

Computer Systems

Sixth Edition
Documentation for Exam Handouts

J. Stanley Warford
Matthew McRaven
October 23, 2025

Decimal Multiples	Decimal Prefix	Decimal Prefix Letter	Binary Multiples	Binary Prefix	Binary Prefix Letter	Hex	Bin
$10^3 = 1000$	kilo-	K	$2^{10} = 1024$	kibi-	Ki	0	0000
$10^6 = 1000^2$	mega-	M	$2^{20} = 1024^2$	mebi-	Mi	1	0001
$10^9 = 1000^3$	giga-	G	$2^{30} = 1024^3$	gibi-	Gi	2	0010
$10^{12} = 1000^4$	tera-	T	$2^{40} = 1024^4$	tebi-	Ti	3	0011
$10^{15} = 1000^5$	peta-	P	$2^{50} = 1024^5$	pebi-	Pi	4	0100
						5	0101
						6	0110
						7	0111
						8	1000
						9	1001
						A	1010
						B	1011
						C	1100
						D	1101
						E	1110
						F	1111

Decimal Multiples	Binary Multiples	Percent difference
$10^3 = 1000$	$2^{10} = 1024$	2.4%
$10^6 = 1,000,000$	$2^{20} = 1,048,576$	4.9%
$10^9 = 1,000,000,000$	$2^{30} = 1,073,741,824$	7.4%
$10^{12} = 1,000,000,000,000$	$2^{40} = 1,099,511,627,776$	10.0%
$10^{15} = 1,000,000,000,000,000$	$2^{50} = 1,125,899,906,842,624$	12.6%

The system performance equation for computing the time it takes to execute a program is

$$\frac{\text{time}}{\text{program}} = \frac{\text{instructions}}{\text{program}} \times \frac{\text{cycles}}{\text{instruction}} \times \frac{\text{time}}{\text{cycle}}$$

where $\frac{\text{cycles}}{\text{instruction}}$ is the average number of Mc2 instructions in an ISA3 instruction, and $\frac{\text{time}}{\text{cycle}}$ is the period T , the time it takes to execute one von Neumann cycle of a Mc2 instruction. The speed of a CPU is specified by its frequency rating f measured in Hertz, where one Hertz (Hz) is one cycle/second.

$$T = \frac{1}{f}$$

The bandwidth equation for computing how much information is transmitted from the source to the destination is

$$\text{information} = \frac{\text{information}}{\text{time}} \times \text{time}$$

where $\frac{\text{information}}{\text{time}}$ is the bandwidth of the channel measured in bits per second.

The total number of bits it takes to store an image is

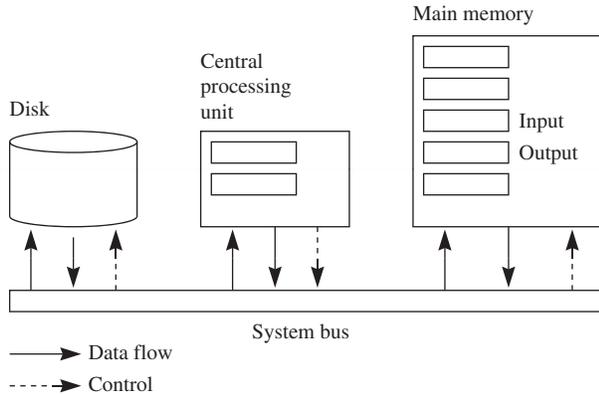
$$\text{number of bits} = \frac{\text{bits}}{\text{pixel}} \times \text{number of pixels}$$

where $\frac{\text{bits}}{\text{pixel}}$ is the bit depth of the display, which depends on whether the display is black and white, grayscale, or color. For black and white displays, the bit depth is 1. For grayscale displays, the bit depth depends on the number of shades of gray. For color displays, the bit depth depends on the number of shades of each color in one RGB pixel.

