Sonne CPU Instruction and Opcode Reference

Mnemonic	Opcode (hexadecimal)	Operation (Low = low order byte of program counter, High = high order byte of program counter, xMx is an implied 16-bit memory address. Pseudo register N stands for the next opcode in the instruction stream, essentially a code literal.)
Group 1 — “PAIR" Instructions (Opcode bit 7 = 0)		
Subgroups “COPY”, “SCROUNGE”, “SIGNAL”, “DIAL” (See documentation for binary selectors.)		
NOP	00	Do nothing. Increment Low by 1.
RET	01	Copy H into High, copy R into Low, set O to zero.
NH	02	Copy next opcode into H, increment Low by 2.
NG	03	Copy next opcode into G, increment Low by 2.
NL	04	Copy next opcode into L, increment Low by 2.
NS	05	Copy next opcode into SOR, increment Low by 2.
NP	06	Copy next opcode into POR. Set POR to low impedance. Increment Low by 2.
NR	07	Copy next opcode into R,set O to 0, increment Low by 2.
ND	08	Copy next opcode into D, increment Low by 2.
NJ	09	Copy next opcode into Low.
NT	0A	If R plus O is not zero, then copy next opcode into Low, else increment Low by 1.
NF	0B	If R plus O is zero, then copy next opcode into Low, else increment Low by 1.
NC	0C	Copy High into H, copy Low into R, copy next opcode into H, set Low to zero, set O to
R0P	0D	Set O to 0. Increment Low by 1.
NA	0E	Copy A into Clip, copy next opcode into A, increment Low by 2. If bit 7 of A is zero then set mXm to 256*H+A, else if bit 6 of A is zero then set xMx to 256*G+A, else set xMx to 256*L+A.
NB	0F	Copy B into Clip, copy next opcode into A, increment Low by 2. If bit 7 of B is zero then set mXm to 256*H+B, else if bit 6 of B is zero then set xMx to 256*G+B, else set xMx to 256*L+B.
ISI	10	Shift SIR left by one position, replace least significant digit by bit on MISO line.
LID	11	Increment High by 1, store High into H, set Low to zero.
MH	12	Copy value at memory address xMx into H. Increment Low by 1.
MG	13	Copy value at memory address xMx into G. Increment Low by 1.
ML	14	Copy value at memory address xMx into L. Increment Low by 1.
MS	15	Copy value at memory address xMx into SOR. Increment Low by 1.
MP	16	Copy value at memory address xMx into POR. Set POR to low impedance. Increment Low by 1.
MR	17	Copy value at memory address xMx into R, set O to 0. Increment Low by 1.
MD	18	Copy value at memory address xMx into D. Increment Low by 1.
MJ	19	Copy value at memory address xMx into Low.
MT	1A	If R plus O is not zero, then copy value at memory address xMx into Low, else increme
MF	1B	If R plus O is zero, then copy value at memory address xMx into Low, else increment L
MC	1C	Copy High into H, copy Low into R, copy value at memory address xMx into H, set Low t
R1P	1D	Set O to 1. Increment Low by 1.
MA	1E	Copy A into Clip, copy value at memory address xMx into A. Increment Low by 1. If bit 7 of A is zero then set mXm to 256*H+A, else if bit 6 of A is zero then set xMx to 256*G+A, else set xMx to 256*L+A.

Mnemonic	Opcode (hexadecimal)	Operation (Low = low order byte of program counter, High = high order byte of program counter, xMx is an implied 16-bit memory address. Pseudo register N stands for the next opcode in the instruction stream, essentially a code literal.)
MB	1F	Copy B into Clip, copy value at memory address xMx into B. Increment Low by 1. If bit 7 of B is zero then set mXm to 256*H+B, else if bit 6 of B is zero then set xMx to 256*G+B, else set xMx to 256*L+B.
OSO	20	Set bit on MOSI line to most significant digit of SOR, shift SOR left by one position.
HM	21	Copy H to memory at address xMx. Increment Low by 1.
REA	22	Copy Clip to A, set xMx. Increment Low by 1.
HG	23	Copy H into G. Increment Low by 1.
HL	24	Copy H into L. Increment Low by 1.
HS	25	Copy H into SOR. Increment Low by 1.
