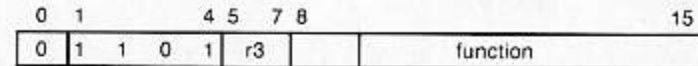


The character string handling instructions are coded in the new 'D' format:



in which 'r3' specifies one of the registers A0 to A7 (for the currently available instructions this field = 0) and 'function' qualifies the operation code.

The parameters for a 'D' format instruction must be loaded into registers before the instruction is executed. The instructions MVS and CS use registers A2, A3 and A4; MVA uses registers A1, A2 and A4, CA uses registers A1, A2 and A3.

An attempt to execute a not wired 'D' format instruction (i.e. OPC = 1101, but 'function' # /00, /40, /80 or /A0) causes an indirect branch via /7C; this trap action can be used for software simulation of 'D' format instructions not available in the instruction set.

In the P853M, all instructions with OPC = 1101 (i.e. character string instructions) are trapped.

MVS*Move string***MVS**P854
P858
P859

Syntax: [label] MVS

This instruction copies a string of consecutive characters from one memory area into another area. The parameters for the instruction must be specified in registers A2, A3 and A4:

- register A3 must be loaded with the first address of the string to be copied
- register A4 must be loaded with the first address of the receiving area
- register A2 must contain the number of characters to be moved.

To avoid overwriting, the instruction compares the contents of registers A3 and A4, and decides on the direction of the move: if $(A3) < (A4)$, characters are moved from higher addresses towards lower addresses, if $(A3) > (A4)$, characters are moved from lower addresses towards higher addresses.

The execution of this instruction may be interrupted after any character transfer; execution is resumed when the interrupt has been serviced.

When execution of the instruction is terminated, register A2 contains 0, and the contents of registers A3 and A4 are not significant.

Type	Function
TBD	if $(A3) > (A4)$ the string is moved forward: $((A3)) \rightarrow (A4)$ $(A3) + 1 \rightarrow A3$ this sequence is repeated $(A4) + 1 \rightarrow A4$ until $(A2) = 0$ $(A2) - 1 \rightarrow A2$ scan interrupts
	if $(A3) < (A4)$ the register contents are modified: $(A3) + (A2) - 1 \rightarrow A3$ $(A4) + (A2) - 1 \rightarrow A4$ and the string is moved backward: $((A3)) \rightarrow (A4)$ $(A3) - 1 \rightarrow A3$ this sequence is repeated $(A4) - 1 \rightarrow A4$ until $(A2) = 0$ $(A2) - 1 \rightarrow A2$ scan interrupts

Note: A3 and A4 contain character addresses, so in each move step one character is moved.

Condition register: Unchanged

bit	0	1		4	5		7	8						15
	0	1	1	0	1	0	0	0	0	0	0	0	0	0

MVA*Move A11***MVA**P854
P858
P859

Syntax: [label] MVA

This instruction loads the character specified in bit 8-15 of register A1 into all character locations of the memory area defined by the start address specified in register A4 and the length specified in register A2.

The execution of this instruction may be interrupted after any character transfer; execution is resumed when the interrupt has been serviced.

When the execution of this instruction is terminated, register A2 contains 0, and register A4 contains (start address + length); register A1 remains unchanged.

Type	Function
TBD	$(A1)_{8-15} \rightarrow (A4)$ this sequence is repeated $(A4) + 1 \rightarrow A4$ until $(A2) = 0$ $(A2) - 1 \rightarrow A2$ scan interrupts

Condition register: Unchanged

bit	0	1		4	5		7	8						15
	0	1	1	0	1	0	0	0	0	1	0	0	0	0

Syntax: [label] CS

This instruction compares the contents of two character strings, character by character, until a difference is detected.

The parameters for the instruction must be specified in registers A2, A3 and A4:

- register A3 must be loaded with the first address of string 1
- register A4 must be loaded with the first address of string 2
- register A2 must be contain the number of characters in string 2

The instruction compares the two strings, until a difference is detected or all characters have been compared. The condition register is set to indicate the difference.

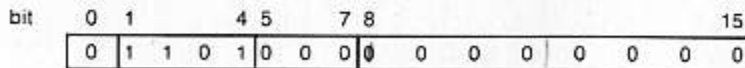
The execution of this instruction may be interrupted after any character transfer; execution is resumed when the interrupt has been serviced.

When execution of this instruction is terminated, registers A3 and A4 point to the addresses where the difference was detected, and A2 contains the remaining length. If no difference was detected, registers A3 and A4 point to the next address after the specified areas, and A2 = 0.

Type	Function
T8D	0 → CR
	((A3)) → ((A4))
	if ((A3)) = ((A4)): (A3) + 1 → A3 this sequence is
	(A4) + 1 → A4 repeated until
	(A2) - 1 → A2 a difference is
	scan interrupts detected
	if ((A3)) ≠ ((A4)): if ((A3)) > ((A4)) 1 → CR
	if ((A3)) < ((A4)) 2 → CR

Condition register:

CR = 0 if all compared characters are identical
 CR = 1 if character in string 1 > character in string 2
 CR = 2 if character in string 1 < character in string 2



Syntax: [label] CA

This instruction compares a single character with all characters in a character string, until a difference is detected.

The parameters for the instruction are specified in registers A1, A2 and A3:

- register A1 must contain a character value in bits 8-15
- register A3 must contain the first address of the string
- register A2 must contain the string length (in characters)

If all characters in the string are identical to the character in register A1, the condition register is set to 0; if a difference is detected, register A3 points to the character that is different from the value in A1, and the condition register is set to indicate the difference.

The execution of this instruction may be interrupted after any character transfer; execution is resumed when the interrupt has been serviced.

The function of the instruction is:

Type	Function
T8D	0 → CR
	(A3) → (A1)
	if (A3) = (A1): (A3) + 1 → A3 this sequence is
	(A2) - 1 → A2 repeat until A2 = 0
	scan interrupts or (A3) ≠ (A1)
	if (A3) ≠ (A1): if (A3) < (A1) 1 → CR
	if (A3) > (A1) 2 → CR

Condition register:

CR = 0 if all compared characters are identical
 CR = 1 if character in string 1 < character in A2
 CR = 2 if character in string 1 > character in A2

