

WESTINGHOUSE P-50 QUICK REFERENCE CARD

Processor Instruction Set & Architecture

Instruction Set

Arithmetic & Logical

M	C	F	μs	Operation
ADD	10	S	18.0	$A + (Y) \rightarrow A$
SUB	11	S	18.0	$A - (Y) \rightarrow A$
AND	12	S	18.0	$A \wedge (Y) \rightarrow A$
EOR	13	S	18.0	$A \oplus (Y) \rightarrow A$
DCR	01	S	13.5	$(Y) - 1 \rightarrow (Y)$

Bit / Sign Control

M	C	F	μs	Operation
SMB	03	S	13.5	set $(Y)_{13}$
CMB	02	S	13.5	clr $(Y)_{13}$

Shift

M	C	F	μs	Operation
LSH	14,15	S	18.0	$(Y) L1 \rightarrow (Y)$
RSH	16,17	S	18.0	$(Y) R1 \rightarrow (Y)$

Designator Control

M	C	F	μs	Operation
EDR	05	—	13.5	$(Y_{4-0}) \rightarrow DES$
SDR	06,07	K	13.5	$DES \rightarrow (Y_{4-0})$

Data Transfer / I-O

M	C	F	μs	Operation
ENL	32,33	S	18.0	$(Y) \rightarrow A$
STL	37	S	18.0	$A \rightarrow (Y)$
INT	30,31	S	18.0	$I/O \rightarrow A$
OUT	34,35	S	18.0	$A \rightarrow I/O$

Addressing Modes

Instr., direct:

13	12	11	10	9	8	7	6	5	4	3	2	1	0
Fn								0	Rel Addr				

Instr., indirect:

13	12	11	10	9	8	7	6	5	4	3	2	1	0
Fn								1	Rel Addr				

Operand addr (indirect target):

13	12	11	10	9	8	7	6	5	4	3	2	1	0
Operand Address													

I/O, direct:

13	12	11	10	9	8	7	6	5	4	3	2	1	0
Fn								0	—	Channel			

I/O, indirect:

13	12	11	10	9	8	7	6	5	4	3	2	1	0
Fn								1	Rel Addr				

I/O data word (indirect target):

13	12	11	10	9	8	7	6	5	4	3	2	1	0
—		Word						Channel					

Direct: operand in same 256-word block (P-reg. high bits + 8-bit relative address). Indirect: operand address = *contents* of that cell. Indirect adds 4.5 μs .

Control Transfer

M	C	F	μs	Operation
JMP	24	K	13.5	$Y \rightarrow P$
SLJ	22	K	13.5	$Y \rightarrow P$, lock
CLJ	23	K	13.5	$Y \rightarrow P$, unlock
ZJP	20	K	13.5/9.0	$Y \rightarrow P$ if Z
EJP	21	K	13.5/9.0	$Y \rightarrow P$ if EV
CJP	25	K	13.5/9.0	$Y \rightarrow P$ if EAC
OJP	26	K	13.5/9.0	$Y \rightarrow P$ if OVF
PJP	27	K	13.5/9.0	$Y \rightarrow P$ if POS
RJP	36	K	22.5	$(P) \rightarrow Y, Y \rightarrow P$

Machine Control

M	C	F	μs	Operation
STP	00,04	K	9.0	halt

Legend

M	Mnemonic
C	Octal code
F	Format class
S	SET — affects designators
K	Control transfer; designators unchanged unless noted
μs	Execution time (direct mode); indirect adds 4.5 μs

Opcode Map (octal)

	0	1	2	3	4	5	6	7
0	STP	DCR	CMB	SMB	STP	EDR	SDR	SDR
1	ADD	SUB	AND	EOR	LSH	LSH	RSH	RSH
2	ZJP	EJP	SLJ	CLJ	JMP	CJP	OJP	PJP
3	INT	INT	ENL	ENL	OUT	OUT	RJP	STL

Row = first octal digit, column = second.

Designator Bits (EDR / SDR)

Bit	Designator	Meaning when clear
4	Even	Odd
3	Zero	Non-zero
2	Positive	Negative
1	Overflow	No overflow
0	End-around carry (EAC)	No end-around carry

Notes & Quick Facts

- **Overflow** set only by DCR, ADD, SUB, LSH; only EDR clears it once set.
- **EAC** always set by AND; always cleared by CMB, SMB, EOR, INT, ENL, OUT, STL.
- Jump/control (K class) never changes designators. EDR loads all five at once from operand bits 0–4; SDR stores them the same way, unchanged.
- 14-bit word; opcode space fully populated, 00–37 octal.
- Direct-mode execution: 9–22.5 μs ; indirect adds 4.5 μs .
- After any P-register transfer (JMP, SLJ, CLJ, RJP, or a taken conditional jump), the instruction at operand address +1 executes next, and one interrupt is inhibited.

