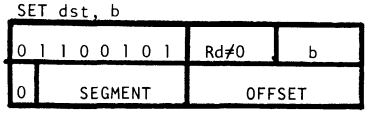
SET dst, b



14

X SSO

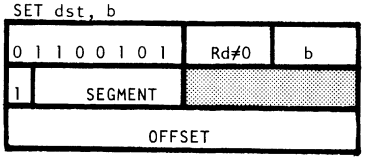
SET dst, b



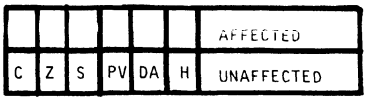
14

X SLO

SET dst, b



17



Flags are not affected.

Operation

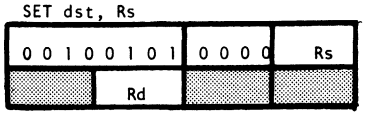
Description

The selected bit of the word destination is set to 1. The remaining 15 bits are unaltered. The destination is determined by the applicable addressing mode, while the bit to be set is determined by the binary value of the b field of the instruction.

SET bit in word (dynamic) **SET**

SET dst, Rs

R S,NS

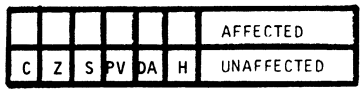


10

Operation

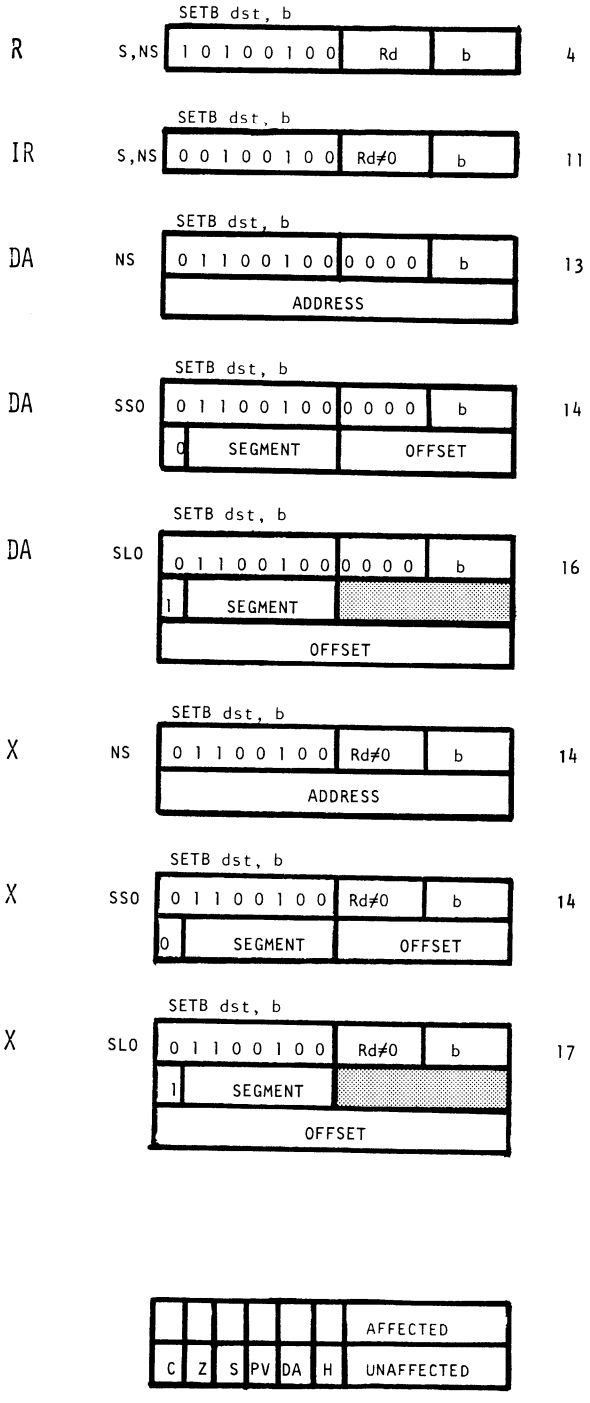
Description

The selected bit of the word destination is set to 1. The destination word operand is the general purpose register designated by the Rd field of the instruction. The bit of the destination register to be set is determined by a binary decode of the least significant 4 bits of a general purpose word register. This register is designated by the Rs field of the instruction. The remaining 15 bits of the destination are unaltered.



Flags are not affected.

SET bit in byte (static) — SETB



Operation

Description

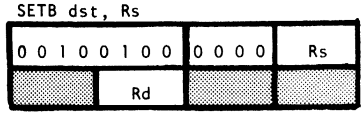
The selected bit of the byte destination is set to 1. The remaining 7 bits are unaltered. The destination is determined by the applicable addressing mode, while the bit to be set to 1 is determined by the binary value of the b field of the instruction.

Flags are not affected.

SET bit in byte (dynamic) **SETB**

SETB dst, Rs

R S,NS

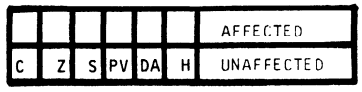


10

Operation

Description

The selected bit of the byte destination is set to 1. The destination byte operand is the general purpose register designated by the Rd field of the instruction. The bit of the destination register to be set is determined by binary decode of the least signficant 3 bits of a general purpose word register. This register is designated by the Rs field of the instruction. The remaining 7 bits of the destination are unaltered.



Flags are not affected.

SETFLG

SETFLG

(S,NS) | 1 0 0 0 1 1 0 1 | C Z S PV | 0 0 0 1 |

7

Operation

Description

The CPU flags C,Z,S and P/V are set or unaltered, according to the bit settings in the instruction field as described in the table below.

Instruction bit	If Ø	If 1
7	no effect	set C flag
6	no effect	set Z flag
5	no effect	set S flag
4	no effect	set P/V flag

C	Z	S	PV			AFFECTED
				DA	H	UNAFFECTED

Flags:

See above.

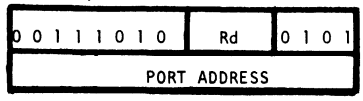
SPECIAL INPUT byte to register from I/O port

SINB

This is a system instruction

DA

SINB Rd, src



12

Operation

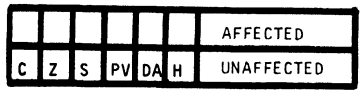
Rd<0:7> ← src<0:7>

Description

A general purpose byte destination register designated by the Rd field of the instruction is loaded from an input port.

The port address is determined directly from the instruction. The original contents of the destination are lost.

The instruction is similar in operation to the corresponding standard I/O instruction. The significant difference is that the data byte is transferred on the most significant eight bus lines. For standard I/O instructions, transfers take place on the least significant eight lines.



Flags are not affected.

SPECIAL INPUT byte from I/O port to memory, autodecrement.

SINDB

This is a system instruction.

SINDB dst, src, Rc

IR S,NS

0 0 1 1 1 0 1 0	Rs	1 0 0 1
0 0 0 0	Rc	Rd	1 0 0 0

21

Operation

dst<0:7> ← src<0:7>

Rd<0:15> ← Rd<0:15> - 1

Rc<0:15> ← Rc<0:15> - 1

Description

Data byte from the port addressed by the contents of the general purpose register designated by the Rs field of the instruction is loaded into a memory destination. The destination is addressed by the contents of the general purpose register designated by the Rd field of the instruction. The original contents of the destination are lost. The contents of the general purpose registers designated by Rd and Rc are then decremented by 1.

This instruction is similar in operation to the corresponding standard I/O instruction. The significant difference is that the data byte is transferred on the most significant eight bus lines. For standard I/O instructions transfers take place on the least significant eight lines.

			PV			AFFECTED
C	Z	S		DA	H	UNAFFECTED

Flags:

P/V: **Set to 1 if result of decrementing Rc register is zero. Reset otherwise.**

SPECIAL INPUT byte from I/O port to memory, autodecrement and repeat.

SINDRB

This is a system instruction.

SINDRB dst,src.Rc

IR S,NS

0 0 1 1 1 0 1 0	Rs	1 0 0 1
0 0 0 0 Rc	Rd	0 0 0 0

11 + 10n*

* n is the number of iterations.

Operation

dst<0:7> ← src<0:7>

Rd<0:15> ← Rd<0:15> - 1

Rc<0:15> ← Rc<0:15> - 1

repeat until termination

Description

Data byte from the port addressed by the contents of the general purpose register designated by the Rs field of the instruction is loaded into the memory destination. The destination is addressed by the contents of the general purpose register designated by the Rd field of the instruction. The original contents of the destination are lost. The contents of the general purpose register designated by Rd are then decremented by 1. The contents of the general purpose register designated by Rc are decremented by 1. The instruction is terminated when the result of this decrementation reaches zero. This instruction is interruptible.

This instruction is similar in operation to the corresponding standard I/O instruction. The significant difference is that the data byte is transferred on the most significant eight bus lines. For standard I/O instructions transfers take place on the least significant eight lines.

			PV			AFFECTED
C	Z	S		DA	H	UNAFFECTED

Flags;

P/V: Set to 1.

SPECIAL INPUT byte from I/O port to memory, autoIncrement.

SINIB

This is a system instruction

SINIB dst, src, Rc

IR S,NS

0	0	1	1	1	0	1	0	Rs				0	0	0	1
0	0	0	0	Rc				Rd				1	0	0	0

21

Operation

dst<0:7> ← src<0:7>

Rd<0:15> ← Rd<0;15> + 1

Rc<0:15> ← Rc<0:15> - 1

Description

Data byte from the port addressed by the contents of the general purpose register designated by the Rs field of the instruction is loaded into a memory destination. The destination is addressed by the contents of the general purpose register designated by the Rd field of the instruction. The original contents of the destination are lost. The contents of the general purpose registers designated by Rd are then incremented by 1. The contents of the general purpose register designated by Rc are decremented by 1.

This instruction is similar in operation to the corresponding standard I/O instruction. The significant difference is that the data byte is transferred on the most significant eight bus lines. For standard I/O instructions transfers take place on the least significant eight lines.

			PV			AFFECTED
C	Z	S		DA	H	UNAFFECTED

Flags:

P/V: Set to 1 if result of decrementing Rc register is zero. Reset otherwise.

SPECIAL INPUT byte from I/O port to memory, auto-increment and repeat

SINIRB

This is a system instruction.

SINIRB dst, src, Rc,

IR S,NS

0 0 1 1 1 0 1 0	Rs	0 0 0 1
0 0 0 0 Rc	Rd	0 0 0 0

11 + 10n*

*n is the number of iterations.

Operation

dst<0:7> ← src<0:7>

Rd<0:15> ← Rd<0:15> + 1

Rc<0:15> ← Rc<0:15> - 1

repeat until termination

Description

Data byte from the port addressed by the contents of the general purpose register designated by the Rs field of the instruction is loaded into a memory destination. The destination is addressed by the contents of the general purpose register designated by the Rd field of the instruction. The original contents of the destination are lost. The contents of the general purpose register designated by Rd are then incremented by 1. The contents of the general purpose register designated by Rc are decremented by 1. This instruction is terminated when the result of this decrementation reaches zero. This instruction is interruptible.

This instruction is similar in operation to the corresponding standard I/O instruction. The significant difference is that the data byte is transferred on the most significant eight bus lines. For standard I/O instructions transfers take place on the least significant eight lines.

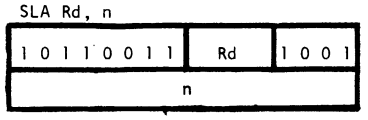
			PV			AFFECTED
C	Z	S		DA	H	UNAFFECTED

Flags:

P/V: Set to 1.

SLA Rd, n

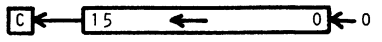
R S,NS



13 + 3n*

*n is the number of places shifted.

Operation



Description

The contents of the word destination register are shifted left. The destination is a general purpose word register designated by the Rd field of the instruction. The number of places to be shifted is determined from the value of the n field of the instruction. The magnitude of the shift must be in the range 0 to 16. The n field is a 16 bit positive integer in 2's complement notation.

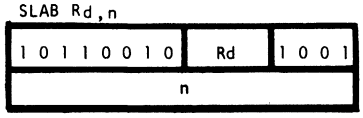
C	Z	S	PV			AFFECTED
				DA	H	UNAFFECTED

Flags:

C: Loaded from the most significant bit shifted out of the register.
Z: Set to 1 if result is zero. Reset otherwise.
S: Set if the most significant bit of the resultant destination is 1. Reset otherwise.
P/V: Set the 1 if sign of register changed during shift operation. Reset otherwise.