HP	26	Copy H into POR. Set POR to low impedance. Increment Low by 1.
HR	27	Copy H into R, set O to 0. Increment Low by 1.
HD	28	Copy H into D. Increment Low by 1.
HJ	29	Copy H into Low.
HT	2A	If R plus O is not zero, then copy H into Low, else increment Low by 1.
HF	2B	If R plus O is zero, then copy H into Low, else increment Low by 1.
HC	2C	Copy H into High, copy Low into R, set Low to zero, set O to zero.
R2P	2D	Set O to 2. Increment Low by 1.
HA	2E	Copy A into Clip, copy H into A. Increment Low by 1. If bit 7 of A is zero then set mXm to 256*H+A, else if bit 6 of A is zero then set xMx to 256*G+A, else set xMx to 256*L+A.
HB	2F	Copy B into Clip, copy H into B. Increment Low by 1. If bit 7 of B is zero then set mXm to 256*H+B, else if bit 6 of B is zero then set xMx to 256*G+B, else set xMx to 256*L+B.
SCL	30	Set SPI clock line low. Increment Low by 1.
GM	31	Copy G into memory at address xMx. Increment Low by 1.
GH	32	Copy G into H. Increment Low by 1.
REB	33	Copy Clip to B, set xMx. Increment Low by 1.
GL	34	Copy G into L. Increment Low by 1.
GS	35	Copy G into SOR. Increment Low by 1.
GP	36	Copy G into POR. Set POR to low impedance. Increment Low by 1.
GR	37	Copy G into R, set O to 0. Increment Low by 1.
GD	38	Copy G into D. Increment Low by 1.
GJ	39	Copy G into Low.
GT	3A	If R plus O is not zero, then copy G into Low, else increment Low by 1.
GF	3B	If R plus O is zero, then copy G into Low, else increment Low by 1.
GC	3C	Copy High into H, copy G into High, copy Low into R, set Low to zero, set O to zero.
R3P	3D	Set O to 3. Increment Low by 1.
GA	3E	Copy A into Clip, copy H into A. Increment Low by 1. If bit 7 of A is zero then set mXm to 256*H+A, else if bit 6 of A is zero then set xMx to 256*G+A, else set xMx to 256*L+A.
GB	3F	Copy B into Clip, copy H into B. Increment Low by 1. If bit 7 of B is zero then set mXm to 256*H+B, else if bit 6 of B is zero then set xMx to 256*G+B, else set xMx to 256*L+B.

Mnemonic	Opcode (hexadecimal)	Operation (Low = low order byte of program counter, High = high order byte of program counter, xMx is an implied 16-bit memory address. Pseudo register N stands for the next opcode in the instruction stream, essentially a code literal.)
SCH	40	Set SPI clock line high. Increment Low by 1.
LM	41	Copy L into memory at address xMx. Increment Low by 1.
LH	42	Copy L into H. Increment Low by 1.
LG	43	Copy L into G. Increment Low by 1.
LL	44	Do nothing. Increment Low by 1.
LS	45	Copy L into SOR. Increment Low by 1.
LP	46	Copy L into POR. Set POR to low impedance. Increment Low by 1.
LR	47	Copy L into R, set O to 0. Increment Low by 1.
LD	48	Copy L into D. Increment Low by 1.
LJ	49	Copy L into Low.
LT	4A	If R plus O is not zero, then copy L into Low, else increment Low by 1.
LF	4B	If R plus O is zero, then copy G into Low, else increment Low by 1.
LC	4C	Copy High into H, copy L into High, copy Low into R, set Low to zero, set O to zero.
R4M	4D	Set O to -4. Increment Low by 1.
LA	4E	Copy A into Clip, copy L into B. Increment Low by 1. If bit 7 of A is zero then set mXm to 256*H+A, else if bit 6 of A is zero then set xMx to 256*G+A, else set xMx to 256*L+A.
LB	4F	Copy B into Clip, copy L into B. Increment Low by 1. If bit 7 of B is zero then set mXm to 256*H+B, else if bit 6 of B is zero then set xMx to 256*G+B, else set xMx to 256*L+B.
HIZ	50	Set POR to tristate. Increment Low by 1.
