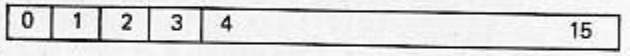
WER — *Write external register* — **WER** — P851M, P852M, P856M, P857M

Syntax: [label] ␣ WER ␣ r3, address

The contents of the register specified by r3 are transferred to the external register whose address is specified in bits 8–15. The contents of the register specified by r3 and the condition register remain unchanged.

Two WER instructions must be used to send two control words, one containing a buffer address and the second one containing the number of words or characters to be transferred, to two registers on the I/O Processor.

1st control word



where: bit 0 = 0 if char. mode
1 if word mode

bit 1 = 0 if transfer is CU → MEM
1 if transfer is MEM → CU

bits 2, 3 are used to extend the memory address in 2nd control word to > 32K.

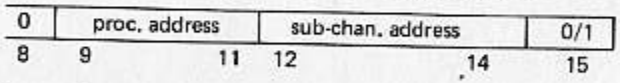
bits 4 through 15 = transfer length in words or characters.

2nd control word

where: bits 0 through 14 = memory address

bit 15 (if char. mode) = 0 left hand character
1 right hand character

The layout of bits 8 through 15 of the WER instruction is:



bit 8 = 0

bits 9 through 11 must be set to zero for the P851M (only one processor connection.
On P852M, P856M or P857M it may be a number from 0 through 7.

bits 12 through 14 device address

bit 15 = 0 WER instruction for the 1st control word
1 WER instruction for the 2nd control word

Note: When a device must be addressed via a multiple control unit card, specify the lowest address.

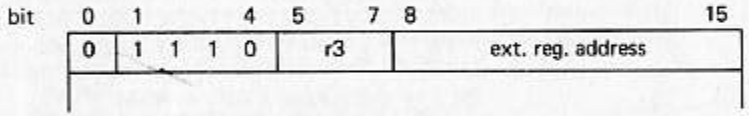
Example:

Output on cassette

LDK A1,/0084	Send 132 characters.
LDKL A2,BUFFER	Take the contents of 'BUFFER'.
WER A1,/A	The cassette has address /05 and is connected to I/O Processor numero 0. Bit 15 = 0 (1st control word).
WER A2,/B	Send the 2nd control word. Bit 15 = 1.

Type	*Function*
T8	(r3) → extern reg.

Condition register: Unchanged



Remark:
* r3 must be ≠ 0.
* This instruction may only be used in system mode.

RER *Read external register* **RER**

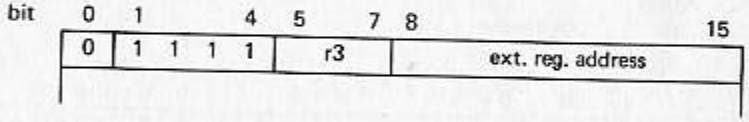
P851M
P852M
P856M
P857M

Syntax: [label] ␣ RER ␣ r3, address

The contents of the external register, specified by its address, are transferred to the register specified by r3. The contents of the external register remain unchanged. Bits 6 and 7 of the external register are copied to the condition register.
Through this instruction the user can check how many characters or words have been transferred.

Type	*Function*
T8	(extern reg) → r3

Condition register: (extern reg 6,7) → CR



Remark:
* r3 must be ≠ 0.
* This instruction may only be used in system mode.

TLR

Segment Table Load/register (MMU option)

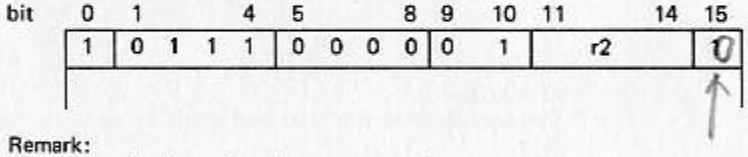
TLR | P857M

Syntax: [label]␣ TLR ␣ r2

This instruction loads 16 consecutive registers, TR0 through TR15, which are located on the MMU, with the contents of 16 consecutive memory locations, the first one being indicated in register r2.

Type	*Function*	
T3	((r2))	→ TR0
	((r2) + 2)	→ TR1
	—	
	—	
	—	
	((r2) + 15 x 2)	→ TR15

Condition register: Unchanged



Remark:
This instruction is restricted to system mode.

TL

Segment Table Load (MMU option)

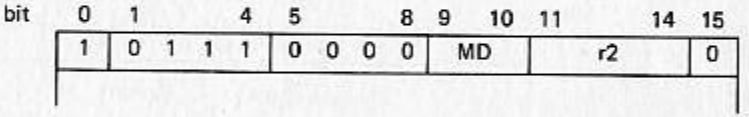
TL | P857M

Syntax: [label]␣ TL[*]␣ m[, r2]

This instruction loads 16 consecutive registers, TR0 through TR15, which are located on the MMU, with the contents of 16 consecutive memory locations.
The address of the first memory location is indicated by the effective memory address.

Type	*Function*		*Syntax*	
T4	(m) . . (m + 15x2)	→ TR0 . . TR15	TL	m
T5	(m + (r2)) . . (m + (r2) + 15x2)	→ TR0 . . TR15	TL	m, r2
T6	((m)) . . ((m) + 15x2)	→ TR0 . . TR15	TL*	m
T7	((m+ (r2))) . . ((m + (r2)) + 15x2)	→ TR0 . . TR15	TL*	m, r2

Condition register: Unchanged



Remark:
This instruction is restricted to system mode.