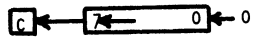
SLAB Rd,n

R S,NS



13 + 3n*

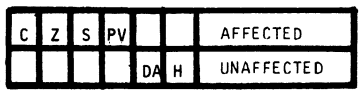
Operation



*n is the number of places shifted.

Description

The contents of the byte destination register are shifted left. The destination is a general purpose byte register designated by the Rd field of the instruction. The number of places to be shifted is determined from the value of the n field of the instruction. The magnitude of the shift must be in the range 0 to 8. The n field is a 16 bit positive integer in 2's complement notation.

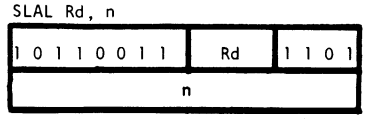


Flags:

C: Loaded from the most significant bit shifted out of the register.
Z: Set to 1 if result is zero. Reset otherwise.
S: Set if the most significant bit of the resultant destination is 1. Reset otherwise.
P/V: Set to 1 if sign of register changed during shift operation. Reset otherwise.

SLAL Rd, n

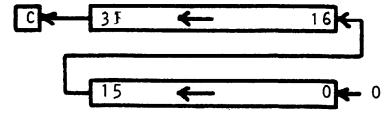
R S,NS



13 + 3n*

*n is the number of places shifted

Operation



Description

The contents of the long word destination register are shifted left. The destination is a general purpose long word register designated by the Rd field of the instruction. The number of places to be shifted is determined from the value of the n field of the instruction. The magnitude of the shift must be in the range 0 to 32. The n field is a 16 bit positive integer in 2's complement notation.

C	Z	S	PV			AFFECTED
				DA	H	UNAFFECTED

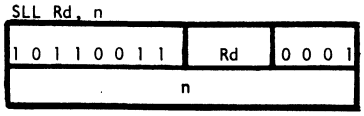
Flags;

C: Loaded from the most significant bit shifted out of the register.
Z: Set to 1 if result is zero. Reset otherwise.
S: Set if the most significant bit of the resultant destination is 1. Reset otherwise.
P/V: Set to 1 if sign of register changed during shift operation. Reset otherwise.

SHIFT word logical left. **SLL**

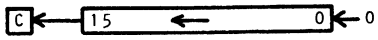
R S,NS

SLL Rd, n



13 + 3n*

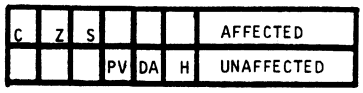
Operation



*n is the number of places shifted.

Description

The contents of the word destination register are shifted left. The destination is a general purpose word register designated by the Rd field of the instruction. The number of places to be shifted is determined from the value of the n field of the instruction. The magnitude of the shift must be in the range 0 to 16. The n field is a 16 bit positive integer in 2's complement notation.



Flags:

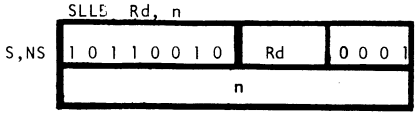
C: Loaded from the last bit shifted out of the register.
Z: Set to 1 if result is zero. Reset otherwise.
S: Set if the most significant bit of the resultant destination is 1. Reset otherwise.
P/V: Undefined.

SHIFT byte logical left.

SLLB

R

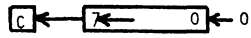
SLLB Rd, n



13 + 3n*

*n is the number of places shifted

Operation



Description

The contents of the byte destination register are shifted left. The destination is a general purpose byte register designated by the Rd field of the instruction. The number of places to be shifted is determined from the value of the n field of the instruction. The magnitude of the shift must be in the range 0 to 8. The n field is a 16 bit positive integer in 2's complement notation.

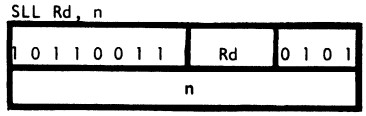
C	Z	S				AFFECTED
			PV	DA	H	UNAFFECTED

Flags:

C: Loaded from the last bit shifted out of the register.
Z: Set to 1 if result is zero. Reset otherwise.
S: Set if the most significant bit of the resultant destination is 1. Reset otherwise.
P/V: Undefined.

SLL Rd, n

R S,NS



13 + 3n*

*n is the number of places shifted

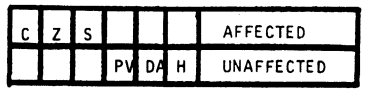
Operation

dst<0:31> ← dst<0:31>shifted

Description

The contents of the register pair are shifted left. The register pair is designated by the Rd field of the instruction. The magnitude of the shift is determined from the value of the n field of the instruction. The magnitude of the shift must be in the range 0 to 32. The n field is a 16 bit positive integer in 2's complement notation.

Flags:



C: Loaded from the most significant bit shifted out of the register.
Z: Set to 1 if result is zero. Reset otherwise.
S: Set if the most significant bit of the resultant destination is 1. Reset otherwise.
P/V: Undefined.

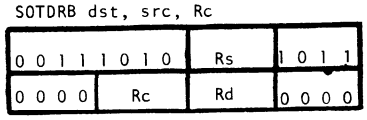
SPECIAL OUTPUT byte from memory to I/O port, autodecrement and repeat.

SOTDRB

This is a system instruction.

SOTDRB dst, src, Rc

IR S,NS



11 + 10n*

*n is the number of iterations.

Operation

dst<0:7> ← src<0:7>
Rs<0:15> ← Rs<0:15>- 1
Rc<0:15> ← Rc<0:15>- 1
repeat until termination

Description

A Data byte in memory, addressed by the contents of the general purpose register designated by the Rs field of the instruction, is loaded into the destination port. The destination is addressed by the contents of the general purpose register designated by the Rd field of the instruction. The source contents are unaltered. The contents of the general purpose register designated by Rs are then decremented by 1. The contents of the general purpose register designated by the Rc field is decremented by 1. The instruction is terminated when the result of this decrementation reaches zero. This instruction is interruptible.

This instruction is similar in operation to the corresponding standard I/O instruction. The significant difference is that the data byte is transferred on the most significant eight bus lines. For standard I/O instructions transfers take place on the least significant eight lines.

			PV			AFFECTED
C	Z	S		DA	H	UNAFFECTED

Flags;

P/V: Set to 1.

SPECIAL OUTPUT byte from memory to I/O port, autoincrement and repeat.

SOTIRB

This is a system instruction.

SOTIRB dst, src, Rc

IR S,NS

0 0 1 1 1 0 1 0	Rs	0 0 1 1
0 0 0 0 Rc	Rc	0 0 0 0

11 + 10n*

*n is the number of iterations.

Operation

dst<0:7> ← src<0:7>

Rs<0:15> ← Rs<0:15>+ 1

Rc<0:15> ← Rc<0:15>- 1

Description

A data byte in the memory, addressed by the contents of the general purpose register designated by the Rs field of the instruction, is loaded into the destination port. The destination is addressed by the contents of the general purpose register designated by the Rd field of the instruction. The source contents are unaltered. The contents of the general purpose register designated by Rs are then incremented by 1. The contents of the general purpose register designated by Rc are decremented by 1. This instruction terminates when the result of this decrementation reaches zero. This instruction is interruptible.

This instruction is similar in operation to the corresponding standard I/O instruction. The significant difference is that the data byte is transferred on the most significant eight bus lines. For standard I/O instructions transfers take place on the least significant eight lines.

Flags:

			PV			AFFECTED
C	Z	S		DA	H	UNAFFECTED

P/V: Set to 1.

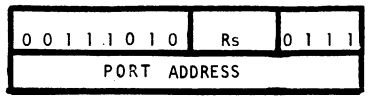
SPECIAL OUTPUT byte from register to I/O port

SOUTB

This is a system instruction

SOUTB dst, Rs

DA S,NS



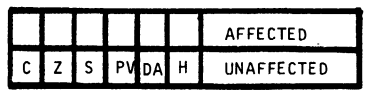
12

Operation

dst<0:7> ← Rs<0:7>

Description

The contents of the general purpose byte source register designated by the Rs field of the instruction are loaded into an output port. The port address is determined directly from the instruction. The source contents are unaltered. The instruction is similar in operation to the corresponding standard I/O instruction. The significant difference is that the data byte is transferred on the most significant eight bus lines, for standard I/O instructions transfers take place on the least significant eight lines.



Flags are not affected.

SPECIAL OUTPUT byte from memory to I/O port, autodecrement.

SOUTDB

This is a system instruction.

CLOCK CYCLES

SOUTDB dst, src, Rc

IR S,NS

0 0 1 1 1 0 1 0	Rs	1 0 1 1
0 0 0 0	Rc	Rd	1 0 0 0

21

Operation

dst<0:7> ← src<0:7>

Rs<0:15> ← Rs<0:15>- 1

Rc<0:15> ← Rc<0:15>- 1

Description

Data byte in memory, addressed by the contents of the general purpose register designated by the Rs field of the instruction, is loaded into the destination port. The destination is addressed by the contents of the general purpose register designated by the Rd field of the instruction. The source contents are unaltered. The contents of the general purpose register designated by Rs are then decremented by 1. The contents of the general purpose register designated by Rc are decremented by 1.

This instruction is similar in operation to the corresponding standard I/O instruction. The significant difference is that the data byte is transferred on the most significant eight bus lines. For standard I/O instructions transfers take place on the least significant eight lines.

			PV			AFFECTED
C	Z	S		DA	H	UNAFFECTED

Flags:

P/V: Set to 1 if result of decrementing Rc is zero. Reset otherwise.

SPECIAL OUTPUT

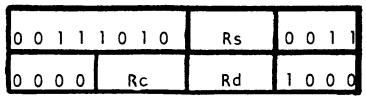
byte from memory to I/O port, autoincrement.

SOUTIB

This is a system instruction.

SOUTIB dst, src, Rc

IR S,NS



21

Operation

dst<0:7> ← src<0:7>

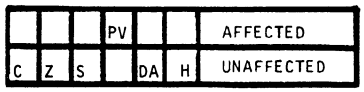
Rs<0:15> ← Rs<0:15>+ 1

Rc<0:15> ← Rc<0:15>- 1

Description

Data byte in memory, addressed by the contents of the general purpose register designated by the Rs field of the instruction, is loaded into the destination port. The destination is addressed by the contents of the general purpose register designated by the Rd field of the instruction. The source contents are unaltered. The contents of the general purpose register designated by Rs are then incremented by 1. The contents of the general purpose register designated by Rc are decremented by 1.

This instruction is similar in operation to the corresponding standard I/O instruction. The significant difference is that the data byte is transferred on the most significant eight bus lines. For standard I/O instructions transfers take place on the least significant eight lines.



Flags:

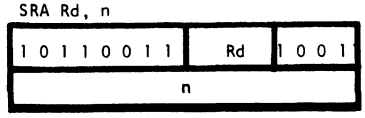
P/V: Set to 1 if result of decrementing Rc is zero. Reset otherwise.

SHIFT word arithmetic right

SRA

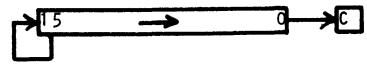
R S,NS

SRA Rd, n



13 + 3n*

Operation



*n is the number of places shifted.

Description

The contents of the word destination register are shifted right. The destination is a general purpose word register designated by the Rd field of the instruction. The number of places to be shifted is determined from the value of the n field of the instruction. The magnitude of the shift must be in the range 0 to 16. The n field is a 16 bit negative integer in 2's complement notation.

This operation is identical to the operation SRL apart from the treatment of the most significant bit of the word, bit 15. This bit is unaltered during the shift operation, and shifts into the adjacent bit, bit 14. Thus a signed operand has the sign preserved during the shifting operation.

C	Z	S	PV			AFFECTED
				DA	H	UNAFFECTED

Flags:

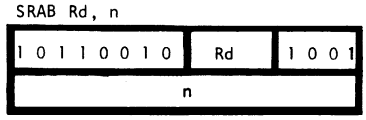
C: Loaded from the least significant bit shifted out of the register.
Z: Set to 1 if result is zero. Reset otherwise.
S: Set if the most significant bit of the resultant destination is 1. Reset otherwise.
P/V: Reset.

SHIFT byte arithmetic right

SRAB

SRAB Rd, n

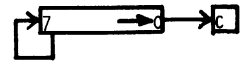
R S,NS



12 + 3n*

*n is the number of places shifted.

Operation



Description

The contents of the byte destination register are shifted right. The destination is a general purpose byte register designated by the Rd field of the instruction. The number of places to be shifted is determined from the value of the n field of the instruction. The magnitude of the shift must be in the range 0 to 8. The n field is a 16 bit negative integer in 2's complement notation.