Char	Bin	Hex									
NUL	0000 0000	00	SP	0010 0000	20	@	0100 0000	40	'	0110 0000	60
SOH	0000 0001	01	!	0010 0001	21	A	0100 0001	41	a	0110 0001	61
STX	0000 0010	02	"	0010 0010	22	B	0100 0010	42	b	0110 0010	62
ETX	0000 0011	03	#	0010 0011	23	C	0100 0011	43	c	0110 0011	63
EOT	0000 0100	04	\$	0010 0100	24	D	0100 0100	44	d	0110 0100	64
ENQ	0000 0101	05	%	0010 0101	25	E	0100 0101	45	e	0110 0101	65
ACK	0000 0110	06	&	0010 0110	26	F	0100 0110	46	f	0110 0110	66
BEL	0000 0111	07	'	0010 0111	27	G	0100 0111	47	g	0110 0111	67
BS	0000 1000	08	(0010 1000	28	H	0100 1000	48	h	0110 1000	68
HT	0000 1001	09)	0010 1001	29	I	0100 1001	49	i	0110 1001	69
LF	0000 1010	0A	*	0010 1010	2A	J	0100 1010	4A	j	0110 1010	6A
VT	0000 1011	0B	+	0010 1011	2B	K	0100 1011	4B	k	0110 1011	6B
FF	0000 1100	0C	,	0010 1100	2C	L	0100 1100	4C	l	0110 1100	6C
CR	0000 1101	0D	-	0010 1101	2D	M	0100 1101	4D	m	0110 1101	6D
SO	0000 1110	0E	.	0010 1110	2E	N	0100 1110	4E	n	0110 1110	6E
SI	0000 1111	0F	/	0010 1111	2F	O	0100 1111	4F	o	0110 1111	6F
DLE	0001 0000	10	0	0011 0000	30	P	0101 0000	50	p	0111 0000	70
DC1	0001 0001	11	1	0011 0001	31	Q	0101 0001	51	q	0111 0001	71
DC2	0001 0010	12	2	0011 0010	32	R	0101 0010	52	r	0111 0010	72
DC3	0001 0011	13	3	0011 0011	33	S	0101 0011	53	s	0111 0011	73
DC4	0001 0100	14	4	0011 0100	34	T	0101 0100	54	t	0111 0100	74
NAK	0001 0101	15	5	0011 0101	35	U	0101 0101	55	u	0111 0101	75
SYN	0001 0110	16	6	0011 0110	36	V	0101 0110	56	v	0111 0110	76
ETB	0001 0111	17	7	0011 0111	37	W	0101 0111	57	w	0111 0111	77
CAN	0001 1000	18	8	0011 1000	38	X	0101 1000	58	x	0111 1000	78
EM	0001 1001	19	9	0011 1001	39	Y	0101 1001	59	y	0111 1001	79
SUB	0001 1010	1A	:	0011 1010	3A	Z	0101 1010	5A	z	0111 1010	7A
ESC	0001 1011	1B	;	0011 1011	3B	[0101 1011	5B	{	0111 1011	7B
FS	0001 1100	1C	<	0011 1100	3C	\	0101 1100	5C		0111 1100	7C
GS	0001 1101	1D	=	0011 1101	3D]	0101 1101	5D	}	0111 1101	7D
RS	0001 1110	1E	>	0011 1110	3E	^	0101 1110	5E	~	0111 1110	7E
US	0001 1111	1F	?	0011 1111	3F	_	0101 1111	5F	DEL	0111 1111	7F

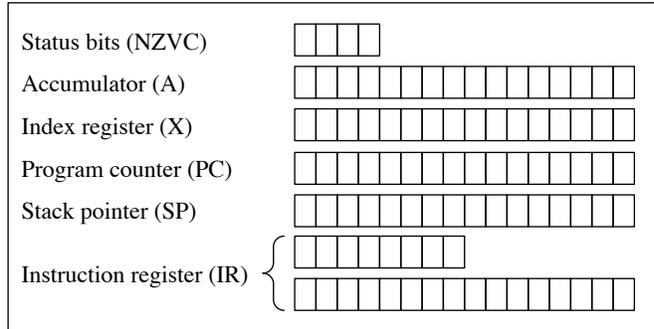
Abbreviations for Control Characters

NUL	null, or all zeros	FF	form feed	CAN	cancel
SOH	start of heading	CR	carriage return	EM	end of medium
STX	start of text	SO	shift out	SUB	substitute
ETX	end of text	SI	shift in	ESC	escape
EOT	end of transmission	DLE	data link escape	FS	file separator
ENQ	enquiry	DC1	device control 1	GS	group separator
ACK	acknowledge	DC2	device control 2	RS	record separator
BEL	bell	DC3	device control 3	US	unit separator
BS	backspace	DC4	device control 4	SP	space
HT	horizontal tabulation	NAK	negative acknowledge	DEL	delete
LF	line feed	SYN	synchronous idle		
VT	vertical tabulation	ETB	end of transmission block		

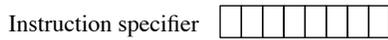


Block diagram of the Pep/10 computer.