SM	51	Copy SIR to memory at address xMx. Increment Low by 1.
SH	52	Copy SIR to H. Increment Low by 1.
SG	53	Copy SIR to G. Increment Low by 1.
SL	54	Copy SIR to L. Increment Low by 1.
SS	55	Copy SIR to SOR. Increment Low by 1.
SP	56	Copy SIR to POR. Set POR to low impedance. Increment Low by 1.
SR	57	Copy SIR to R. Increment Low by 1.
SD	58	Copy SIR to D. Increment Low by 1.
SJ	59	Copy SIR to Low.
ST	5A	If R plus O is not zero, then copy SIR into Low, else increment Low by 1.
SF	5B	If R plus O is zero, then copy SIR into Low, else increment Low by 1.
SC	5C	Copy High into H, copy SIR into High, copy Low into R, set Low to zero, set O to zero
R3M	5D	Set O to -3. Increment Low by 1.
SA	5E	Copy A into Clip, copy SIR into A. Increment Low by 1. If bit 7 of A is zero then set mXm to 256*H+A, else if bit 6 of A is zero then set xMx to 256*G+A, else set xMx to 256*L+A.
SB	5F	Copy B into Clip, copy SIR into B. Increment Low by 1. If bit 7 of B is zero then set mXm to 256*H+B, else if bit 6 of B is zero then set xMx to 256*G+B, else set xMx to 256*L+B.
LLF	60	Copy value at memory address 256*L+0xC4 into R, set O to 0, increment L by one. Incre
PM	61	Copy PIR to memory at address xMx. Increment Low by 1.
PH	62	Copy PIR to H. Increment Low by 1.

Mnemonic	Opcode (hexadecimal)	Operation (Low = low order byte of program counter, High = high order byte of program counter, xMx is an implied 16-bit memory address. Pseudo register N stands for the next opcode in the instruction stream, essentially a code literal.)
PG	63	Copy PIR to G. Increment Low by 1.
PL	64	Copy PIR to L. Increment Low by 1.
PS	65	Copy PIR to SOR. Increment Low by 1.
PP	66	Do nothing. Increment Low by 1.
PR	67	Copy PIR to R, set O to 0. Increment Low by 1.
PD	68	Copy PIR to D. Increment Low by 1.
PJ	69	Copy PIR to Low.
PT	6A	If R plus O is not zero, then copy PIR into Low, else increment Low by 1.
PF	6B	If R plus O is zero, then copy PIR into Low, else increment Low by 1.
PC	6C	Copy High into H, copy PIR into High, copy Low into R, set Low to zero, set O to zero
R2M	6D	Set O to -2. Increment Low by 1.
PA	6E	Copy A into Clip, copy PIR into A. Increment Low by 1. If bit 7 of A is zero then set mXm to 256*H+A, else if bit 6 of A is zero then set xMx to 256*G+A, else set xMx to 256*L+A.
PB	6F	Copy B into Clip, copy PIR into B. Increment Low by 1. If bit 7 of B is zero then set mXm to 256*H+B, else if bit 6 of B is zero then set xMx to 256*G+B, else set xMx to 256*L+B.
ELF	70	Decrement L by one, store R into memory at address 256*L+0xC4. Increment Low by 1.
RM	71	Copy R + O to memory at address xMx. Increment Low by 1.
RH	72	Copy R + O to H. Increment Low by 1.
RG	73	Copy R + O to G. Increment Low by 1.
RL	74	Copy R + O to L. Increment Low by 1.
RS	75	Copy R + O to SOR. Increment Low by 1.
RP	76	Copy R + O to POR. Set POR to low impedance. Increment Low by 1.
RR	77	Do nothing. Increment Low by 1.
RD	78	Copy R + O to D. Increment Low by 1.
RJ	79	Copy R + O to Low.
RT	7A	If R plus O is not zero, then copy R into Low, else increment Low by 1.
RF	7B	If R plus O is zero, then copy R into Low, else increment Low by 1.
RC	7C	Copy High into H, copy R + O into High, copy Low into R, set Low to zero, set O to zero.
R1M	7D	Set O to -1. Increment Low by 1.