This operation is identical to the operation SRLB apart from the treatment of the most significant bit of the byte, bit 7. This bit is unaltered during the shift operation, and shifts into the adjacent bit, bit 6. Thus a signed operand has its sign preserved during the shifting operation.

C	Z	S	PV			AFFECTED
				DA	H	UNAFFECTED

Flags:

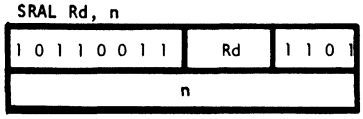
C: Loaded from the least significant bit shifted out of the register.
Z: Set to 1 if result is zero. Reset otherwise.
S: Set if the most significant bit of the resultant destination is 1. Reset otherwise.
P/V: Reset.

SHIFT long word right arithmetic

SRAL

R

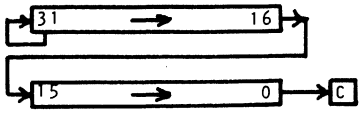
SRAL Rd, n



12 + 3n*

*n is the number of places shifted.

Operation



Description

The contents of the long word destination register are shifted right. The destination is a general purpose long word register designated by the Rd field of the instruction. The number of places to be shifted is determined from the value of the n field of the instruction. The magnitude of the shift must be in the range 0 to 32. The n field is a 16 bit negative integer in 2's complement notation.

This operation is identical to the operation SRLL apart from the treatment of the most significant bit of the long word, bit 31. This bit is unaltered during the shift operation, and shifts into the adjacent bit, bit 30. Thus a signed operand has its sign preserved during the shifting operation.

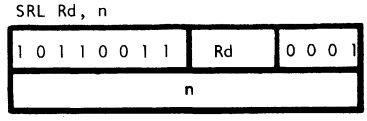
C	Z	S	PV			AFFECTED
				DA	H	UNAFFECTED

Flags:

C: Loaded from the least significant bit shifted out of the register.
Z: Set to 1 if result is Ø. Reset otherwise.
S: Set if the most significant bit of the resultant destination is 1. Reset otherwise.
P/V: Reset.

SRL Rd, n

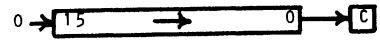
R S,NS



13 + 3n*

*n is the number of places shifted.

Operation



Description

The contents of the word destination register are shifted right.
The destination is a general purpose word register designated by the Rd field of the instruction. The number of places to be shifted is determined from the value of the n field of the instruction. The magnitude of the shift must be in the range 0 to 16. The n field is a 16 bit negative integer in 2's complement notation.

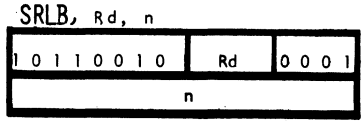
C	Z	S				AFFECTED
			PV	DA	H	UNAFFECTED

Flags:

C: Loaded from the least significant bit shifted out of the register.
Z: Set to 1 if result is zero. Reset otherwise.
S: Set if the most significant bit of the resultant destination is 1. Reset otherwise.
P/V: Undefined.

SRLB, Rd, n

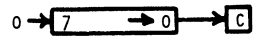
R



13 + 3n*

*n is the number of places shifted.

Operation



Description

The contents of the byte destination register are shifted right. The destination is a general purpose byte register designated by the Rd field of the instruction. The number of places to be shifted is determined from the value of the n field of the instruction. The magnitude of the shift must be in the range 0 to 8. The n field is a 16 bit negative integer in 2's complement notation.

C	Z	S				AFFECTED
			PV	DA	H	UNAFFECTED

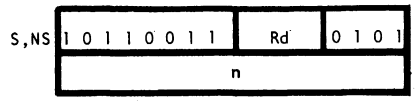
Flags:

C: Loaded from the least significant bit shifted out of the register.
Z: Set to 1 if result is zero. Reset otherwise.
S: Set if the most significant bit of the resultant destination is 1. Reset otherwise.
P/V: Undefined.

SHIFT long word logical right (static) **SRLL**

R

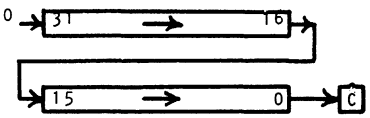
SRLL Rd, n



13 + 3n*

*n is the number of places shifted

Operation



Description

The contents of the long word destination register are shifted right. The destination is a general purpose register pair designated by the Rd field of the instruction. The number of places to be shifted is determined from the value of the n field of the instruction. The magnitude of the shift must be in the range 0 to 32. The n field is a 16 bit negative integer in 2's complement notation.

C	Z	S				AFFECTED
			PV	DA	H	UNAFFECTED

Flags:

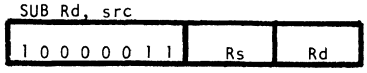
C: loaded from the least significant bit shifted out of the register.
Z: Set to 1 if result is zero. Reset otherwise.
S: Set if the most significant bit of the resultant destination is 1. Reset otherwise.
P/V: Undefined.

SUBTRACT word from register

SUB

SUB Rd, src

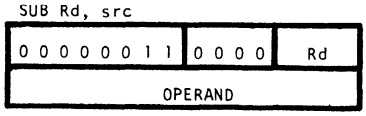
R S,NS



4

SUB Rd, src

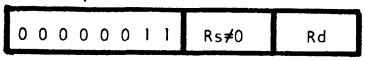
IM S,NS



7

SUB Rd, src

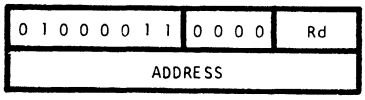
IR S,NS



7

SUB Rd, src

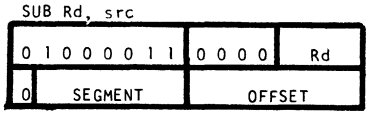
DA NS



9

SUB Rd, src

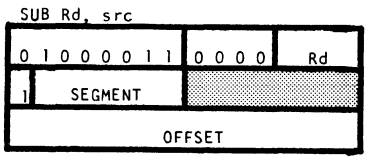
DA SSO



10

SUB Rd, src

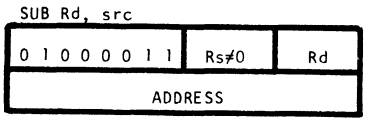
DA SLO



12

SUB Rd, src

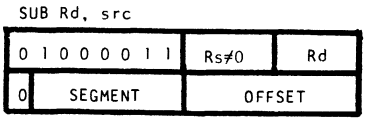
X NS



10

SUB Rd, src

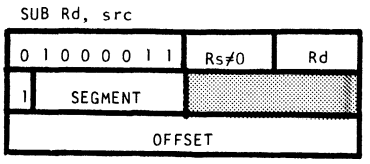
X SSO



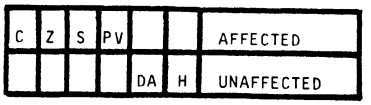
10

SUB Rd, src

X SLO



13



Operation

dst<0:15> ←dst<0:15> - src<0:15>

Description

The source word operand contents are subtracted from the contents of the general purpose word register designated by the Rd field of the instruction. The result is loaded into the destination. The source operand is obtained using the applicable addressing mode. The original contents of the destination register are lost while those of the source operand are unaltered.

Flags:

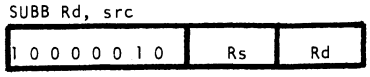
C: Reset on carry from most significant bit of result. Set to 1 otherwise (i.e., borrow).
Z: Set to 1 if result is zero. Reset otherwise.
S: Set to 1 if result is negative. Reset otherwise.
P/V: Set to 1 on arithmetic overflow. Reset otherwise.

SUBTRACT byte from register

SUBB

R S,NS

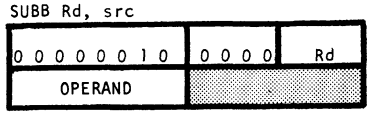
SUBB Rd, src



6

IM S,NS

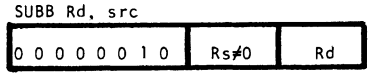
SUBB Rd, src



7

IR S,NS

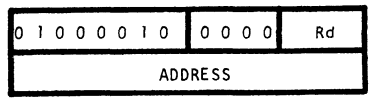
SUBB Rd, src



7

DA NS

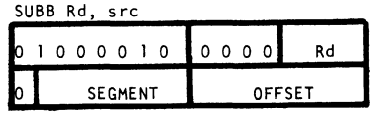
SUBB Rd, src



9

DA SSO

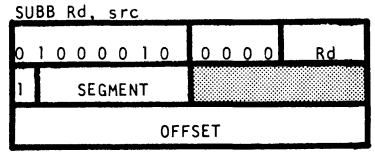
SUBB Rd, src



10

DA SLO

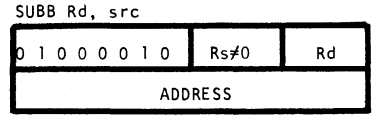
SUBB Rd, src



12

X NS

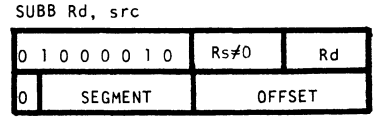
SUBB Rd, src



10

X SSO

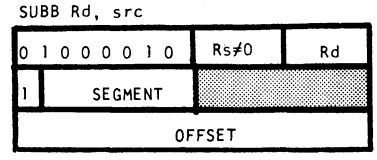
SUBB Rd, src



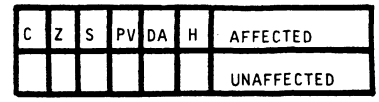
10

X SLO

SUBB Rd, src



13



Operation

dst<0:7> ←dst<0:7> - src<0:7>

Description

The source byte operand contents are subtracted from the contents of the general purpose byte register designated by the Rd field of the instruction. The result is loaded into the destination. The source operand is obtained using the applicable addressing mode. The original contents of the destination register are lost and those of the source operand are unaltered.

Flags:

C: Reset on carry from most significant bit of result. Set to 1 otherwise (i.e., borrow).
Z: Set to 1 if result is zero. Reset otherwise.
S: Set to 1 if result is negative. Reset otherwise.
P/V: Set to 1 on arithmetic overflow. Reset otherwise.
DA: Set to 1 always.
H: Reset on carry from most significant bit of lower 4 bits of result. Set otherwise (i.e., borrow).

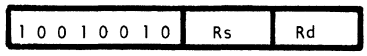
SUBTRACT long word from register

SUBL

R

S,NS

SUBL Rd, src

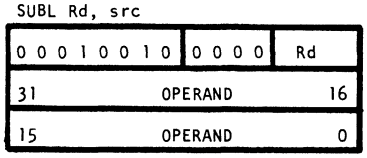


8

IM

S,NS

SUBL Rd, src

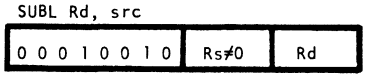


14

IR

S,NS

SUBL Rd, src

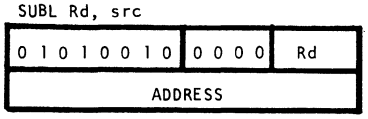


14

DA

NS

SUBL Rd, src

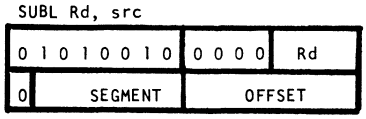


15

DA

SSO

SUBL Rd, src

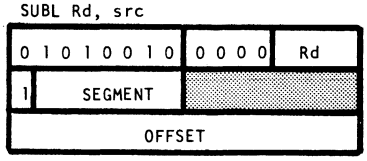


16

DA

SLO

SUBL Rd, src

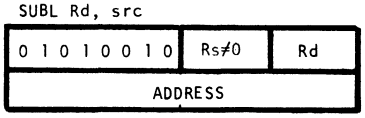


18

X

NS

SUBL Rd, src

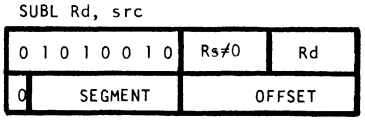


16

X

SSO

SUBL Rd, src

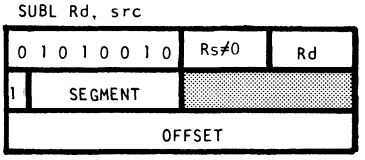


16

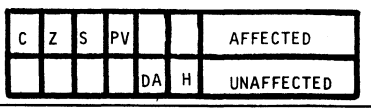
X

SLO

SUBL Rd, src



19



Operation

dst<0:31> ←dst<0:31>- src<0:31>

Description

The source long word operand contents are subtracted from the contents of the general purpose register pair designated by the Rd field of the instruction. The result is loaded into the destination. The source operand is obtained using the applicable addressing mode. The original contents of the destination register are lost while those of the source operand are unaltered.