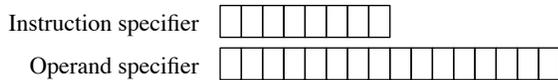
Central processing unit (CPU)



The CPU of the Pep/10 computer.



A monadic instruction.



The two parts of a dyadic instruction.

Load program into memory at Mem[0000]

PC ← 0000

do

Fetch: IR[0:7] ← Mem[PC]

Decode: Decode instruction specifier IR[0:7]

Increment: PC ← PC + 1

if IR[0:7] is a dyadic instruction

IR[8:23] ← Mem[PC]

PC ← PC + 2

Execute: Execute the instruction in the IR

while not shut down && instruction in IR is legal

The program execution process.

a	Addressing Mode
0	Immediate
1	Indexed

The addressing-a field.

r	Register
0	Accumulator (A)
1	Index register (X)

The register-r field.

Addressing Mode	aaa	Letter	Operand
Immediate	000	i	OprndSpec
Direct	001	d	Mem[OprndSpec]
Indirect	010	n	Mem[Mem[OprndSpec]]
Stack-relative	011	s	Mem[SP + OprndSpec]
Stack-relative deferred	100	sf	Mem[Mem[SP + OprndSpec]]
Indexed	101	x	Mem[OprndSpec + X]
Stack-indexed	110	sx	Mem[SP + OprndSpec + X]
Stack-deferred indexed	111	sfx	Mem[Mem[SP + OprndSpec] + X]

The addressing-aaa field.

Operation	Macro	Macro Expansion
Character input	@CHARI OprndSpec, AddrMode	LDWA charIn, d STBA OprndSpec, AddrMode
Character output	@CHARO OprndSpec, AddrMode	LDWA OprndSpec, AddrMode STBA charOut, d

The @CHARI and @CHARO macro expansions.

Symbol	Value		
DECI	0000	FB3E	NZVC
DECO	0001	FB3F	A
HEXO	0002	FB41	X
STRO	0003	FB43	PC
SNOP	0004	FB45	SP
		FB47	IR

Symbol table for the SCALL system calls.

The PCB for a system call

Operation	Macro	Macro Expansion
Decimal input	@DECI OprndSpec, AddrMode	LDWA DECI, i SCALL OprndSpec, AddrMode
Decimal output	@DECO OprndSpec, AddrMode	LDWA DECO, i SCALL OprndSpec, AddrMode
Hexadecimal output	@HEXO OprndSpec, AddrMode	LDWA DECI, i SCALL OprndSpec, AddrMode
String output	@STRO OprndSpec, AddrMode	LDWA STRO, i SCALL OprndSpec, AddrMode
System call no operation	@SNOP OprndSpec, AddrMode	LDWA SNOP, i SCALL OprndSpec, AddrMode

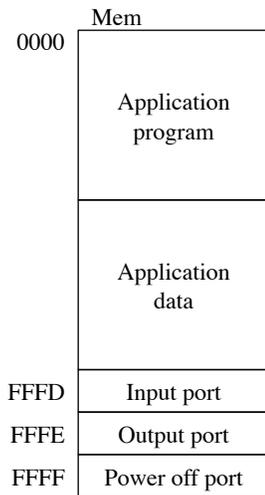
The system call macro expansions.

#1u	One-byte unsigned decimal	#2u	Two-byte unsigned decimal
#1d	One-byte signed decimal	#2d	Two-byte signed decimal
#1h	One-byte hexadecimal	#2h	Two-byte hexadecimal
#1c	One-byte character		

