

The branch instructions AB, ABL, ABR, ABI, RB and RF branch to an address or the contents of an address or register when a certain condition is fulfilled. If that condition does not arise the program determines the next instruction to be executed. The condition is given by a number from 1 through 7 or by one or two letters. The following table gives a survey:

Condition Notation

Cond. reg. contents	(cnd)			
	GENERAL	ARITHM.	COMPARE	I/O
0	(0)	(Z) Zero	(E) Equal	(A) Accepted
1	(1)	(P) Pos.	(G) Greater	(R) Refused
2	(2)	(N) Neg.	(L) Less	—
3	(3)	(O) Overfl.	—	(U) Unknown
NOT - Condition				
≠ 0	(4)	(NZ) Not Zero	(NE) Not Equal	(NA) Not Accepted
≠ 1	(5)	(NP) Not Pos.	(NG) Not Greater	(NR) Not Refused
≠ 2	(6)	(NN) Not Neg.	(NL) Not Less	—
n.s.	(7)	Unconditional		

Note:

The instruction counter P always points to the next instruction to be executed. Wherever in the description the notation (P) + 2 (or 4) appears, the hardware function is meant. When the following program must be assembled calculate the displacement in locations as follows:

```
BEGIN EQU *
      HLT
      LDK A1,/000A
      SUK A1,2
      RF(Z) ++4
      RB +-4
      ABL +-8
      END START
```

where ++4 refers to ABL
 +-4 refers to SUK
 +-8 refers to LDK

When the same program is to be put in memory with the toggle switches the value for RF(Z) ++4 is 5002 and not 5004 as the P-register is already pointing to the next instruction.

The value for RB +-4 must be 5F06 and not 5F04, as the P-register is already pointing to the next instruction.

The address in the ABL instruction must be the relative address pointing to LDK.

The values put in memory for the program listed above must be:

```

207F  START  HLT
010A          LDK  A1,/000A
1902          SUK  A1,2
5002          RF(Z)  ++4
5F06          RB   +-4
8F20          ABL  +-8
0002          END  START

```

AB
ABL

Absolute conditional branch

AB
ABL

P851M
P852M
P856M
P857M

Syntax: [label]┌ AB [(cnd)]┌ k - T8
 └ ABL[(cnd)]└ lk - T2

This instruction means that the next instruction to be executed is found either at the address specified by the constant (k indicating one of the first 256 addresses of the memory, lk being specified in the word following the instruction) or in normal sequence, depending on the (cnd) and the contents of the condition register.

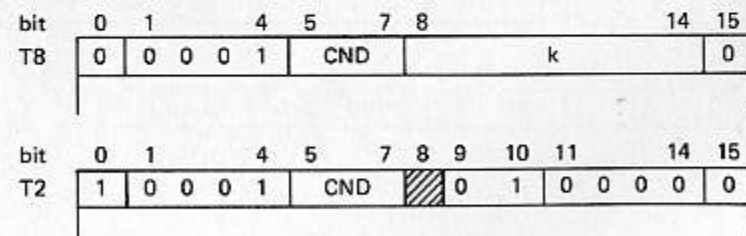
If (cnd) is equal to (7) the next instruction is at the effective memory address. Note that if (cnd) is omitted, the default value is (7).

The least significant bit in either constant, is always zero (word addressing). See also table and note on page 6.0.1.

Effective branch:	Type	Function
	T8	k → P
	T2	lk → P

No branch:	Type	Function
	T8	(P) + 2 → P
	T2	(P) + 4 → P

Condition register: Unchanged



ABR*Absolute conditional branch to register***ABR**P851M
P852M
P856M
P857M

Syntax: [label]_ ABR [(cnd)] [*]_ r2

This instruction indicates that the address of next instruction to be executed is found either in the register specified by r2 or at the memory address indicated by the register or in normal sequence depending on (cnd) and the contents of the condition register.

If (cnd) = (7), the next instruction is at the effective memory address (unconditional branch).

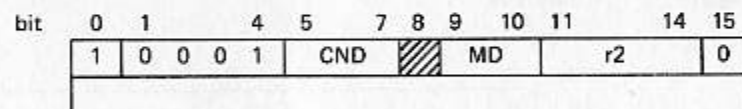
If (cnd) is omitted, the default value is (7).

See also table and note on page 6.0.1.

Effective branch:	Type	Function	MD	I/s	Syntax
	T1	(r2) → P	00	n.s.	ABR(cnd) r2
	T3	((r2)) → P	01	0	ABR(cnd)* r2

No branch:	Type	Function	MD	I/s
	T1	(P) + 2 → P	00	n.s.
	T3	(P) + 2 → P	01	0

Condition register: Unchanged

**ABI***Absolute branch indirect***ABI**P851M
P852M
P856M
P857M

Syntax: [label]_ ABI [(cnd)] [*]_ m[, r2]

The address of the next instruction to be executed is found either at the effective memory address or in the next instruction, depending on (cnd) and the contents of the condition register.

If (cnd) = (7) (see below) the instruction branched to is always at the effective memory address (unconditional branch).

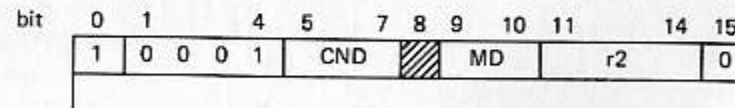
In all other cases the program must first fulfil a condition before the branch takes place. If (cnd) is omitted, the default value is (7).

See also table and note on page 6.0.1.