Flags:

C: Reset on carry from most significant bit of result. Set to 1 otherwise (i.e., borrow).
Z: Set to 1 if result is Ø. Reset otherwise.
S: Set to 1 if result is negative. Reset otherwise.
P/V: Set to 1 on arithmetic overflow. Reset otherwise.

TCC Rd, CC

R S,NS

5

Operation

dst<Ø> ← 1 if condition is met.

Description

The contents of the flags are compared with those specified by the CC field of the instruction. If the comparison is successful, the least significant bit of the destination word is set to 1. Otherwise this bit is unaffected. Remaining bits of the destination are not altered.

						AFFECTED
C	Z	S	P/V	DA	H	UNAFFECTED

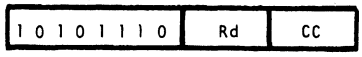
Flags are not affected.

TEST condition codes,
and set a bit in byte

TCCB

TCCB Rd, CC

R S,NS



5

Operation

Dst <Ø> ← 1 if condition is met.

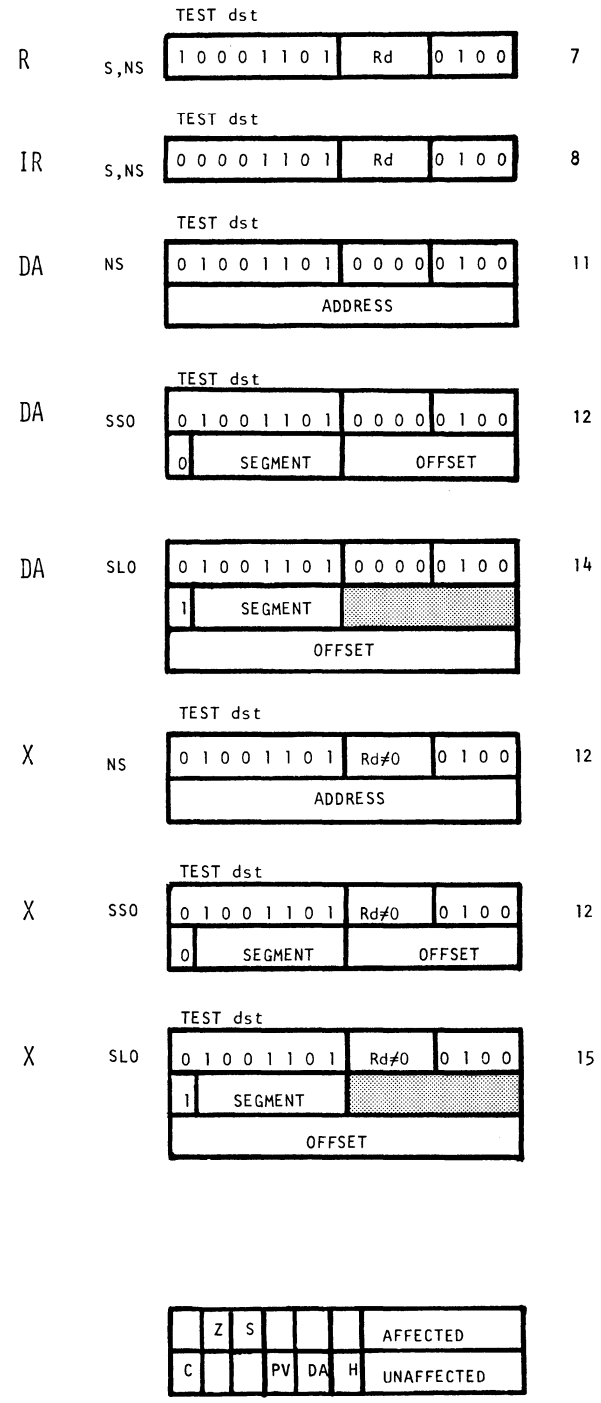
Description

The contents of the flags are compared with those specified by the CC field of the instruction. If the comparison is successful, the least significant bit of the destination byte is set to 1. Otherwise, unaffected. Remaining bits of the destination are not altered.

						AFFECTED
C	Z	S	Pv	DA	H	UNAFFECTED

Flags are not affected.

TEST



Operation

dst<0:15> ← dst<0:15> V Ø

Description

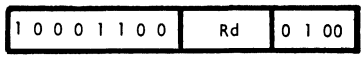
The contents of the destination word operand are tested to set the appropriate flags. Testing is done by performing a logical OR operation between destination word and zero. The destination is determined by the applicable addressing mode and the contents of the destination are not altered.

Flags:

Z: Set to 1 if result is zero. Reset otherwise.
S: Set to 1 if result is negative. Reset otherwise.

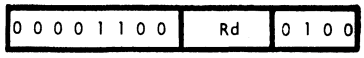
TEST byte — TESTB

R S,NS — TESTB dst



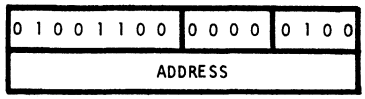
7

IR S,NS — TESTB dst



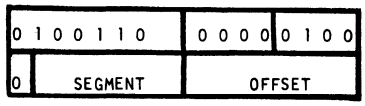
8

DA NS — TESTB dst



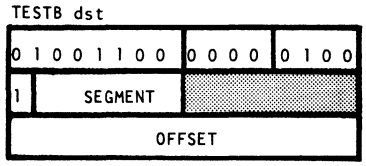
11

DA SSO — TESTB dst



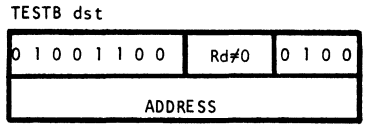
12

DA SLO — TESTB dst



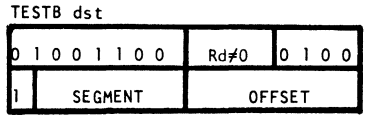
14

X NS — TESTB dst



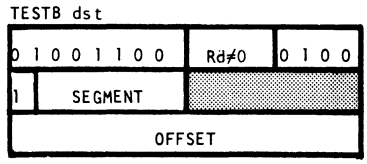
12

X SSO — TESTB dst

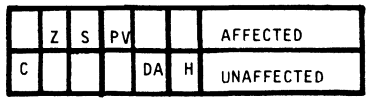


12

X SLO — TESTB dst



15



Operation

dst<0:7> ← dst<0:7> ∨ Ø

Description

The contents of the destination byte operand destination are tested to set the appropriate flags. Testing is done by performing a logical OR operation between destination byte and zero. The destination is determined by the applicable addressing mode and the contents of the destination are not altered.

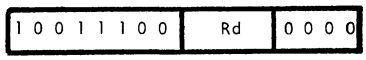
Flags:

Z: Set to 1 if operand is zero. Reset otherwise.
S: Set to 1 if operand is negative. Reset otherwise.
P/V: Set to 1 if parity of operand is even. Reset otherwise.

TESTL

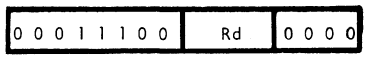
R S,NS — 13

TESTL dst



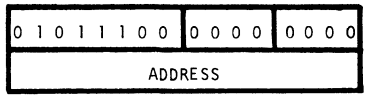
IR S,NS — 13

TESTL dst



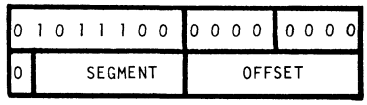
DA NS — 16

TESTL dst



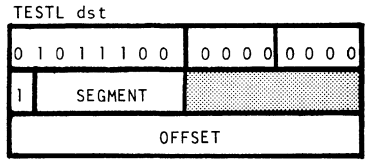
DA SSO — 17

TESTL dst



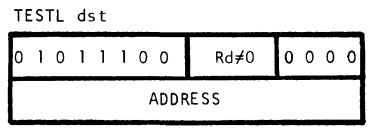
DA SLO — 19

TESTL dst



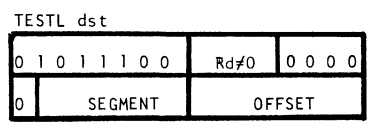
X NS — 17

TESTL dst



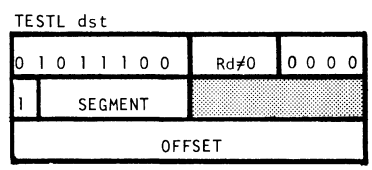
X SSO — 17

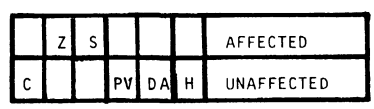
TESTL dst



X SLO — 20

TESTL dst





Operation

dst<0:31> ←dst<0:31> V Ø

Description

The contents of the long word destination are tested to set appropriate flags. Testing is done by performing a logical OR operation between destination and zero. The destination is determined by the applicable addressing mode and the contents of the destination are not altered.

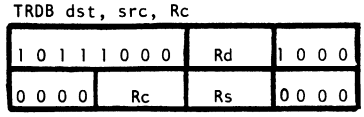
Flags:

Z: Set to 1 if result is zero.
Reset otherwise.
S: Set to 1 if result is negative.
Reset otherwise.

TRANSLATE byte, autodecrement **TRDB**

TRDB dst, src, Rc

IR S,NS



25 Operation

Description

The general purpose register (register pair in AmZ8001) designated by the Rs field of the instruction contains the starting address of a byte table.

The general purpose register (register pair in AmZ8001) designated by the Rd field of the instruction contains the address of a byte string to be translated.

The general purpose register designated by the Rc field of the instruction contains the length (in bytes) of the string remaining to be translated.

A byte is read from the address specified by the Rd register. This byte is used as a table index and is added to the starting address of the table. The translated byte is read from this address, and is loaded into the address specified by the Rd register.

The address specified by the Rd register is decremented by 1 to point to the next byte of the string. The contents of the register designated by the Rc field of the instruction are decremented by 1, to indicate the remaining length of the string to be translated. This completes one iteration.

This instruction terminates after 1 iteration. It is a special case of the instruction TRDRB.

			PV			AFFECTED
C	Z	S		DA	H	UNAFFECTED

Flags:

Z: Undefined.
P/V: Set to 1 if result of decrementing Rc is zero. Reset otherwise.

TRANSLATE byte, autodecrement and repeat.

TRDRB

TRDRB dst, src, Rc

IR S,NS

1 0 1 1 1 0 0 0	Rd	1 1 0 0	11 + 14n*
0 0 0 0 Rc	Rs	0 0 0 0	

*n is the number of iterations

Operation

Description

The general purpose register (register pair in AmZ8001) designated by the Rs field of the instruction contains the starting address of a byte table.

The general purpose register (register pair in AmZ8001) designated by the Rd field of the instruction contains the address of a byte string to be translated.

The general purpose register designated by the Rc field of the instruction contains the length (in bytes) of the string remaining to be translated.

A byte is read from the address specified by the Rd register. This byte is used as a table index and is added to the starting address of the table. The translated byte is read from this address and is loaded into the address specified by the Rd register.

The address specified by the Rd register is decremented by 1 to point to the next byte of the string. The contents of the register designated by the Rc field of the instruction are decremented by 1, to indicate the remaining length of the string to be translated. This completes one iteration.

The instruction repeats until the contents of the Rc register reach zero, indicating that the string has been exhausted. This instruction is interruptible at the end of each iteration.

			PV			AFFECTED
C	Z	S		DA	H	UNAFFECTED

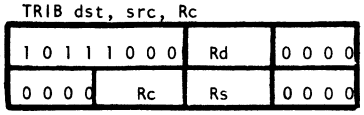
Z: Undefined.
P/V: Set to 1.

TRANSLATE byte, autoincrement

TRIB

TRIB dst, src, Rc

IR



25 Operation

Description

The general purpose register (register pair in AmZ8001) designated by the Rs field of the instruction contains the starting address of a byte table.

The general purpose register (register pair in AmZ8001) designated by the Rd field of the instruction contains the address of a byte string to be translated.

The general purpose register designated by the Rc field of the instruction contains the length (in bytes) of the string remaining to be translated.

A byte is read from the address specified by the Rd register. This byte is used as a table index and is added to the starting address of the table. The translated byte is read from this address, and is loaded into the address specified by the Rd register.

The address specified by the Rd register is incremented by 1 to point to the next byte of the string. The contents of the register designated by the Rc field of the instruction are decremented by 1, to indicate the remaining length of the string to be translated. This completes one iteration.

This instruction terminates after 1 iteration. It is a special case of the instruction TRIRB.

