A publication of

Philips Data Systems
S.S.S. Documentation
Apeldoorn, The Netherlands

Pub. No. 5122 991 27386

January 1982

Printed in the Netherlands

Preface

This volume is intended to be used together with the P800M publications concerning programming.

Part 1 describes in great detail the powerful instruction set for the P800M computers and shows the programmer the functional operation, the syntax, the setting of the condition register, the instruction time and examples.
The instructions are grouped in the following operational categories :

Load and store instructions
Arithmetic instructions
Logical instructions
Character handling instructions
Branch instructions
Shift instructions
Table handling instructions
External transfer instructions
Control instructions
I/O instructions
String instructions

Table of Contents

PART 1 INSTRUCTION SET

Key to symbols used in the instruction set

Label	Identifier, or label, consisting of max. 6 characters of which the first must always be a letter. All instructions, and most of the assembler directives, may be preceded by a label.
*	Asterisk. Indicates: – indirect addressing – current value of location counter
[ ]	The syntactic item(s) between these brackets may be omitted
{ }	Choose one of the items between these brackets
r1	Register A1 . . . A15
r2	Register A1 . . . A15. Used as an index register in memory reference instructions.
r3	Register A1 . . . A7
m	Memory expression
k	Constant in bits 8–15 (short constant)
lk	Constant or address in bits 0–15 of the word following the instruction (long constant)
P	P-register. (Instruction counter)
T1	Register to register operation.
T2	Long constant instruction.
T3	Register addressing.
T3A	Register r2 is not the stackpointer A15
T3B	Register r2 is the stackpointer A15
TxS	The result must be stored in memory
T4	Direct addressing
T5	Indexed addressing
T6	Indirect addressing
T7	Indirect indexed addressing
T8	Short constant instruction
l/s	Load/store indicator. Load: bit 15 = 0 Store: bit 15 = 1
MD	Addressing mode
∧	Logical AND
∨	Logical OR
∀	Exclusive OR
↔	Compare
/	Divide
x	Multiply
+	Add

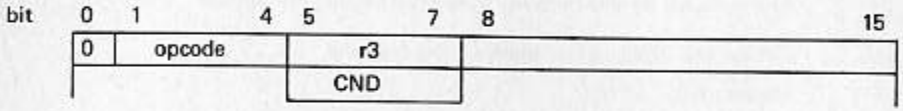
Instruction formats

Machine instructions conform to one of the following two formats:

- format 0
- format 1.

Format 0 instructions

Instructions of this type consist of one word, where the 16 bits indicate the following functions :

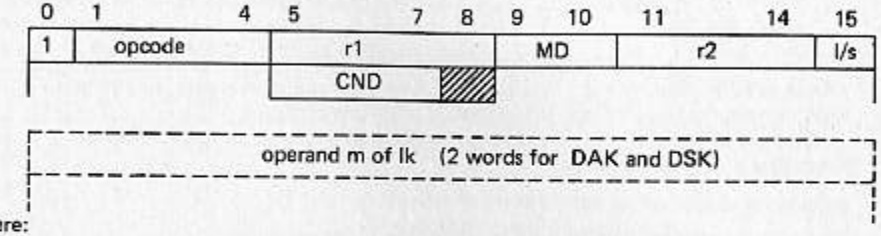


where

bit 0 – indicates the instruction format
bits 1–4 – operation code
bits 5–7 – one of the registers A1–A7 or the condition value in a Branch instruction.
bits 8–15 – the contents of this field varies according to the type of instruction and may contain one of the following values:
- an 8-bit *positive* constant (constant instruction)
- an even displacement value (branch instruction)
- an indication of the shift required (shift instruction)
- device address (I/O instruction) + function bits
- fixed parameters (miscellaneous instruction)

Format 1 instructions

Format 1 instructions perform a number of operations by reference to two of the 16 registers available for user access: one of these registers may point to a data item either in a word following the instruction or elsewhere in memory as it is possible to use that register as an index register.



where:

bit 0 – indicates the instruction format
bits 1–4 – operation code
bits 5–8 – one of the registers A1 . . . A15 specified as follows:
registers A1 . . . A7 are in group 0 and registers A8 . . . A15 are in group 1.
The group to which a register belongs is indicated by bit 8.
This may be either 0 (group 0) or 1 (group 1)
- in branch instruction, however, bits 5 to 7 inclusive indicate a condition value and bit 8 is not used.

bits 9–10 – addressing mode code. These bits will specify direct or indirect addressing, i.e. whether the word following the instruction, or another memory word, has to be taken into account.
bits 11–14 – the number of one of the 16 registers, expressed in the same way as in bits 5–8.
bit 15 – load/store indicator. Used in certain instructions to indicate that the result of the operation is to be placed either in the register shown by bits 5–8 (I/s = 0) or in a memory word (I/s = 1).

