SLA *Single left arithmetic shift* **SLA**

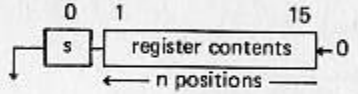
P851M
P852M
P856M
P857M

Syntax: [label] ␣ SLA ␣ r3, n

The bits of the register specified by r3 are shifted left n bit positions. Overflow occurs when the sign bit was modified during the operation. Vacant bits are filled with zeroes.

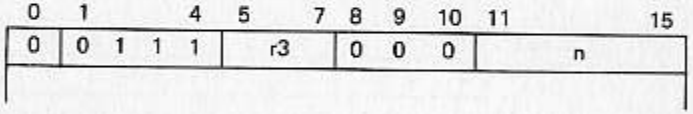
Type *Function*

T8



Condition register:

CR = 0 if result = 0
1 if result > 0
2 if result < 0
3 in case of overflow



Remark:
r3 ≠ 0.

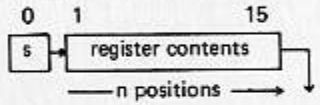
SRA *Single right arithmetic shift* **SRA** P851M P852M P856M P857M

Syntax: [label]␣ SRA ␣ r3, n

The contents of the register specified by r3 are shifted right n bit positions. The sign bit is not changed. It is shifted into the vacant position(s) of the register. The vacant bit positions are filled with the same values as the sign bit, i.e. either 0 or 1.
If n ≥ 15, all bits of the register will be the same as the sign bit.

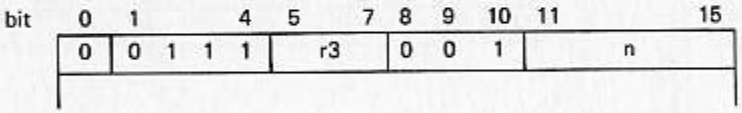
Type *Function*

T8



Condition register:

CR = 0 if result = 0
1 if result > 0
2 if result < 0



Remark:
r3 ≠ 0.

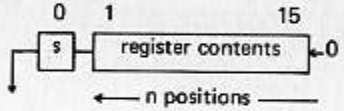
SLL *Single left logical shift* **SLL** P851M P852M P856M P857M

Syntax: [label]␣ SLL ␣ r3, n

The bits of the register specified by r3 are shifted left n bit positions. Vacant bits become zero. After 16 or more shifts the whole register contains zero.

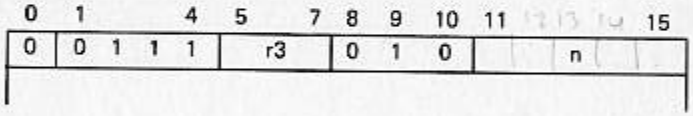
Type *Function*

T8



Condition register:

CR = 0 if result = 0
1 if result > 0
2 if result < 0



Remark:
r3 ≠ 0.

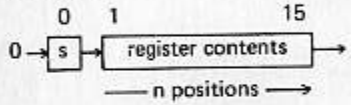
SRL *Single right logical shift* **SRL** P851M P852M P856M P857M

Syntax: [label] ␣ SRL ␣ r3, n

The contents of the register specified by r3 are shifted right n bit positions. Vacant bits become zero. After 16 or more shifts the register contains zero.

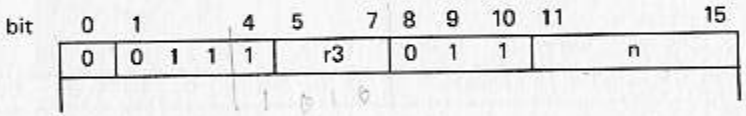
Type *Function*

T8



Condition register:

CR = 0 if result = 0
1 if result > 0
2 if result < 0



Remark:
r3 ≠ 0.

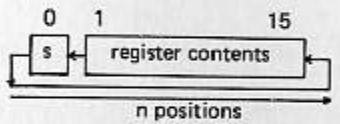
SLC *Single left circular shift* **SLC** P851M P852M P856M P857M

Syntax: [label] ␣ SLC ␣ r3, n

The contents of the register specified by r3 are shifted left, end around, n bit positions.

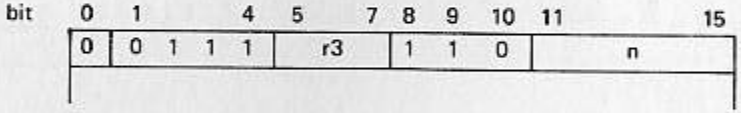
Type *Function*

T8



Condition register:

CR = 0 if result = 0
1 if result > 0
2 if result < 0



Remark:
r3 ≠ 0.

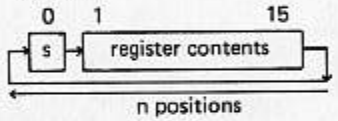
SRC *Single right circular shift* **SRC** P851M P852M P856M P857M

Syntax: [label]␣SRC␣r3, n

The contents of the register specified by r3 are shifted right, end around, n bit positions.