			PV			AFFECTED
C	Z	S		DA	H	UNAFFECTED

Flags:

Z: Undefined.
P/V: Set to 1 if result of decrementing Rc is zero. Reset otherwise.

TRANSLATE byte string, autoincrement and repeat

TRIRB

TRIRB dst, src, Rc

IR S,NS

1 0 1 1 1 0 0 0	Rd	0 1 0 0
0 0 0 0 Rc	Rs	0 0 0 0

11 + 14n*

Operation

*n is the number of iterations.

Description

The general purpose register (register pair in AmZ8001) designated by the Rs field of the instruction contains the starting address of a byte table.

The general purpose register (register pair in AmZ8001) designated by the Rd field of the instruction contains the address of a byte string to be translated.

The general purpose register designated by the Rc field of the instruction contains the length (in bytes) of the string remaining to be translated.

A byte is read from the address specified by the Rd register. This byte is used as a table index and is added to the starting address of the table. The translated byte is read from this address, and is loaded into the address specified by the Rd register.

The address specified by the Rd register is incremented by 1 to point to the next byte of the string. The contents of the register designated by the Rc field of the instruction are decremented by 1, to indicate the remaining length of the string to be translated. This completes one iteration.

The instruction repeats until the contents of the Rc register reach zero, indicating that the string has been exhausted. This instruction is interruptible at the end of each iteration.

			PV			AFFECTED
C	Z	S		DA	H	UNAFFECTED

Flags:

Z: Undefined.
P/V: Set to 1.

TRANSLATE AND TEST byte, autodecrement — TRTDB

TRTDB, dst, src, Rc

IR S,NS

1 0 1 1 1 0 0 0	Rd	1 0 1 0
0 0 0 0	Rc	Rs	0 0 0 0

25 Operation

Description

The general purpose register (register pair in AmZ8001) designated by the Rs field of the instruction contains the starting address of a byte table.

The general purpose register (register pair in AmZ8001) designated by the Rd field of the instruction contains the address of a byte string to be translated and tested.

The general purpose register designated by the Rc field of the instruction contains the length (in bytes) of the string remaining to be translated and tested.

A byte is read from the address specified by the Rd register. This byte is used as a table index and is added to the starting address of the table. The translated byte is read from this address and is loaded into the general purpose byte register RØ for testing.

The address specified by the Rd register is decremented by 1 to point to the next byte of the string. The contents of the register designated by the Rc field of the instruction are decremented by 1, to indicate the remaining length of the string to be translated and tested. This completes one iteration.

This instruction terminates after 1 iteration. It is a special case of the instruction TRTDRB.

	Z		PV			AFFECTED
C		S		DA	H	UNAFFECTED

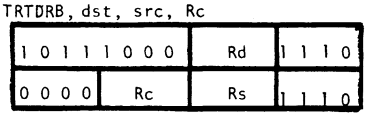
Flags:

Z: Set to 1 if the translated byte is zero. Reset otherwise.
P/V: Set to 1 if result of decrementing Rc is zero. Reset otherwise.

TRANSLATE AND TEST byte, autodecrement and repeat. TRTDRB

TRTDRB, dst, src, Rc

IR S,NS



11 + 14n* Operation

*n is the number of iterations.

Description

The general purpose register (register pair in AmZ8001) designated by the Rs field of the instruction contains the starting address of a byte table.

The general purpose register (register pair in AmZ8001) designated by the Rd field of the instruction contains the address of a byte string to be translated and tested.

The general purpose register designated by the Rc field of the instruction contains the length (in bytes) of the string remaining to be translated and tested.

A byte is read from the address specified by the Rd register. This byte is used as a table index and is added to the starting address of the table. The translated byte is read from this address and is loaded into the general purpose byte register RØ for testing.

The address specified by the Rd register is decremented by 1 to point to the next byte of the string. The contents of the register designated by the Rc field of the instruction are decremented by 1, to indicate the remaining length of the string to be translated and tested. This completes one iteration.

The instruction repeats until the value loaded into the RØ register is non zero or until the contents of the Rc register reach zero, indicating that the string has been exhausted. This instruction is interruptible at the end of each iteration.

	Z		PV			AFFECTED
C		S		DA	H	UNAFFECTED

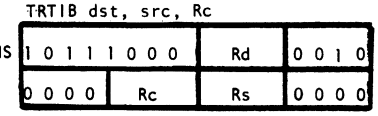
Flags:

Z: Set to 1 if the translated byte is zero. Reset otherwise.
P/V: Set to 1 if result of decrementing Rc is zero. Reset otherwise.

TRANSLATE AND TEST byte, autoincrement — TRTIB

TRTIB dst, src, Rc

IR S,NS



25

Operation

Description

The general purpose register (register pair in AmZ8001) designated by the Rs field of the instruction contains the starting address of a byte table.

The general purpose register (register pair in AmZ8001) designated by the Rd field of the instruction contains the address of a byte string to be translated and tested. The general purpose register designated by the Rc field of the instruction contains the length (in bytes) of the string remaining to be translated and tested.

A byte is read from the address specified by the Rd register. This byte is used as a table index and is added to the starting address of the table. The translated byte is read from this address, and is loaded into the general purpose byte register RØ for testing.

The address specified by the Rd register is incremented by 1 to point to the next byte of the string. The contents of the register designated by the Rc field of the instruction are decremented by 1, to indicate the remaining length of the string to be translated and tested. This completes one iteration.

This instruction terminates after 1 iteration. It is a special case of the instruction TRTIRB.

	Z		PV			AFFECTED
C		S		DA	H	UNAFFECTED

Flags:

Z: Set to 1 if the translated byte is zero. Reset otherwise.
P/V: Set to 1 if result of decrementing Rc is zero. Reset otherwise.

TRANSLATE AND TEST byte string, autoincrement, and repeat

TRTIRB

TRTIRB dst, src, Rc,

IR S,NS

1 0 1 1 1 0 0 0	Rd	0 1 1 0
0 0 0 0 Rc	Rs	1 1 1 0

Operation: 11 + 14n*

*n is the number of translate iterations.

Description

The general purpose register (register pair in AmZ8001) designated by the Rs field of the instruction contains the starting address of a byte table.

The general purpose register (register pair in AmZ8001) designated by the Rd field of the instruction contains the address of a byte string to be translated and tested.

The general purpose register designated by the Rc field of the instruction contains the length (in bytes) of the string remaining to be translated and tested.

A byte is read from the address specified by the Rd register. This byte is used as a table index and is added to the starting address of the table. The translated byte is read from this address, and is loaded into the general purpose byte register RØ for testing.

The address specified by the Rd field is incremented by 1 to point to the next byte of the string. The contents of the register designated by the Rc field of the instruction are decremented by 1, to indicate the remaining length of the string to be translated and tested. This completes one iteration.

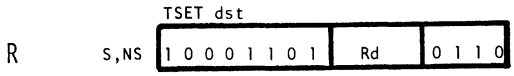
The instruction repeats until the value loaded into the RØ register is non zero, or until the contents of the Rc register reach zero, indicating that the string has been exhausted. This instruction is interruptible at the end of each iteration.

	Z		PV			AFFECTED
C		S		DA	H	UNAFFECTED

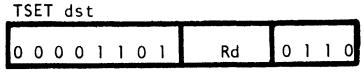
Flags:

Z: Set to 1 if the table entry is zero. Reset otherwise.
P/V: Set to 1 if result of decrementing Rc is zero. Reset otherwise.

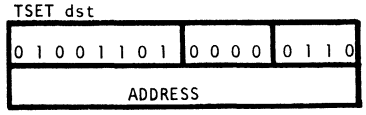
TEST word and set — TSET



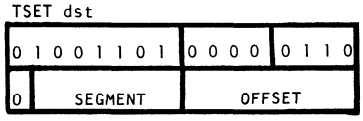
R S,NS 7



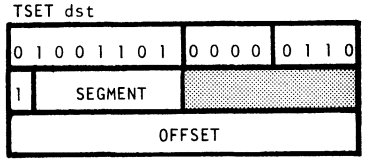
IR S,NS 11



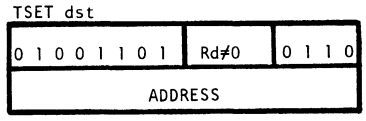
DA NS 14



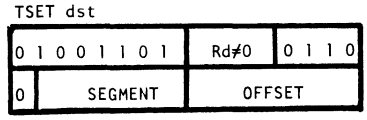
DA SSO 15



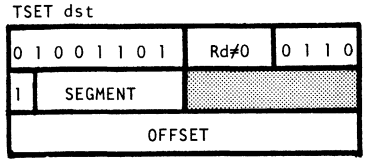
DA SLO 17



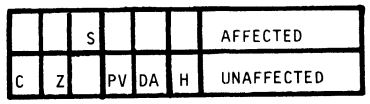
X NS 15



X SSO 15



X SLO 18



Operation

If dst<0:15> ← is negative
then S Flag ← 1
otherwise S flag ← 0
dst<0:15> ← FFFF

Description

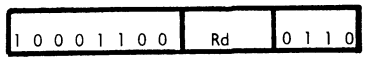
The most significant (sign) bit of the destination word is loaded into the S flag. The contents of the destination are then set to all 1's. The destination is determined by the applicable addressing mode.

Flags:

S: Set to 1 if the most significant bit of the destination is 1. Reset otherwise.

R S,NS

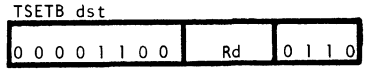
TSETB dst



7

IR S,NS

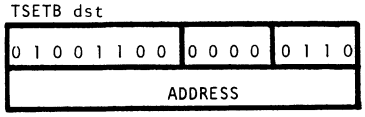
TSETB dst



11

DA NS

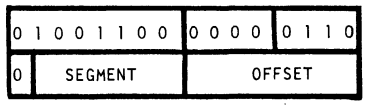
TSETB dst



14

DA SSO

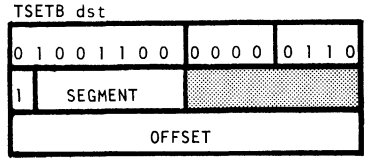
TSETB dst



15

DA SLO

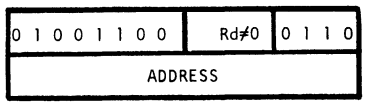
TSETB dst



17

X NS

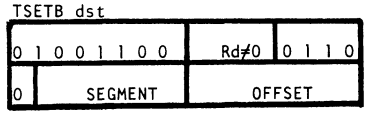
TSETB dst



15

X SSO

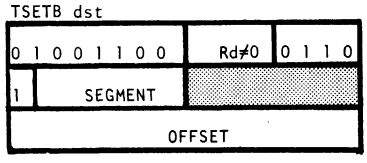
TSETB dst



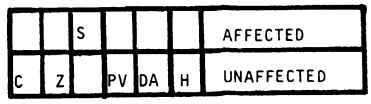
15

X SLO

TSETB dst



18



Operation

If dst<0:7> is negative
then S Flag ← 1
otherwise ← 0
dst<0:7> ← FF

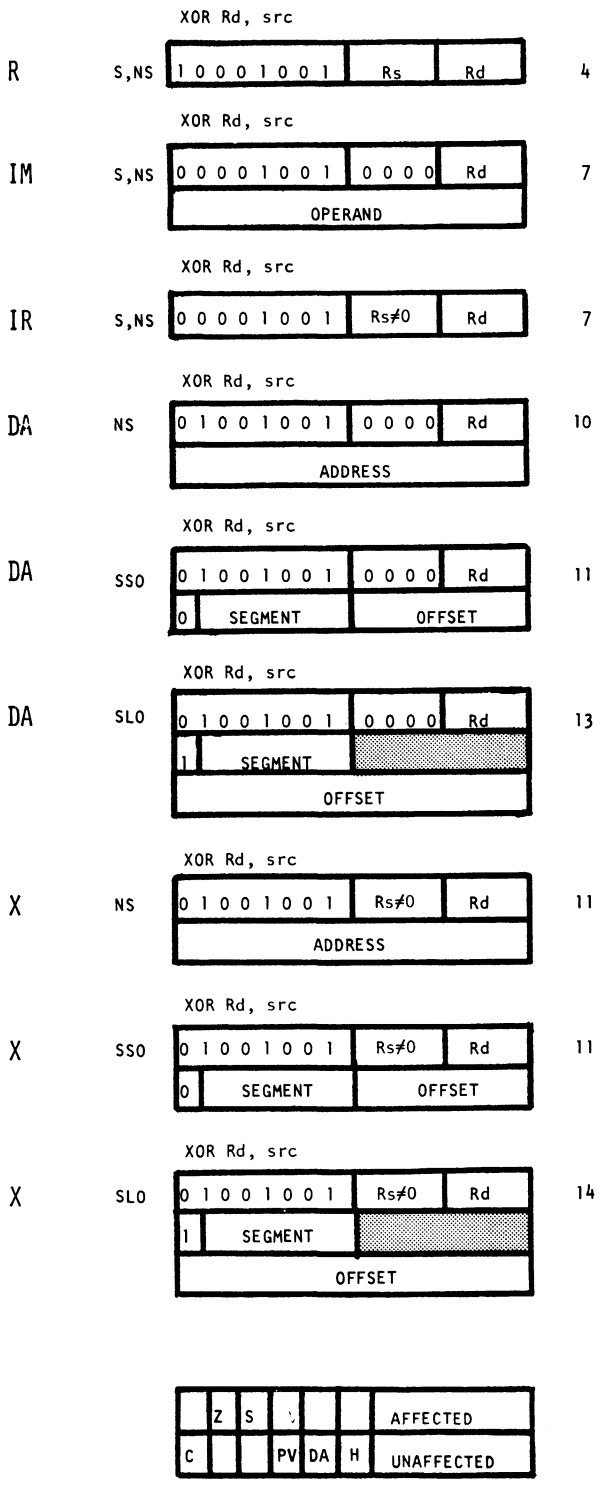
Description

The most significant (sign) bit of the destination byte is loaded into the S flag. The contents of the destination are then set to all 1's. The destination is determined by the applicable addressing mode.