This type of instruction may be followed by a data word (16 bits) containing an address (m) or a positive or negative value. In the case of an address, bit 15 is not significant, except for character handling instructions.

The binary values of bits 5 through 8 in r1 and 11 through 14 for r2 are: 4 2 1 8, and in r3 4 2 1.

Example: A3 in r1 or r2 is written as 0110 and A12 as 1001. For r3 this is 011.
A12 cannot be specified in the field r3.

Registers

16 registers are available for use by the programmer. These 16 registers, which have the predefined symbols A0 through A15, are called the scratchpad. They may be addressed from either the instruction being carried out or from the toggle switches on the control panel.
The specific designation of registers within the scratchpad is:

P-register (A0)

This register is used to hold the address of the next instruction to be executed. It is incremented in steps of two if the program is to carry out in sequence, or altered to hold the required new address if a branch is to be performed.
The instruction counter (P) points always already to the next instruction before execution of an instruction.

Working registers (A1–A14)

The working registers may be used in any of the following ways:

- as accumulator where the data to be processed can be found in a register.
- as pointers where the contents of the specified registers contain the operand address rather than the operand itself.
- as index registers where the contents of the specified registers and the contents of the word following the instruction are summed to produce the operand address.

Register A15

This register is used by the interrupt system as the stackpointer and, as such, it is updated by the system whenever it is used for memory addressing.
It may be addressed by instruction in the same way as the registers A1 through A14.

Type of instruction

The instruction in the instruction set may use various methods of forming one of the operands to be used. To make a clear distinction between these methods, each instruction in the instruction set description has received a notation T1 thru T8 to indicate the manner in which the operand is formed. The latter is usually governed by the values of the format, address mode and the r2 field (bits 11 thru 14) in the instruction. The result of this operation may be an address which is called the effective memory address.

Type	Format	Mode	r2 field	Description
T1	1	00	≠ 0	Register to register operation
T2	1	01	0000	Long constant instruction
T3 (T3A) (T3B)	1	01	≠ 0	Address in register r2 (The register specified is not A15) (The register specified is A15)
T4	1	10	0000	Address in next word (direct addressing)
T5	1	10	≠ 0	Indexed addressing
T6	1	11	0000	Indirect addressing
T7	1	11	≠ 0	Indexed indirect addressing
T8	0	–	–	Short constant

T1 Register to register operation
The operand is the value in the register specified by r2.

T2 Long constant instruction
The operand is the value contained in the least significant word of the double length instruction.

T3 Address in register
The operand is held in memory. The memory address of the operand is the value in the register specified by r2.
T3A r2 ≠ A15
T3B r2 = A15

T4 Address in next word (direct addressing)
The operand is held in memory. The memory address of the operand is the value in the least significant word of the double length instruction.

T5 Indexed address in next word (indexed addressing)
The operand is held in memory. The memory address of the operand is found by adding the value in the register specified by r2 to the value in the least significant word of the double length instruction.

T6 Indirect address in next word (indirect addressing)
The operand is held in memory. The memory address of the operand is also held in memory. This indirect address is the value in the least significant word of the double length instruction.

T7 Indexed indirect address in next word (indexed indirect addressing)
The operand is held in memory. The memory address of the operand is also held in memory. This indirect address is found by adding the value in the register specified by r2 to the value in the least significant word of the double length instruction.

T8 Short constant instruction
The operand is the value in the least significant eight bits of the instruction.

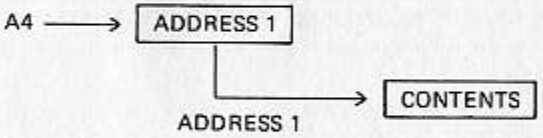
The notations T1, T2 etc. are followed by a description of the function of the instruction, which uses some of the symbols given in the list on page 1.1.1 and the symbols described here:

(.) contents of

e.g. (r3) the contents of the register specified in r3
(A15) the contents of register A15
(P) the contents of register P

((.)) the contents of the contents of

e.g. ((r3)) the contents of the contents of the register specified in r3.
((A4)) the contents of the contents of register A4.



The contents of A4 refers to address 1. This address contains the address of the actual information.

Example:

T4 (r1) → m Meaning: Direct addressing. The contents of register r1 is placed in the memory location with address m.

T6 (r1) − ((m)) → r1 Meaning: Indirect addressing. The contents of the register r1 is subtracted by the contents of the contents of address m. The result is placed in r1.

Software simulation of instructions

The following instructions may be software simulated on the P852M but are hardware present on the P851M, P856M and the P857M.

MS	MUK	DLL	DRN	DVR
MSR	MUR	DLC	DS	
ML	DA	DLN	DSK	
MLK	DAK	DRA	DSR	
MLR	DAR	DRC	DV	
MU	DLA	DRL	DVK	