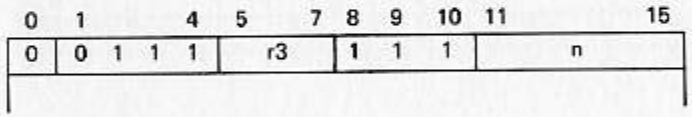
Type *Function*

T8



Condition register:

CR = 0 if result = 0
1 if result > 0
2 if result < 0



Remark:
r3 ≠ 0.

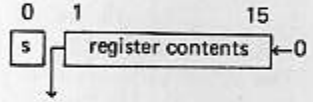
SLN *Single left and normalize shift* **SLN** P851M P852M P856M P857M

Syntax: [label]␣SLN␣r3, r2

The contents of the register specified by r3 are shifted left until the two most significant bits differ. The sign bit remains unaffected; zero bits are inserted in the least significant positions. The number of shifted positions is placed in the register specified by r2.

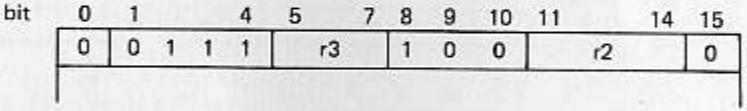
Type *Function*

T8



Condition register:

Unchanged



Remark:
* r3 ≠ 0.
* If (r3) = 0 the number of shifted positions will be 16.
* Restricted to system mode if r2 = A15.

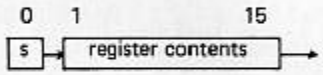
SRN *Single right and normalize shift* **SRN** P851M P852M P856M P857M

Syntax: [label]␣SRN␣r3, r2

The contents of the register specified by r3 are shifted right until a 1-bit appears in bit 15 of that register.
The sign bit is not changed and is copied each time a shift is given.
The number of times a shift had to be performed is placed in the register specified in r2.

Type *Function*

T8



Condition register: Unchanged

bit

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	1	1	1	r3			1	0	1	r2				0

Remark:
* r3 ≠ 0.
* If (r3) = 0 the number of shifted positions will be 16.
* Restricted to system mode if r2 = A15.

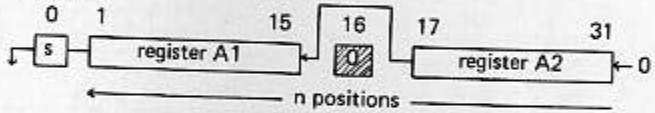
DLA *Double left arithmetic shift* **DLA** P851M P852M P856M P857M (Softw. sim.)

Syntax: [label]␣DLA␣n

This instruction treats the A1 and A2 register as one 31-bit register (bit 0 of A2 is set to zero). The contents are shifted left n positions and zeroes are placed from the right in vacated positions.
Overflow occurs when the sign bit is changed during execution of this instruction.

Type *Function*

T8



Condition register:

CR = 0 if result = 0
1 if result > 0
2 if result < 0
3 in case of overflow

bit

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

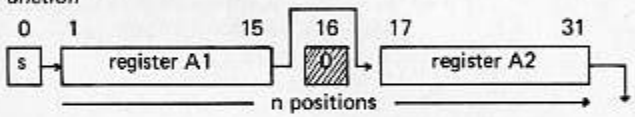
DRA *Double right arithmetic shift* **DRA** P851M P852M P856M P857M (Softw. sim.)

Syntax: [label] ␣ DRA ␣ n

This instruction treats the A1 and A2 registers as one 31-bit register. The contents are shifted right n positions and zeroes or ones are propagated into vacated positions depending on the value of the sign bit of A1.
After 30 or more shifts the two registers are filled the value of the sign bit (all zeroes or ones), except for the sign bit of A2 which is always set to 0.

Type *Function*

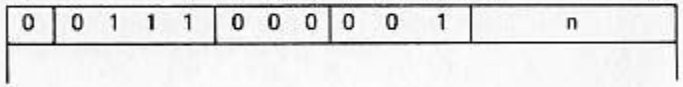
T8



Condition register:

CR = 0 if result = 0
1 if result > 0
2 if result < 0

bit



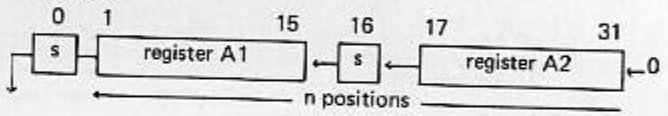
DLL *Double left logical shift* **DLL** P851M P852M P856M P857M (Softw. sim.)

Syntax: [label] ␣ DLL ␣ n

This instruction treats the registers A1 and A2 as one 32-bit register. The contents are shifted left n positions. Zeroes are propagated into vacated positions of A1 and A2.

Type *Function*

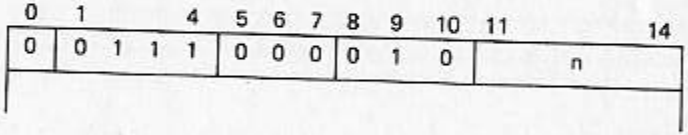
T8



Condition register:

CR = 0 if result = 0
1 if result > 0
2 if result < 0

bit



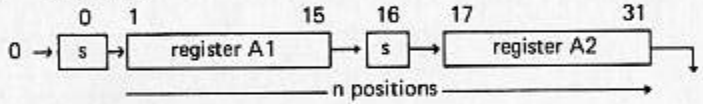
DRL *Double right logical shift* **DRL** P851M P852M P856M P857M (Softw. sim.)