Flags:

S: Set to 1 if the most significant bit of the destination is 1. Reset otherwise.

EXCLUSIVE OR word with register. XOR



Operation

dst <0:15> ← src <0:15>⊕dst<0:15>

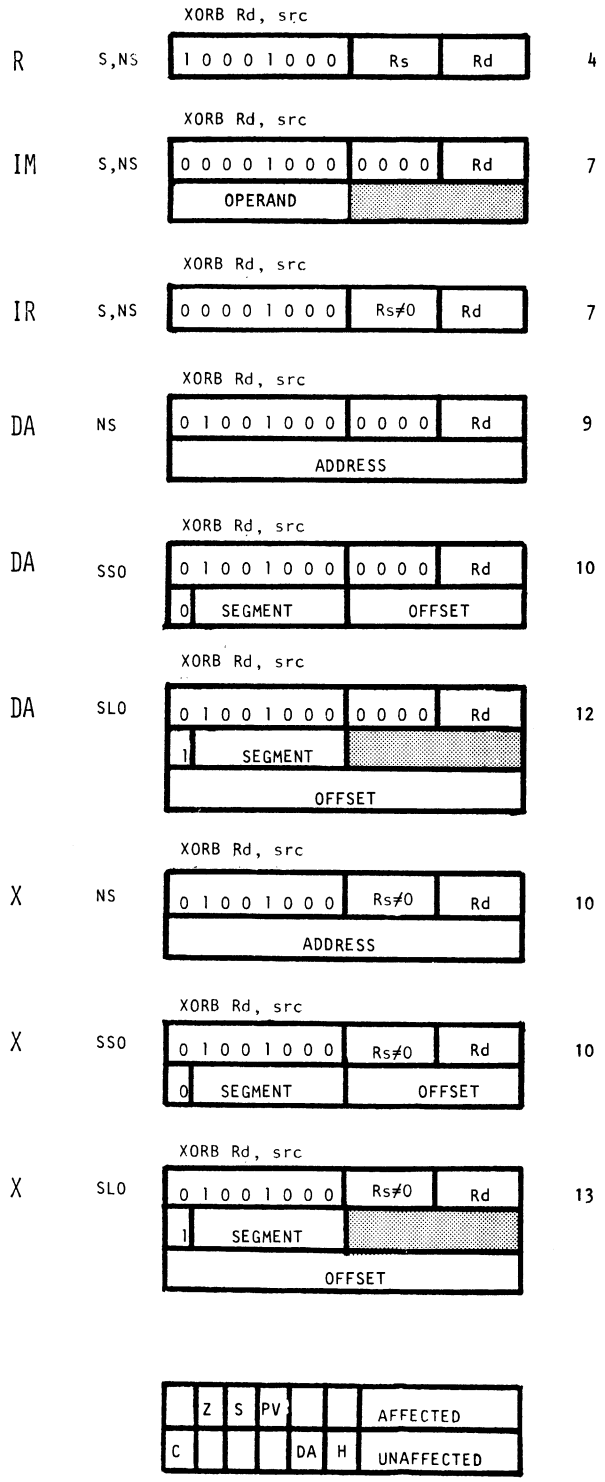
Description

A logical EXCLUSIVE OR operation is performed between corresponding bits of the source and destination words. The source operand is obtained by the appropriate addressing mode, and the destination operand is always a general purpose word register designated by the Rd field of the instruction. The 16-bit result is loaded into the destination, whose original contents are lost. The contents of the source are not altered.

Flags:

Z: Set to 1 if result is zero. Reset otherwise.
S: Set to 1 if result is negative. Reset otherwise.

EXCLUSIVE OR byte with register. XORB



Operation

dst<0:7> ←src<0:7>⊕dst<0:7>

Description

A logical EXCLUSIVE OR operation is performed between corresponding bits of the source and destination bytes. The source operand is obtained by the appropriate addressing mode, and the destination operand is always a general purpose byte register designated by the Rd field of the instruction. The 8-bit result is loaded into the destination, whose original contents are lost. The contents of the source are not altered.

Flags:

Z: Set to 1 if result is zero. Reset otherwise.
S: Set to 1 if result is negative. Reset otherwise.
P/V: Set to 1 if parity of result is even. Reset otherwise.

INSTRUCTIONS ARRANGED BY MNEMONIC

MNEMONIC	OPERANDS FOR THE GROUP	NAME	ADDRESSING MODES FOR THE GROUP	PAGE
ADC	Rd,Rs	ADD words with carry	R	48
ADCB		ADD bytes with carry		49
ADD	Rd,src	ADD word to register	R,IR,DA,X,IM	50
ADDB		ADD byte to register		51
ADDL		ADD long word to register		52
AND	Rd,src	AND word with register	R,IR,DA,X,IM	53
ANDB		AND byte with register		54
BIT	dst,Rs	BIT test in a word (dynamic	R	55
BITB		BIT test in a byte (dynamic)		57
BIT		BIT test in a word (static)	R,IR,DA,X	56
BITB	dst	BIT test in a byte (static)		58
CALL	dst	CALL subroutine	IR,DA,X	59
CALR	d	CALL subroutine relative	RA	60
CLR	dst	CLEAR word	R,IR,DA,X	61
CLRB		CLEAR byte		62
CLRL		CLEAR long word		63
COM	dst	COMPLEMENT word	R,IR,DA,X	64
COMB		COMPLEMENT byte		65
COMFLG		COMPLEMENT flags		66
CP	Rd,src	COMPARE register with word	R,IM,IR,DA,X	67
CPB		COMPARE register with byte		68
CPL		COMPARE register with long word		79
CP	IM,dst	COMPARE immediate word with memory	IR,DA,X	75
CPB		COMPARE immediate byte with memory		76
CPD	dst,src,Rc	COMPARE register to memory word, autodecrement	IR	69
CPDB		COMPARE register to memory byte, autodecrement	IR	70
CPDR	dst,src,Rc,CC	COMPARE register to memory word, autodecrement and repeat	IR	71
CPDRB		COMPARE register to memory byte, autodecrement and repeat		72
CPI	dst,src,Rc	COMPARE register to memory word, autoincrement	IR	73
CPIB		COMPARE register to memory byte, autoincrement	IR	74

INSTRUCTIONS ARRANGED BY MNEMONIC

MNEMONIC	OPERANDS FOR THE GROUP	NAME	ADDRESSING MODES FOR THE GROUP	PAGE
CPIR	dst,src,Rc,CC	COMPARE register to memory word, autoincrement and repeat	IR	77
CPIRB		COMPARE register to memory byte autoincrement and repeat	IR	78
CPSD	dst,src,Rc	COMPARE word stings in memory, autodecrement	IR	80
CPSDB		COMPARE byte strings in memory, autodecrement	IR	81
CPSDR	dst,src,Rc,CC	COMPARE word strings in memory, autodecrement and repeat	IR	82
CPSDRB		COMPARE byte strings in memory, autodecrement and repeat	IR	83
CPSI	dst,src,Rc	COMPARE byte strings in memory, autoincrement	IR	84
CPSIB		COMPARE byte strings in memory, autoincrement	IR	85
CPSIR	dst,src,Rc,CC	COMPARE word strings in memory, autoincrement and repeat	IR	86
CPSIRB		COMPARE byte strings in memory, autoincrement and repeat		87
DAB	Rd	DECIMAL adjust byte	R	88
DEC	dst,N	DECREMENT word	R,IR,DA,X	90
DECB		DECREMENT byte		91
DI	-	DISABLE Interrupt	-	92
DIV	dst,src	DIVIDE register pair by source word	R,IM,IR,DA,X	93
DIVL		DIVIDE register quadruple by source long word		94
DJNZ	Rc,d	DECREMENT word register & jump on non-zero	RA	95
DBJNZ		DECREMENT byte register & jump on non-zero	RA	89
EI	-	ENABLE Interrupt		96
EX	Rd,src	EXCHANGE source word with destination word	R,IR,DA,X	97
EXB		EXCHANGE source byte with destination byte		98
EXTS	Rd	EXTEND sign of word	R	99
EXTSB		EXTEND sign of byte		100
EXTSL		EXTEND sign of long word		101
HALT	-	HALT		102

INSTRUCTIONS ARRANGED BY MNEMONIC

MNEMONIC	OPERANDS FOR THE GROUP	NAME	ADDRESSING MODES FOR THE GROUP	PAGE
IN	Rd,src	INPUT word to register from I/O port	IR,DA	103
INB		INPUT byte to register from I/O port		104
INC	dst,N	INCREMENT word	R,IR,DA,X	105
INCB		INCREMENT byte		106
IND	dst,stc,Rc	INPUT word from I/O port to memory, autodecrement	IR	107
INDB		INPUT byte from I/O port to memory, autodecrement		108
INDR	dst,src,Rc	INPUT word from I/O port to memory, autodecrement and repeat	IR	109
INDRB		INPUT byte from I/O port to memory, autodecrement and repeat	IR	110
INI	dst,src,Rc	INPUT word from I/O port to memory, autoincrement	IR	111
INIB		INPUT byte from I/O port to memory, autoincrement	IR	112
INIR	dst,src,Rc	INPUT word from I/O port to memory, autoincrement and repeat	IR	113
INIRB	Rc	INPUT byte from I/O port to memory, autoincrement and repeat	IR	114
IRET	-	RETURN from interrupt	-	115
JP	CC,dst	JUMP conditional	IR,DA,X	116
JR	CC,d	JUMP conditional relative	RA	117
LD/LDR	dst,Rs	LOAD word register into memory	IR,DA,X,RA	118
LDB/LDRB		LOAD byte register into memory	,BA,BX	123
LDL/LDRL		LOADlong word register to memory		138
LD/LDR	Src,Rd	LOAD word into register	R,IM,IR	119
LDB/LDRB		LOAD byte into register	DA,X,RA	124
LDL/LDRL		LOAD long word into register	BA,BX	139
LD	Rd,IM	LOAD immediate word into memory	IR,DA,X	120
LDB		LOAD immediate byte into memory		125
LDL		LOAD immediate long word into memory		140
LDB/LDK	dst,IM	LOAD constant into register	R,IM	121
LDA/LDAR	Rd,d	LOAD address to register	RA,BA,BX,DA,X	122

MNEMONIC	OPERANDS FOR THE GROUP	NAME	ADDRESSING MODES FOR THE GROUP	PAGE
LDCTL	Rd,CW	LOAD control word into a register	R	126
LDCTL	Rs,CW	LOAD control word from register	R	127
LDCTLB	Rd	LOAD flag byte into register	R	128
LDCTLB	Rs	LOAD flag byte from register	R	129
LDD	dst,src	LOAD memory word to memory, autodecrement	IR	130
LDDB	Rc	LOAD memory byte to memory, autodecrement		131
LDDR	dst,src,Rc	LOAD memory word to memory, autodecrement and repeat	IR	132
LDDRB		LOAD memory byte to memory, autodecrement and repeat		133
LDI	dst,src,Rc	LOAD memory word to memory, autoincrement	IR	134
LDIB	Rc	LOAD memory byte to memory, autoincrement		135
LDIR	dst,src	LOAD memory word to memory, autoincrement and repeat	IR	136
LDIRB		LOAD memory byte to memory, autoincrement and repeat		137
LDM	Rd,src,N	LOAD multiple registers from memory	IR,DA,X	142
LDM		LOAD multiple registers into memory		141
LDPS	src	LOAD program status	IR,DA,X	143
MBIT	-	MULTI-MICRO test	-	144
MREQ	-	MULTI-MICRO request	-	145
MRES	-	MULTI-MICRO reset	-	146
MSET	-	MULTI-MICRO set	-	147
MULT	Rd,src	MULTIPLY register with word	R,IM,IR,DA	148
MULTL		MULTIPLY register with long word	X	149
NEG	dst	NEGATE word	R,IR DA,X	150
NEGB		NEGATE byte		151
NOP	-	NO Operation	-	152