WESTINGHOUSE P-50 QUICK REFERENCE CARD

Assembler Pseudo-Operators & Input Syntax, Editor Support

Source Line Columns		Directives		
Col.	Field	Dir.	Operand	Effect
1–6	Label (optional, max 6 chars)	ORG	expr	Set location counter
7	Modifier: blank = decimal, 8 = octal, for this line's bare numeric literal	EQU	expr	Define symbol (decimal default)
8–10	Mnemonic (opcode or directive), exactly 3 chars	SYN	expr	Symbol synonym; same as EQU
11	Space	OCT	expr[,expr...]	Store word(s), always octal
12+	Operand (one token); everything after it is remarks	DEC	expr[,expr...]	Store word(s), always decimal
		WRD	expr	Same as OCT; used for pool words
		GEN	—	Flush current block's literal pool
		VFD	w1,w2,... v1,v2,...	Declare/pack a bit-field template
		RPT	n	Repeat next line <i>n</i> times (decimal; <i>n</i> =0 discards it)
		BCD	c text	Pack <i>c</i> words of zoned-BCD text, 2 chars/word
		DLE	—	Annotation only; no effect
		EJE	—	Listing: page eject
		UNL	—	Listing: suppress detail
		HED	text	Listing: print heading
		END	—	End of source program

No comment delimiter exists. Column 12 blank ⇒ no operand, and remarks (if any) begin at column 12. OCT, DEC, and BCD ignore the modifier column; VFD fields are always octal.

Operand / Expression Syntax

- Expression = numbers, symbols, L (this instruction's own address), combined with + - * (standard precedence).
- Leading - (first token, e.g. -0, -1) = ones'-complement negate; - between two operands (e.g. L-1) = ordinary subtract.
- ** = intentionally blank field, patched at install time (e.g. an I/O channel).
- EXPR = literal-pool reference: assembler allocates/reuses a cell holding EXPR, uses its address as the operand. Reuse matches expression *text*, not value.
- ,I suffix = indirect addressing; combine with a pool reference as EXPR,I.

Predefined Symbols

Sym.	Addr.	Meaning
ACC	101 (65 dec.)	The accumulator's memory cell
PC	0	Address 0

Outside their own 256-word block, reference either indirectly: ACC,I, PC,I.

Invoking the Assembler

p50asm [OPTIONS] SOURCE.p50

Two-pass assembler; always writes SOURCE.lst. Errors print as prog:source:line: error: msg. No tape/HEX is written if either pass reports an error.

Option	Effect
-o,	Write P-50 binary tape to <i>P</i> (default; else .p50tap)
--output= <i>P</i>	
--hex[= <i>P</i>]	Write Intel HEX (default .hex)
-e,	End tape with transfer to symbol/addr instead of load-stop (<i>A</i> always octal)
--entry= <i>S</i> <i>A</i>	
-n, --check	Syntax check only; errors to std-out; no files written
-v, --verbose	Print version, files written, statistics
-h, --help	Print usage and exit

Exit status: 0 clean, 1 assembly error, 2 usage error.

Macros

MAC opens a definition (labelled with the macro's name, up to 6 chars); TER closes it. Parameters are named on the MAC line and substituted throughout the body.

Definition:

```
NAME MAC P1,P2,...  
...body...  
TER
```

Invocation (looks like an instruction; no spaces around commas):

```
NAME A1,A2,...
```

The invoked name shares the 3-character mnemonic column, so only the first 3 characters of a macro's name are significant at the call site. A label on the invocation attaches to the expansion's first instruction. No local scoping: a body label is shared by every invocation.

VFD Templates (detail)

NAME VFD w1,w2,... declares NAME as a template of fields (bit widths, most-significant first). A later line NAME v1,v2,... packs those values, msb-first, into one word. A blank field (v1,,v3) packs as zero. Widths summing past 14 bits truncate silently. Must be declared before use — no forward reference.

Editor Support

- Emacs** (p50asm-mode.el): syntax highlighting; TAB/S-TAB jumps between fixed-column fields; C-c C-a/C-c C-r/C-c C-c realign the line/region/buffer; imenu support for labels.
- Vim** (ftdetect/syntax/ftplugin): same highlighting; sets colorcolumn=7,11,31,73 and expandtab automatically for *.p50; Tab/ S-Tab field-jump; :P50Align, :P50AlignBuffer.
- Kate**/KTextEditor (p50asm.xml): highlighting only (label, 8 modifier, opcodes, directives, comments, literals), registered as "P50 Assembler"; no column ruler or realign command — use Emacs or Vim for those, then reopen in Kate.