Syntax: [label] ␣ DRL ␣ n

The A1 and A2 registers are treated as one 32-bit register. The contents are shifted right n positions. Zeroes are propagated into vacated positions. The max. number of shifts is 31.

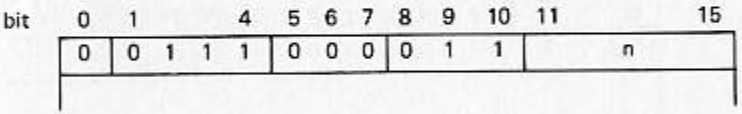
Type *Function*

T8



Condition register:

CR = 0 if result = 0
1 if result > 0
2 if result < 0



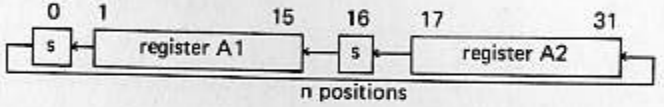
DLC *Double left circular shift* **DLC** P851M P852M P856M P857M (Softw. sim.)

Syntax: [label] ␣ DLC ␣ n

The A1 and A2 registers are treated as one 32-bit register. The contents are shifted left, end around, n positions.

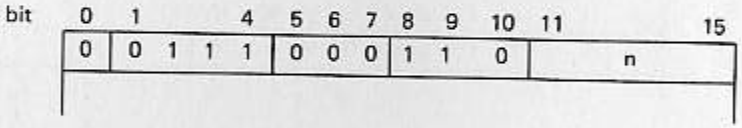
Type *Function*

T8



Condition register:

CR = 0 if result = 0
1 if result > 0
2 if result < 0



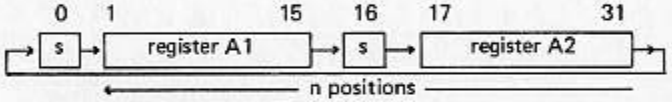
DRC *Double right circular shift* **DRC** P851M P852M P856M P857M (Softw. sim.)

Syntax: [label]␣DRC␣n

The A1 and A2 registers are treated as one 32-bit register. The contents are shifted right, end around, n positions.

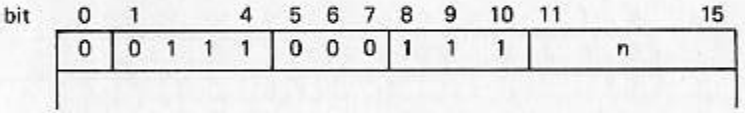
Type *Function*

T8



Condition register:

CR = 0 if result = 0
1 if result > 0
2 if result < 0



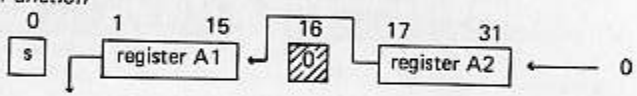
DLN *Double left and normalize shift* **DLN** P851M P852M P856M P857M (Softw. sim.)

Syntax: [label]␣DLN␣r2

The A1 and A2 registers are treated as one 31-bit register. Its contents are shifted left until bit zero and bit one have a different value.
Zeroes are shifted, from the right hand side on, into vacated positions of the register. The sign bit of register A1 remains unchanged.
The number of shifted positions is stored in register r2.
The sign bit of A2 becomes zero.

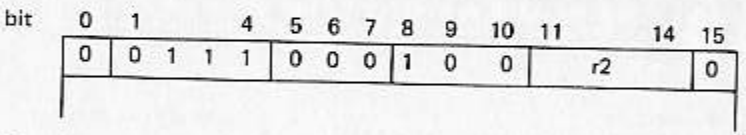
Type *Function*

T8



Condition register:

Unchanged



Remark:
Restricted to system mode if r2 = A15.

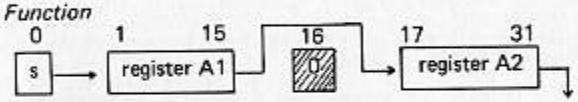
DRN *Double right and normalize shift* **DRN** P851M P852M P856M P857M (Softw. sim.)

Syntax: [label] ␣ DRN ␣ r2

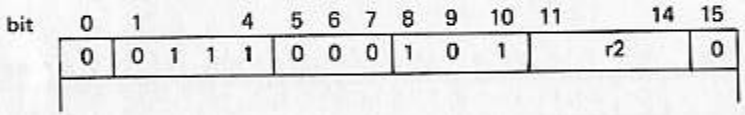
The A1 and A2 registers are treated as one 31-bit register. The contents are shifted right until a 1-bit appears in the least significant position of the register. The sign bit is shifted to the right each time a shift takes place.
The number of shifted postions is stored in register r2.
The sign bit of A2 becomes zero.

Type *Function*

T8



Condition register: Unchanged



Remark:
Restricted to system mode if r1 = A15.