MNEMONIC	OPERANDS FOR THE GROUP	NAME	ADDRESSING MODES FOR THE GROUP	PAGE
OR	Rd,src	OR word with register	R,IM,IR,DA,X	153
ORB		OR byte with register		154
OTDR	dst,src,Rc	OUTPUT word from memory to I/O port, autodecrement and repeat	IR	155
OTDRB		OUTPUT byte from memory to I/O port, autodecrement and repeat		156
OTIR	dst,src,Rc	OUTPUT word to I/O port from memory, autoincrement and repeat	IR	157
OTIRB	dst,src,Rc	OUTPUT byte to I/O port from memory, autoincrement and repeat		158
OUT	Rs,dst	OUTPUT word to I/O port from register	IR,DA	159
OUTB		OUTPUT byte to I/O port from register		160
OUTD	dst,src,Rc	OUTPUT word to I/O port from memory, autodecrement	IR	161
OUTDB	dst,src,Rc	OUTPUT byte to I/O port from memory, autodecrement		162
OUTI	dst,src Rc	OUTPUT word to I/O port from memory, autoincrement	IR	163
OUTIB	dst,src,Rc	OUTPUT byte to I/O port from memory, autoincrement		164
POP	dst src	POP word	R IR,DA X	165
POPL		POP long word		166
PUSH	dst,src	PUSH word	R,IM,IR,DA,X	167
PUSHL		PUSH long word		168
RES	dst,b	RESET bit in word (static)	R,IR,DA	169
RESB		RESET bit in byte (static)	X	171
RES	dst,Rs	RESET bit in word (dynamic)	R	170
RESB		RESET bit in byte (dynamic)		172
RESFLG	-	RESET flags	-	173
RET	CC	RETURN conditional	-	174

MNEMONIC	OPERANDS FOR THE GROUP	NAME	ADDRESSING MODES FOR THE GROUP	PAGE
RL	Rd,n	ROTATE word left	R	175
RLB		ROTATE byte left		176
RLC	Rd,n	ROTATE word left through carry	R	177
RLCB		ROTATE byte left through carry		178
RLDB	Rs,Rd	ROTATE digit left, byte	R	179
RRDB		ROTATE digit right, byte		184
RR	Rd,n	ROTATE word right	R	180
RRB		ROTATE BYTE right		181
RRC	Rd,n	ROTATE word right through carry	R	182
RRCB		ROTATE byte right through carry		183
SBC	Rs,Rd	SUBTRACT word with carry	R	185
SBCB		SUBTRACT byte with carry		186
SC	N	SYSTEM call	-	187
SDA		SHIFT word arithmetic (dynamic)		188
SDAB		SHIFT byte arithmetic (dynamic)		189
SDAL		SHIFT long word arithmetic (dynamic)		190
SDL	dst,Rs	SHIFT word logical (dynamic)	R	191
SDLB		SHIFT byte logical (dynamic)		192
SDLL		SHIFT long word logical (dynamic)		193
SET	dst,b	SET bit in word (static)	R,IR,DA,X	194
SETB		SET bit in byte (static)		196
SET	dst,Rs	SET bit in word (dynamic)	R	195
SETB		SET bit in byte (dynamic)		197
SETFLG	-	SET flags	-	198
SINB	Rd,src	SPECIAL input byte to register from I/O port	DA	199
SINDB	dst src, Rc	SPECIAL input byte from I/O port to memory, autodecrement	IR	200
SINDRB	dst,src,Rc	SPECIAL input byte from I/O port to memory, autodecrement and repeat	IR	201
SINIB	dst,src,Rc	SPECIAL input byte from I/O port to memory, autoincrement	IR	202
SINIRB	dst,src,Rc	SPECIAL input byte from I/O port to memory, autoincrement and repeat	IR	203

MNEMONIC	OPERANDS FOR THE GROUP	NAME	ADDRESSING MODES FOR THE GROUP	PAGE
SLA	Rd,n	SHIFT word arithmetic left (static)	R	204
SLAB		SHIFT byte arithmetic left (static)		205
SLAL		SHIFT long word arithmentic left (static)		206
SLL	Rd,n	SHIFT word logical left	R	207
SLLB		SHIFT byte logical left		208
SLLL		SHIGT long word logical left		209
SRA	Rd,n	SHIFT word arithmetic right (static)	R	215
SRAB		SHIFT byte arithmetic right (static)		216
SRAL		SHIFT long word right arithmetic (static)		217
SRL	Rd,n	SHIFT word logical right (static)	R	218
SRLB		SHIFT byte logical right (static)		219
SRLL		SHIFT long word logical right (static)		220
SOTDRB	dst,src,Rc	SPECIAL output byte from memory to I/O port, autodecrement and repeat	IR	210
SOTIRB	dst,src,Rc	SPECIAL output byte to I/O port, autoincrement and repeat	IR	211
SOUTB	dst,Rs	SPECIAL output byte from register to I/O port	DA	212
SOUTDB	dst,src,Rc	SPECIAL output byte from memory to I/O port, autodecrement	IR	213
SOUTIB	dst,src,Rc	SPECIAL output byte from memory to I/O port, autoincrement	IR	214
SUB	Rd,src	SUBTRACT word from register	R,IM,IR	221
SUBB		SUBTRACT byte from register	DA,X	222
SUBL		SUBTRACT long word from register		223
TCC	Rd,CC	TEST condition codes and set bit in word	R	224
TCCB		TEST condition codes and set bit in byte		225
TEST	dst	TEST word	R,IR,DA,X	226
TESTB		TEST byte		227
TESTL		TEST long word		228
TRDB	dst,src,Rc	TRANSLATE byte, autodecrement	IR	229
TRDRB	dst,src,Rc	TRANSLATE bute, autodecrement and repeat		230
TRIB	dst,src,Rc	TRANSLATE byte,autoincrement	IR	231

MNEMONIC	OPERANDS FOR THE GROUP	NAME	ADDRESSING MODES FOR THE GROUP	PAGE
TRIRB	dst,src,Rc	TRANSLATE byte, autoincrement and repeat	IR	232
TRTDB	dst,src,Rc	TRANSLATE & TEST byte,autodecrement	IR	233
TRTDRB	dst,src,Rc	TRANSLATE & TEST byte, autodecrement and repeat		234
TRTIB	dst,src,Rc	TRANSLATE & TEST byte, autoincrement	IR	235
TRTIRB	dst,src,Rc	TRANSLATE & TEST byte, autoincrement and repeat		236
TSET	dst	TEST word and set	R,IR,DA,X	237
TSETB	dst	TEST byte and set		238
XOR	Rd,src	EXCLUSIVE OR word with register	R,IM,DA,X,IR	239
XORB		EXCLUSIVE OR byte with register		240

Oct. 1979 ERRATA SHEET - AMZ8001/2 PROCESSOR INSTRUCTION SET

This list is to correct known errors in the first edition of the AmZ8001/2 Instruction Book (Ampub-086, no Rev Letter). These errors are corrected in the second edition (Ampub-086, Rev A).

Page 7	Next to last sentence: "AMZ8002 PC will be automatically loaded from memory address 4 upon reset.
Page 10	Figure 5: bit 13 of FCW should be zero, not SE.
	Figure 6: bit 13 of FCW should be zero, not SE.
	bit 14 of FCW should S/N, not N/S.
Page 11	last para. line 1,"The S/N bit - - - - - - - "
	delete all 3rd para. "The stop enable - - - - - - - "
Page 12	3rd para last line " - - - - - loaded from location 2."
Page 15	2nd sentence: "When the CPU is reset for initialization this bit is set to "0", ie. refresh is disabled."
Page 28	Figure 22A. 3rd box down should read "PC OFFSET"
Page 37	First para. last sentence: "For example, in AmZ8001, four locations are used for segmentation error, four locations for system call trap and so on "
Page 39	Figure 29. Area labeled "SEGMENT TRAP" should read "UNUSED".
Page 50	P/V flag: "Set to 1 on arithmetic overflow. Reset otherwise.
Page 51	ADDB, IM: Operand should be repeated in lower half of immediate word.
	P/V flag: Set to 1 on arithmetic overflow. Reset otherwise.
Page 54	ANDB, IM: Operand should be repeated in lower half of immediate word.
Page 64	COM dst, (Register Mode) Execute cycles should read 7.
Page 65	As per page 64.
Page 68	CPB, IM: Operand should be repeated in lower half of immediate word.
Page 76	Mnemonic should read CPB. (Also change mnemonic above each box.)
Page 76	CPB, DA, SLO: bit 15 of segment word should be a '1'.
Page 76	CPB, all modes Byte operand should be repeated in lower half of immediate word.

NOTE: PLEASE DISCARD ALL PREVIOUSLY PUBLISHED ERRATA SHEETS.

Page 78 — CPIRB: 8 bit opcode should read "10111010"

Page 89 — DBJNZ: Operation, 2nd line: "If Rc<0:7> ≠ 0"

Page 92 — DI: Execution time should be 7 clock cycles

Page 96 — EI; As per Page 92.

Page 98 — EXB, DA, SSO: bit 15 of address should be 0.

Page 120 — LD, DA, X: Format of instruction should be op code followed by address followed by immediate operand.

Page 121 — Delete line 4 of operation:

description should read: "The immediate value in the instruction field, IM, is loaded into the lease significant bits of the destination. In the case of LDK, the destination is a general purpose word register designated by the Rd field of the instruction, and for LDB the destination is a byte register. Following LDK, the remaining bits of the destination register are cleared."

Page 125 — LDB, DA, X Format of instruction should be opcode followed by address followed by immediate operand.

Page 125 — LDB, all modes: Byte operand should be repeated in both halves of immediate word.

Page 125 — LDB, DA, SLO Bit 15 of segment word should be 1.

Page 130 — LDD: Rs and Rd fields should be swapped.

Page 131 — LDDB: As per page 130

Page 132 — LDDR: As per page 130

Page 133 — LDDRB: As per page 130

Page 134 — LDI: As per page 130

Page 135 — LDIB: As per page 130

Page 136 — LDIR: As per page 130

Page 137 — LDIRB: As per page 130

Page 138 — LDL X SSO, SLO: Opcode should read 01011101 Rd≠0 Rs

Page 140 — LDL IR, DA, X: Instruction format should be opcode followed by address followed by immediate operand.

Page 141 — LDM, DA, SSO: bit 15 of address should be 0.

Page 146 — MRES: Execution time should be 5 clock cycles.

Page 147 — MSET: Execution time should be 5 clock cycles.

Page 149 MULTL DA, SSO: Execution time should be 284 + 7n.

Page 149 MULTL X, SSO: Execution time should be 284 + 7n.

Page 153 OR, DA, SSO: bit 15 of address should be 0.

Page 154 ORB, IM: Byte operand should be repeated in lower half of immediate word.

Page 161 OUTD: Line 3 of operation should be deleted

Rd

Page 166 POPL, DA, SSO: bit 15 of address should be 0.

Page 168 PUSHL, IM: Opcode should read "00001101Rd1011"

Page 175 RL should read 10110011 Rd 00b0

Page 176 RLB: should read 10110010 Rd 00b0

Page 177 RLC: should read 10110011 Rd 10b0

Page 178 RLCB: should read 10110011 Rd 10b0

Page 180 RR: should read 10110011 Rd 10b0

Page 181 RRB: should read 10110010 Rd 01b0

Page 182 RRC: should read 10110011 Rd 11b0

Page 183 RRCB: should read 10110010 Rd 11b0

Page 187 SC operation: non segmented operation, last two lines should read:

FCW⟵(NPSAP <0:15> +12)
PC⟵(NPSAP <0:15> +14)

segmented operation, last three lines should read:

FCW⟵(NPSAP <0:22> +24)
PC SEGMENT⟵(NPSAP <0:22> +26)
PC OFFSET⟵(NPSAP <0:22> +28)

Page 216 SRAB: Execution time should real 13 + 3n.