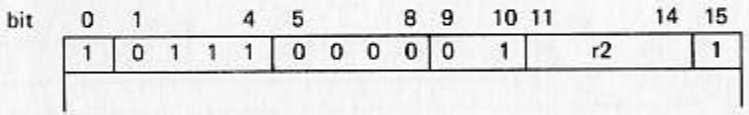
TSR — *Segment Table Store/register (MMU option)* — **TSR** — **P857M**

Syntax: [label] ␣ TSR ␣ r2

This instruction places the contents of 16 consecutive registers, TR0 through TR15, located on the MMU, in 16 consecutive memory locations. The first memory location is indicated by the contents of register r2.

Type	*Function*	
T3	(TR0)	→ (r2)
	(TR1)	→ (r2) + 2
	—	
	—	
	—	
	(TR15)	→ (r2) + 15x2

Condition register: Unchanged



Remark:
This instruction is restricted to system mode.

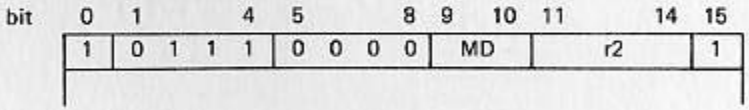
TS — *Segment Table Store* — **TS** — **P857M**

Syntax: [label] ␣ TS[*] ␣ m[, r2]

The contents of 16 consecutive registers, TR0 through TR15, located on the MMU, replace the contents of 16 memory locations. The first memory location is indicated by the effective memory address.

Type	*Function*		*Syntax*	
T4	(TR0)	→ m		
	(TR1)	→ m + 2		
	—			
	(TR15)	→ m + 15x2	TS	m
T5	(TR0)	→ m + (r2)		
	(TR1)	→ m + (r2) + 2		
	—			
	(TR15)	→ m + (r2) + 15x2	TS	m, r2
T6	(TR0)	→(m)		
	(TR1)	→(m + 2)		
	—			
	(TR15)	→(m + 15x2)	TS*	m
T7	(TR0)	→(m + (r2))		
	(TR1)	→(m + (r2) + 2)		
	—			
	(TR15)	→(m + (r2) + 15x2)	TS*	m, r2

Condition register: Unchanged



Remark:
This instruction is restricted to system mode.

Syntax: [label]␣ FLDR ␣ r2

The contents of three consecutive memory locations are loaded into three accumulators FPA1, FPA2, FPA3 on the Floating Point Processor.
The first memory location is indicated in the register r2.

Type	*Function*	
T3	((r2))	→ FPA1
	((r2) + 2)	→ FPA2
	((r2) + 4)	→ FPA3

Condition register:

CR = 0 if floating point operand = 0
1 if floating point operand > 0
2 if floating point operand < 0

bit

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	0	0	0	0	0	1	0	0	1	r2				0

Syntax: [label]␣ FLD[*]␣ m[, r2]

The contents of three consecutive memory locations are loaded into three accumulators FPA1, FPA2 and FPA3 on the Floating Point Processor.
The first memory location is indicated by the effective memory address.

Type	*Function*		*MD*	*Syntax*
T4	(m)	→ FPA1	10	FLD m
	(m + 2)	→ FPA2		
	(m + 4)	→ FPA3		
T5	(m + (r2))	→ FPA1	10	FLD m, r2
	(m + (r2) + 2)	→ FPA2		
	(m + (r2) + 4)	→ FPA3		
T6	((m))	→ FPA1	11	FLD* m
	((m) + 2)	→ FPA2		
	((m) + 4)	→ FPA3		
T7	((m + (r2)))	→ FPA1	11	FLD* m, r2
	((m + (r2)) + 2)	→ FPA2		
	((m + (t2)) + 4)	→ FPA3		

Condition register:

CR = 0 if floating point operand = 0
1 if floating point operand > 0
2 if floating point operand < 0
3 unnormalized operand (operation aborted)

bit

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	0	0	0	0	0	1	0	MD		r2				0

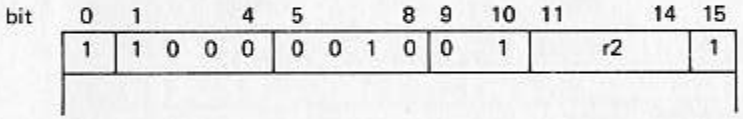
Floating Point Store/register (F.F.P. option)

Syntax: [label]␣ FSTR ␣ r2

The contents of three accumulators FPA1, FPA2 and FPA3 on the Floating Point Processor replace the contents of three consecutive memory locations. The first location is indicated in register r2.

Type	*Function*	
T3	(FPA1)	→ (r2)
	(FPA2)	→ (r2) + 2
	(FPA3)	→ (r2) + 4

Condition register: Unchanged



Floating Point Store (F.F.P. option)

Syntax: [label]␣ FST[*]␣ m[, r2]

The contents of three accumulators FPA1, FPA2 and FPA3 on the floating point processor replace the contents of three consecutive memory locations. The first location is indicated by the effective memory address.

Type	*Function*		*MD*	*Syntax*	
T4	(FPA1)	→ m	10	FST	m
	(FPA2)	→ m + 2			
	(FPA3)	→ m + 4			
T5	(FPA1)	→ m + (r2)	10	FST	m, r2
	(FPA2)	→ m + (r2) + 2			
	(FPA3)	→ m + (r2) + 4			
T6	(FPA1)	→(m)	11	FST*	m
	(FPA2)	→(m) + 2			
	(FPA3)	→(m) + 4			
T7	(FPA1)	→(m + (r2))	11	FST*	m, r2
	(FPA2)	→(m + (r2) + 2)			
	(FPA3)	→(m + (r2) + 4)			

Condition register: Unchanged