Effective branch:	Type	Function	MD	Syntax
	T4	(m) → P	10	ABI(cnd) m
	T5	(m + (r2)) → P	10	ABI(cnd) m, r2
	T6	((m)) → P	11	ABI(cnd)* m
	T7	((m + (r2))) → P	11	ABI(cnd)* m, r2

No branch:	Type	Function	MD
	T4	(P) + 4 → P	10
	T5	(P) + 4 → P	10
	T6	(P) + 4 → P	11
	T7	(P) + 4 → P	11

Condition register: Unchanged



RF*Relative forward conditional branch***RF**P851M
P852M
P856M
P857MSyntax: [label] \sqsubset RF [(cnd)] \sqsubset m

This instruction indicates that the next instruction to be executed is found either at the effective memory address or in normal sequence, depending on (cnd) and the contents of the condition register. If (cnd) = (7) the next instruction can be found at the effective memory address (unconditional relative branch).
If (cnd) is omitted, the default value of (7) is assumed.

The assembler calculates from the effective memory address, a displacement D relative (forwards) to the current value of the instruction counter (P). This value is stored in bits 8–15 of the instruction as a positive number. Thus its maximum is 255. In programming terms, this means that this instruction can only be used to branch by ≤ 128 words.

See also table and note on page 6.0.1.

Type	Function
T8	$(P) + 2 + D \rightarrow P$ (branch effective)
T8	$(P) + 2 \rightarrow P$ (no branch)

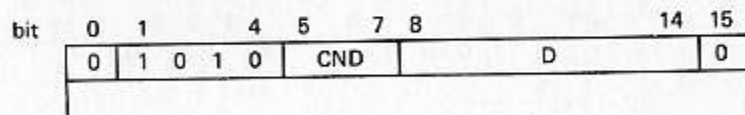
Example:

RF(Z) END

RF(3) $\ast\ast 12$

Condition register:

Unchanged

**RB***Relative backwards conditional branch***RB**P851M
P852M
P856M
P857MSyntax: [label] \sqsubset RB [(cnd)] \sqsubset m

This instruction means that the next instruction to be executed is found either at the effective memory address or in normal sequence, depending on (cnd) and the contents of the condition register. If (cnd) = (7) the next instruction to be executed is found at the effective memory address.
If (cnd) is omitted, the default value of (7) is assumed.

The assembler calculates from the effective memory address, a displacement D relative (backwards) to the current value of the instruction counter (P). This value is stored in bits 8–15 of the instruction as a positive number. Thus its maximum is 255. In programming terms, this means that this instruction can only be used to branch backwards by ≤ 128 words.

It should be noted that

 \sqsubset RB (cnd) \sqsubset *

is equivalent to branch to itself and causes a continuous loop.

See also table and note on page 6.0.1.

Type	Function
T8	$(P) + 2 - D \rightarrow P$ (branch effective)
T8	$(P) + 2 \rightarrow P$ (no branch)

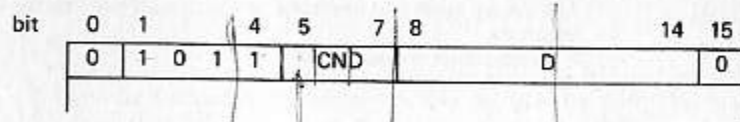
Example:

RB(4) LABEL

RB(NE) $\ast -2$

Condition register:

Unchanged



N.A

CF

Call function

CF

P851M
P852M
P856M
P857M

Syntax: [label] CF r1, lk

This instruction provides a link to a subroutine by storing successively the contents of the P-register and the program status word (PSW) in a memory stack. The PSW contains, amongst other things, the priority level and condition register. The stack pointer is held in the register specified by r1 and is automatically updated. Then a branch is made to the address specified by lk.

The subroutine must be terminated by an RTN instruction to branch back to the main program.

Type	Function
T2	(P) → (r1), (r1) - 2 → r1
	(PSW) → (r1), (r1) - 2 → r1
lk	→ P

Condition register:

Unchanged. Its contents shows the result of a previous operation and is stored in the memory stack for use on return from the subroutine.

bit	0	1	4	5	8	9	10	11	14	15	
	1	1	1	0	r1	0	1	0	0	0	1

Remark:

An interrupt 'stack overflow' is generated when r1 = A15 and the word address reached by the pointer = </100. Bit 13 is set in PSW.

- * r1 must be ≠ 0.
 - * Restricted to system mode if r1 = A15.
 - * The system stack and user stack are both built towards the lower addresses.
- P is stored first and next PSW.

CFR

Call function register

CFR

P851M
P852M
P856M
P857M

Syntax: [label] CFR[*] r1, r2

This instruction provides a link to a subroutine by storing successively the contents of the P-register, which points to the next instruction of the main program, and the contents of the program status word (PSW) in a memory stack. The PSW contains, amongst other things the priority level and the condition register. The stack pointer held in the register specified by r1 is automatically updated by decreasing the stack pointer by 2, as the stack pointer is filled from the higher address towards the lower address.

Next a branch is made to the effective memory address specified by the contents of a register specified by r2.

The subroutine must be terminated by an RTN instruction to branch back to the main program.

Type	Function	MD	Syntax
T1, T3	(P) → (r1), (r1) - 2 → r1		
	(PSW) → (r1), (r1) - 2 → r1		

then:

T1	(r2) → P	00	CFR r1, r2
T3	((r2)) → P	01	CFR* r1, r2

Condition register:

Unchanged. Its contents shows the result of a previous operation and is stored in the memory stack for use on return from the subroutine.

bit	0	1	4	5	8	9	10	11	14	15
	1	1	1	0	r1	MD		r2		1

Remark:

- * An interrupt 'stack overflow' is generated when r1 = A15 and the word address reached by the pointer = </100. Bit 13 in the PSW is set to 1.
- * r1 must be ≠ 0.
- * Restricted to system mode if r1 = A15.