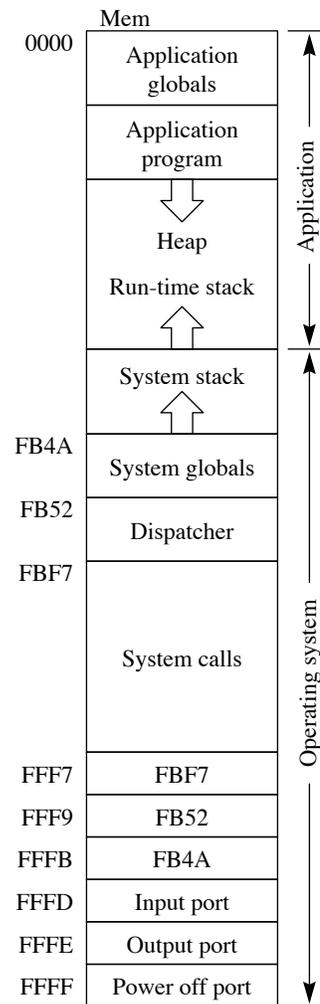
Format trace tags

Instruction Specifier	Mnemonic	Instruction	Addressing Modes	Status Bits
0000 0000		Illegal instruction		
0000 0001	RET	Return from CALL	Monadic	
0000 0010	SRET	Return from system CALL	Monadic	
0000 0011	MOVFLGA	Move NZVC flags to A[12 : 15]	Monadic	
0000 0100	MOVAFLG	Move A[12 : 15] to NZVC flags	Monadic	NZVC
0000 0101	MOVSPA	Move SP to A	Monadic	
0000 0110	MOVASP	Move A to SP	Monadic	
0000 0111	NOP	No operation	Monadic	
0001 100r	NEGr	Negate r	Monadic	NZVC
0001 101r	ASLr	Arithmetic shift left r	Monadic	NZVC
0001 110r	ASRr	Arithmetic shift right r	Monadic	NZVC
0001 111r	NOTr	Bitwise NOT r	Monadic	NZ
0010 000r	ROLr	Rotate left r	Monadic	NZC
0010 001r	RORr	Rotate right r	Monadic	NZC
0010 010a	BR	Branch unconditional	i, x	
0010 011a	BRLE	Branch if less than or equal to	i, x	
0010 100a	BRLT	Branch if less than	i, x	
0010 101a	BREQ	Branch if equal to	i, x	
0010 110a	BRNE	Branch if not equal to	i, x	
0010 111a	BRGE	Branch if greater than or equal to	i, x	
0011 000a	BRGT	Branch if greater than	i, x	
0011 001a	BRV	Branch if V	i, x	
0011 010a	BRC	Branch if C	i, x	
0011 011a	CALL	Call subroutine	i, x	
0011 1aaa	SCALL	System call	i, d, n, s, sf, x, sx, sfx	
0100 0aaa	ADDSP	Add to SP	i, d, n, s, sf, x, sx, sfx	
0100 1aaa	SUBSP	Subtract from SP	i, d, n, s, sf, x, sx, sfx	
0101 raaa	ADDr	Add to r	i, d, n, s, sf, x, sx, sfx	NZVC
0110 raaa	SUBr	Subtract from r	i, d, n, s, sf, x, sx, sfx	NZVC
0111 raaa	ANDr	Bitwise AND to r	i, d, n, s, sf, x, sx, sfx	NZ
1000 raaa	ORr	Bitwise OR to r	i, d, n, s, sf, x, sx, sfx	NZ
1001 raaa	XORr	Bitwise XOR to r	i, d, n, s, sf, x, sx, sfx	NZ
1010 raaa	CPWr	Compare word to r	i, d, n, s, sf, x, sx, sfx	NZVC
1011 raaa	CPBr	Compare byte to r[8 : 15]	i, d, n, s, sf, x, sx, sfx	NZVC
1100 raaa	LDWr	Load word r from memory	i, d, n, s, sf, x, sx, sfx	NZ
1101 raaa	LDBr	Load byte r[8 : 15] from memory	i, d, n, s, sf, x, sx, sfx	NZ
1110 raaa	STWr	Store word r to memory	d, n, s, sf, x, sx, sfx	
1111 raaa	STBr	Store byte r[8 : 15] to memory	d, n, s, sf, x, sx, sfx	

