MVF *Move Table Forward* MVF P857M

Syntax: [label]␣MVF␣r2

This instruction copies a string of consecutive words from one memory area into another area, beginning with the last location from the buffer to be copied towards the start address of that buffer. Should the buffer to be copied and the receiving buffer overlap, the user must take care not to overwrite the contents of the locations in the buffer to be copied. Use in that case the instruction MVB.

- register A1 must be loaded with the start address of the memory area to be copied.
- register A2 must be loaded with the start address of the receiving buffer.
- register r2 must contain the number of characters to be copied (the number must be even and unsigned).

The execution of this instruction may be interrupted after any word transfer. When the interrupt is accepted the contents of the instruction counter, which is pointing to this instruction, are saved in the stack. The contents of A1 and A2 remain unchanged.

Register r2 contains the remaining number of characters to be transferred. The execution of this instruction is resumed when the interrupt has been serviced. When the execution is terminated A1 and A2 contain the initial values.

Type	*Function*
T8	(r2) − 2 → r2, ((A1) + (r2)) → (A2) + (r2)
	–
	–
	0 → r2, ((A1)) → (A2)

Condition register: Unchanged

bit	0	1			4	5		7	8		10	11	14	15
	0	1	1	1	0	0	0	0	0	0	0	r2		0

Remark:

- * When used in system mode r2 = A15 or ≠ A15.
- * When used in user mode r2 ≠ A15.
- * r2 must be ≠ 0.

MVB Move Table Backward MVB P857M

Syntax: [label]␣MVB␣r2

This instruction copies a string of consecutive words from one memory area into another area, beginning with the first location of the buffer to be copied towards the last location of that buffer. Should the buffer to be copied and the receiving buffer overlap, the user must take care not to overwrite the contents of locations in the buffer to be copied. Use in that case the instruction MVF.

- register A1 must contain the start address of the buffer to be copied.
- register A2 must contain the start address of the receiving buffer.
- register r2 must contain the number of characters to be copied. (The number must be even and unsigned.)

The execution of this instruction may be interrupted after any word transfer. When the interrupt is accepted the contents of the instruction counter, which points to this instruction, are saved in the stack. The contents of registers A1 and A2 point to the first location to be transferred when resuming the execution. Register r2 contains the remaining number of characters to be transferred.
The execution of the instruction is resumed when the instruction interrupt has been serviced. When the execution is terminated A1 and A2 point to the first address after the buffer.

```
Type   Function
T8     ((A1))      → (A2)
       ( r2  ) − 2 → r2;  (A1) + 2 → A1;  (A2) + 2 → A2;  ((A1)) → (A2)
         –
         –
       0 → r2;  (A1) + 2 → A1;  (A2) + 2 → A2
```

Condition register: Unchanged

bit	0	1			4	5		7	8	9	10	11			14	15
	0	1	1	1	1	0	0	0	0	0	0	r2				0

Remark:

- When used in system mode r2 = A15 or ≠ A15.
- When used in user mode r2 ≠ A15.
- r2 must be ≠ 0.

MVUS Move Table from User to System area (MMU option) MVUS P857M

Syntax: [label]␣MVUS␣r2

This instruction is used to copy a table of consecutive words from a user area (sending buffer) to a system area (receiving buffer), beginning with the first location towards its last location.

- register A1 must contain the logical start address (MMU) of the buffer to be copied.
- register A2 must contain the physical start address (NO MMU) of the receiving buffer.
- register r2 must be loaded with the number of characters to be copied. This number must be even and not signed.

The execution of this instruction may be interrupted after any word transfer. When the interrupt is accepted the contents of the instruction counter, which points to this instruction, are saved in the stack. The contents of A1 and A2 point to the first location to be transferred when the execution is resumed.
Register r2 contains the remaining number of characters to be transferred. The execution of this instruction is resumed when the interrupt is serviced. When the execution is terminated A1 and A2 point to the first address after the receiving buffer.

```
Type   Function
T8     ((A1))      → (A2)
       ( r2 )− 2   → r2;  (A1) + 2 → A1;  (A2)+ 2 → A2;  ((A1)) → (A2)
         –
         –
       0 → r2; (A1) + 2 → A1;  (A2) + 2 → A2
```

Condition register: Unchanged

bit	0	1			4	5		7	8	9	10	11			14	15
	0	1	1	1	1	0	0	0	1	0	0	r2				0

Remark:

- When used in system mode r2 = A15 or ≠ A15.
- When used in user mode r2 ≠ A15. In that case or if MMU is not available this instruction is the same as the MVB instruction.
- r2 must be ≠ 0.

MVSU

Move Table from System to User area
(MMU option)

MVSU P857M

Syntax : [label] ␣ MVSU ␣ r2

This instruction is used to copy a table of consecutive words from a system area (sending buffer) to a user area (receiving buffer), beginning with the last location of the sending area towards the first location.

- register A1 must contain the physical start address (NO MMU) of the sending buffer.
- register A2 must contain the logical start address (MMU) of the receiving buffer.
- register r2 must be loaded with the number of characters to be copied (this number must be even and unsigned).

The execution of this instruction may be interrupted after any word transfer. When the interrupt is accepted the contents of the instruction counter, which points to this instruction, are saved in the stack. The contents of registers A1 and A2 remain unchanged.

Register r2 contains the remaining number of characters to be transferred. The execution of the instruction is resumed when the interrupt is serviced. When the execution is terminated A1 and A2 contain their initial values.

Type *Function*

T8 (r2) − 2 → r2, ((A1) + (r2)) → (A2) + (r2)

—

—

0 → r2 , ((A1)) → (A2)

Condition register: Unchanged

bit	0	1			4	5		7	8		10	11			14	15
	0	1	1	1	0	0	0	0	1	0	0	r2				0

Remark:

- When used in system mode r2 = A15 or ≠ A15.
- When used in user mode r2 ≠ A15. In that case or if MMU is not available this instruction is the same as the MVF instruction.
- r2 must be ≠ 0.