Page 217 SRAL: As per page 216.

Page 222 SUBB, Register Mode Exection time should read 4 clock cycles.

Page 222 SUBB, IM: Byte operand should be repeated in lower half of immediate word.

Page 227	TESTB, X, SSO: Bit 15 of address should be 0.
Page 233	Description para. 4 last sentence: "The translated byte is read from the address and is loaded into the general purpose byte register RH**1** for testing.
Page 234	As per 233.
Page 235	As per 233.
Page 236	As per 233
Page 239	XOR Execution time for DA and X should read 9, 10, 12, 10, 10, 13 clock cyctes respectively.
Page 240	XORB IM Byte operand should be repeated in lower half of immediate word.
Page 43	Second paragraph, fifth line. Delete three sentences beginning with "A byte transferred ..." through "...special I/O operations."
Page 60	CALR Instruction. Last line of both "operation" sections should end with "- 2x displacement", not "+ 2x displacement." Tenth line of description should begin with "subtracted from", not "added to." Range should be +2048 to -2047 words.
Page 199, 200,201, 202,203	Change last paragraph to read "The instruction is identical in operation to the corresponding standard input instruction, except for the CPU status information."
Pp.210,211 212,213,214	Change last paragraph to read: "The instruction is identical in operation to the corresponding standard output instruction, except for the CPU status information. "

ERRATA-AMZ8001/2 PROCESSOR INSTRUCTION SET

Page 7	Line 16, 17: "AmZ8002 PC will be automatically loaded from memory address 4 upon reset.
Page 10	Figure 5: bit 13 of FCW should be zero, not SE.
	Figure 6: bit 13 of FCW should be zero, not SE.
	bit 14 of FCW should S/N, not N/S.
Page 11	last para. line 1,"The S/N bit - - - - - - - "
	delete all 3rd para. "The stop enable - - - - - - - "
Page 12	3rd para last line " - - - - - loaded from location 2."
Page 15	2nd sentence: "When the CPU is reset for initialization this bit is set to "0", ie. refresh is disabled."
Page 28	Figure 22A. 3rd box down should read "PC OFFSET"
Page 37	First para. last sentence: "For example, in AmZ8001, four locations are used for segmentation error, four locations for system call trap and so on "
Page 39	Figure 29. Area labeled "SEGMENT TRAP" should read "UNUSED".
Page 50	P/V flag: "Set to 1 on arithmetic overflow. Reset otherwise.
Page 51	ADDB, IM: Operand should be repeated in lower half of immediate word.
	P/V flag: Set to 1 on arithmetic overflow. Reset otherwise.
Page 54	ANDB, IM: Operand should be repeated in lower half of immediate word.
Page 64	COM dst, (Register Mode) Execute cycles should read 7.
Page 65	As per page 64.
Page 68	CBP, IM: Operand should be repeated in lower half of immediate word.
Page 76	Mnemonic should read CPB. (Also change mnemonic above each box.)
Page 76	CBP, DA, SLO: bit 15 of segment word should be a '1'.
Page 76	CPB, all modes Byte operand should be repeated in lower half of immediate word.

Page 78 — CPIRB: 8 bit opcode should read "10111010"

Page 89 — DBJNZ: Operation, 2nd line: "If Rc<0:7> ≠ 0"

Page 92 — DI: Execution time should be 7 clock cycles

Page 96 — EI; As per Page 92.

Page 98 — EXB, DA, SSO: bit 15 of address should be 0.

Page 120 — LD, DA, X: Format of instruction should be op code followed by address followed by immediate operand.

Page 121 — Delete line 4 of operation: ~~"dst <0:15> ← Ø"~~

description should read: "The immediate value in the instruction field, IM, is loaded into the lease significant bits of the destination. In the case of LDK, the destination is a general purpose word register designated by the Rd field of the instruction, and for LDB the destination is a byte register. Following LDK, the remaining bits of the destination register are cleared."

Page 125 — LDB, DA, X Format of instruction should be opcode followed by address followed by immediate operand.

Page 125 — LDB, all modes: Byte operand should be repeated in both halves of immediate word.

Page 125 — LDB, DA, SLO Bit 15 of segment word should be 1.

Page 130 — LDD: Rs and Rd fields should be swapped.

Page 131 — LDDB: As per page 130

Page 132 — LDDR: As per page 130

Page 133 — LDDRB: As per page 130

Page 134 — LDI: As per page 130

Page 135 — LDIB: As per page 130

Page 136 — LDIR: As per page 130

Page 137 — LDIRB: As per page 130

Page 140 — LDL IR, DA, X: Instruction format should be opcode followed by address followed by immediate operand.

Page 141 — LDM, DA, SSO: bit 15 of address should be 0.

Page 146 — MRES: Execution time should be 5 clock cycles.

Page 147 — MSET: Execution time should be 5 clock cycles.

Page 149 MULTL DA, SSO: Execution time should be 284 + 7n.

Page 149 MULTL X, SSO: Execution time should be 284 + 7n.

Page 153 OR, DA, SSO: bit 15 of address should be 0.

Page 154 ORB, IM: Byte operand should be repeated in lower half of immediate word.

Page 161 OUTD: Line 3 of operation should be deleted

~~Rd <0:15> ⟵ Rd <0:15> -2~~

Page 166 POPL, DA, SSO: bit 15 of address should be 0.

Page 168 PUSHL, IM: Opcode should read "10010001Rd1001"

Page 175 RL: bit 3 of opcode should be 0.

Page 176 RLB: bit 3 of opcode should be 0.

Page 177 RLC: bit 3 of opcode should be 1.

Page 178 RLCB: bit 3 of opcode should be 1.

Page 178 RLCB: bit 1 of opcode should be 'b'.

Page 180 RR: bit 3 of opcode should be 0.

Page 181 RRB: bit 3 of opcode should be 0.

Page 182 RRC: bit 3 of opcode should be 1.

Page 183 RRCB: bit 3 of opcode should be 1.

Page 187 SC operation: non segmented operation, last two lines should read:

FCW⟵(NPSAP <0:15> +12)
PC⟵(NPSAP <0:15> +14)

segmented operation, last three lines should read:

FCW⟵(NPSAP <0:22> +24)
PC SEGMENT⟵(NPSAP <0:22> +26)
PC OFFSET⟵(NPSAP <0:22> +28)

Page 216 SRAB: Execution time should real 13 + 3n.

Page 217 SRAL: As per page 216.

Page 222 SUBB, Register Mode Exection time should read 4 clock cycles.

Page 222 SUBB, IM: Byte operand should be repeated in lower half of immediate word.

Page 227 — TESTB, X, SSO: Bit 15 of address should be 0.

Page 233 — Description para. 4 last sentence: "The translated byte is read from the address and is loaded into the general purpose byte register RHI for testing.

Page 234 — As per 233.

Page 235 — As per 233.

Page 236 — As per 233

Page 239 — XOR Execution time for DA and X should read 9, 10, 12, 10, 10, 13 clock cyctes respectively.

Page 240 — XORB IM Byte operand should be repeated in lower half of immediate word.

ERRATA-AMZ8001/2 PROCESSOR INSTRUCTION SET

Page 7	Line 16, 17: "AmZ8002 PC will be automatically loaded from memory address 4 upon reset.
Page 10	Figure 5: bit 13 of FCW should be zero, not SE.
	Figure 6: bit 13 of FCW should be zero, not SE.
	bit 14 of FCW should S/N, not N/S.
Page 11	last para. line 1,"The S/N bit - - - - - - - "
	delete all 3rd para. "The stop enable - - - - - - - "
Page 12	3rd para last line " - - - - - loaded from location 2."
Page 15	2nd sentence: "When the CPU is reset for initialization this bit is set to "0", ie. refresh is disabled."
Page 28	Figure 22A. 3rd box down should read "PC OFFSET"
Page 37	First para. last sentence: "For example, in AmZ8001, four locations are used for segmentation error, four locations for system call trap and so on "
Page 39	Figure 29. Area labeled "SEGMENT TRAP" should read "UNUSED".
Page 50	P/V flag: "Set to 1 on arithmetic overflow. Reset otherwise.
Page 51	ADDB, IM: Operand should be repeated in lower half of immediate word.
	P/V flag: Set to 1 on arithmetic overflow. Reset otherwise.
Page 54	ANDB, IM: Operand should be repeated in lower half of immediate word.
Page 64	COM dst, (Register Mode) Execute cycles should read 7.
Page 65	As per page 64.
Page 68	CBP, IM: Operand should be repeated in lower half of immediate word.
Page 76	Mnemonic should read CPB. (Also change mnemonic above each box.)
Page 76	CBP, DA, SLO: bit 15 of segment word should be a '1'.
Page 76	CPB, all modes Byte operand should be repeated in lower half of immediate word.

Page 78 CPIRB: 8 bit opcode should read "10111010"

Page 89 DBJNZ: Operation, 2nd line: "If Rc<0:7> ≠ 0"

Page 92 DI: Execution time should be 7 clock cycles

Page 96 EI; As per Page 92.

Page 98 EXB, DA, SSO: bit 15 of address should be 0.

Page 120 LD, DA, X: Format of instruction should be op code followed by address followed by immediate operand.

Page 121 Delete line 4 of operation: "dst ~~<0:15> ← Ø~~"

description should read: "The immediate value in the instruction field, IM, is loaded into the lease significant bits of the destination. In the case of LDK, the destination is a general purpose word register designated by the Rd field of the instruction, and for LDB the destination is a byte register. Following LDK, the remaining bits of the destination register are cleared."

Page 125 LDB, DA, X Format of instruction should be opcode followed by address followed by immediate operand.

Page 125 LDB, all modes: Byte operand should be repeated in both halves of immediate word.

Page 125 LDB, DA, SLO Bit 15 of segment word should be 1.

Page 130 LDD: Rs and Rd fields should be swapped.

Page 131 LDDB: As per page 130

Page 132 LDDR: As per page 130

Page 133 LDDRB: As per page 130

Page 134 LDI: As per page 130

Page 135 LDIB: As per page 130

Page 136 LDIR: As per page 130

Page 137 LDIRB: As per page 130

Page 140 LDL IR, DA, X: Instruction format should be opcode followed by address followed by immediate operand.

Page 141 LDM, DA, SSO: bit 15 of address should be 0.

Page 146 MRES: Execution time should be 5 clock cycles.

Page 147 MSET: Execution time should be 5 clock cycles.

Page 149 MULTL DA, SSO: Execution time should be 284 + 7n.

Page 149 MULTL X, SSO: Execution time should be 284 + 7n.

Page 153 OR, DA, SSO: bit 15 of address should be 0.

Page 154 ORB, IM: Byte operand should be repeated in lower half of immediate word.

Page 161 OUTD: Line 3 of operation should be deleted

~~Rd <0:15> ← Rd <0:15> -2~~

Page 166 POPL, DA, SSO: bit 15 of address should be 0.

Page 168 PUSHL, IM: Opcode should read "10010001Rd1001"

Page 175 RL: bit 3 of opcode should be 0.

Page 176 RLB: bit 3 of opcode should be 0.

Page 177 RLC: bit 3 of opcode should be 1.

Page 178 RLCB: bit 3 of opcode should be 1.

Page 178 RLCB: bit 1 of opcode should be 'b'.

Page 180 RR: bit 3 of opcode should be 0.

Page 181 RRB: bit 3 of opcode should be 0.

Page 182 RRC: bit 3 of opcode should be 1.

Page 183 RRCB: bit 3 of opcode should be 1.

Page 187 SC operation: non segmented operation, last two lines should read:

FCW←(NPSAP <0:15> +12)
PC←(NPSAP <0:15> +14)

segmented operation, last three lines should read:

FCW←(NPSAP <0:22> +24)
PC SEGMENT←(NPSAP <0:22> +26)
PC OFFSET←(NPSAP <0:22> +28)

Page 216 SRAB: Execution time should real 13 + 3n.

Page 217 SRAL: As per page 216.

Page 222 SUBB, Register Mode Exection time should read 4 clock cycles.

Page 222 SUBB, IM: Byte operand should be repeated in lower half of immediate word.

Page 227 TESTB, X, SSO: Bit 15 of address should be 0.

Page 233 Description para. 4 last sentence: "The translated byte is read from the address and is loaded into the general purpose byte register RHI for testing.

Page 234 As per 233.

Page 235 As per 233.

Page 236 As per 233

Page 239 XOR Execution time for DA and X should read 9, 10, 12, 10, 10, 13 clock cyctes respectively.

Page 240 XORB IM Byte operand should be repeated in lower half of immediate word.