Instruction	Register transfer language specification
RET	$PC \leftarrow Mem[SP]; SP \leftarrow SP + 2$
SRET	$T \leftarrow SP; NZVC \leftarrow Mem[T][4:7]; A \leftarrow Mem[T+1]; X \leftarrow Mem[T+3];$ $PC \leftarrow Mem[T+5]; SP \leftarrow Mem[T+7]; Mem[FFFB] \leftarrow T+12$
MOVFLGA	$A[8:11] \leftarrow 0, A[12:15] \leftarrow NZVC$
MOVAFLG	$NZVC \leftarrow A[12:15]$
MOVSPA	$A \leftarrow SP$
MOVASP	$SP \leftarrow A$
NOF	{No operation}
NEGr	$r \leftarrow \neg r; N \leftarrow r < 0, Z \leftarrow r = 0, V \leftarrow \{overflow\}, C \leftarrow \{carry\}$
ASLr	$C \leftarrow r[0], r[0:14] \leftarrow r[1:15], r[15] \leftarrow 0; N \leftarrow r < 0, Z \leftarrow r = 0, V \leftarrow \{overflow\}$
ASRr	$C \leftarrow r[15], r[1:15] \leftarrow r[0:14]; N \leftarrow r < 0, Z \leftarrow r = 0, V \leftarrow 0$
NOTr	$r \leftarrow \neg r; N \leftarrow r < 0, Z \leftarrow r = 0$
ROLr	$C \leftarrow r[0], r[0:14] \leftarrow r[1:15], r[15] \leftarrow C; N \leftarrow r < 0, Z \leftarrow r = 0$
RORr	$C \leftarrow r[15], r[1:15] \leftarrow r[0:14], r[0] \leftarrow C; N \leftarrow r < 0, Z \leftarrow r = 0$
BR	$PC \leftarrow Oprnd$
BRL	$N = 1 \vee Z = 1 \Rightarrow PC \leftarrow Oprnd$
BRLT	$N = 1 \Rightarrow PC \leftarrow Oprnd$
BREQ	$Z = 1 \Rightarrow PC \leftarrow Oprnd$
BRNE	$Z = 0 \Rightarrow PC \leftarrow Oprnd$
BRGE	$N = 0 \Rightarrow PC \leftarrow Oprnd$
BRGT	$N = 0 \wedge Z = 0 \Rightarrow PC \leftarrow Oprnd$
BRV	$V = 1 \Rightarrow PC \leftarrow Oprnd$
BRC	$C = 1 \Rightarrow PC \leftarrow Oprnd$
CALL	$SP \leftarrow SP - 2; Mem[SP] \leftarrow PC; PC \leftarrow Oprnd$
SCALL	$T \leftarrow Mem[FFFB]; Mem[T-3] \leftarrow IR; Mem[T-5] \leftarrow SP; Mem[T-7] \leftarrow PC;$ $Mem[T-9] \leftarrow X; Mem[T-11] \leftarrow A; Mem[T-12][4:7] \leftarrow NZVC;$ $SP \leftarrow T - 12; PC \leftarrow Mem[FFFE]$
ADDSP	$SP \leftarrow SP + Oprnd$
SUBSP	$SP \leftarrow SP - Oprnd$
ADDr	$r \leftarrow r + Oprnd; N \leftarrow r < 0, Z \leftarrow r = 0, V \leftarrow \{overflow\}, C \leftarrow \{carry\}$
SUBr	$r \leftarrow r - Oprnd; N \leftarrow r < 0, Z \leftarrow r = 0, V \leftarrow \{overflow\}, C \leftarrow \{carry\}$
ANDr	$r \leftarrow r \wedge Oprnd; N \leftarrow r < 0, Z \leftarrow r = 0$
ORr	$r \leftarrow r \vee Oprnd; N \leftarrow r < 0, Z \leftarrow r = 0$
XORr	$r \leftarrow r \oplus Oprnd; N \leftarrow r < 0, Z \leftarrow r = 0$
CPWr	$T \leftarrow r - Oprnd; N \leftarrow T < 0, Z \leftarrow T = 0, V \leftarrow \{overflow\}, C \leftarrow \{carry\}; N \leftarrow N \oplus V$
CPBr	$T \leftarrow r[8:15] - \text{byte } Oprnd; N \leftarrow T < 0, Z \leftarrow T = 0, V \leftarrow 0, C \leftarrow 0$
LDWr	$r \leftarrow Oprnd; N \leftarrow r < 0, Z \leftarrow r = 0$
LDBr	$r[0:7] \leftarrow 0, r[8:15] \leftarrow \text{byte } Oprnd; N \leftarrow 0, Z \leftarrow r[8:15] = 0$
STWr	$Oprnd \leftarrow r$
STBr	$\text{byte } Oprnd \leftarrow r[8:15]$



The Pep/10 memory map in bare metal mode.



The Pep/10 memory map with an operating system.