RA	7E	Copy A into Clip, copy R + O into A. Increment Low by 1. If bit 7 of A is zero then set mXm to 256*H+A, else if bit 6 of A is zero then set xMx to 256*G+A, else set xMx to 256*L+A.
RB	7F	Copy B into Clip, copy R + O into B. Increment Low by 1. If bit 7 of B is zero then set mXm to 256*H+B, else if bit 6 of B is zero then set xMx to 256*G+B, else set xMx to 256*L+B.
Group 2 — "TRAP" Instructions (Opcode bits 7,6 = 10)		
T00	80	Copy High into H, copy Low into R, set O to zero, copy 0x00 into H, set Low to zero.
T01	81	Copy High into H, copy Low into R, set O to zero, copy 0x01 into H, set Low to zero.
T02	82	Copy High into H, copy Low into R, set O to zero, copy 0x02 into H, set Low to zero.
T03	83	Copy High into H, copy Low into R, set O to zero, copy 0x03 into H, set Low to zero.
T04	84	Copy High into H, copy Low into R, set O to zero, copy 0x04 into H, set Low to zero.

Mnemonic	Opcode (hexadecimal)	Operation (Low = low order byte of program counter, High = high order byte of program counter, xMx is an implied 16-bit memory address. Pseudo register N stands for the next opcode in the instruction stream, essentially a code literal.)
T05	85	Copy High into H, copy Low into R, set O to zero, copy 0x05 into H, set Low to zero.
T06	86	Copy High into H, copy Low into R, set O to zero, copy 0x06 into H, set Low to zero.
T07	87	Copy High into H, copy Low into R, set O to zero, copy 0x07 into H, set Low to zero.
T08	88	Copy High into H, copy Low into R, set O to zero, copy 0x08 into H, set Low to zero.
T09	89	Copy High into H, copy Low into R, set O to zero, copy 0x09 into H, set Low to zero.
T0A	8A	Copy High into H, copy Low into R, set O to zero, copy 0x0A into H, set Low to zero.
T0B	8B	Copy High into H, copy Low into R, set O to zero, copy 0x0B into H, set Low to zero.
T0C	8C	Copy High into H, copy Low into R, set O to zero, copy 0x0C into H, set Low to zero.
T0D	8D	Copy High into H, copy Low into R, set O to zero, copy 0x0D into H, set Low to zero.
T0E	8E	Copy High into H, copy Low into R, set O to zero, copy 0x0E into H, set Low to zero.
T0F	8F	Copy High into H, copy Low into R, set O to zero, copy 0x0F into H, set Low to zero.
T10	90	Copy High into H, copy Low into R, set O to zero, copy 0x10 into H, set Low to zero.
T11	91	Copy High into H, copy Low into R, set O to zero, copy 0x11 into H, set Low to zero.
T12	92	Copy High into H, copy Low into R, set O to zero, copy 0x12 into H, set Low to zero.
T13	93	Copy High into H, copy Low into R, set O to zero, copy 0x13 into H, set Low to zero.
T14	94	Copy High into H, copy Low into R, set O to zero, copy 0x14 into H, set Low to zero.
T15	95	Copy High into H, copy Low into R, set O to zero, copy 0x15 into H, set Low to zero.
T16	96	Copy High into H, copy Low into R, set O to zero, copy 0x16 into H, set Low to zero.
T17	97	Copy High into H, copy Low into R, set O to zero, copy 0x17 into H, set Low to zero.
T18	98	Copy High into H, copy Low into R, set O to zero, copy 0x18 into H, set Low to zero.
T19	99	Copy High into H, copy Low into R, set O to zero, copy 0x19 into H, set Low to zero.
T1A	9A	Copy High into H, copy Low into R, set O to zero, copy 0x1A into H, set Low to zero.
T1B	9B	Copy High into H, copy Low into R, set O to zero, copy 0x1B into H, set Low to zero.
T1C	9C	Copy High into H, copy Low into R, set O to zero, copy 0x1C into H, set Low to zero.
T1D	9D	Copy High into H, copy Low into R, set O to zero, copy 0x1D into H, set Low to zero.
T1E	9E	Copy High into H, copy Low into R, set O to zero, copy 0x1E into H, set Low to zero.
T1F	9F	Copy High into H, copy Low into R, set O to zero, copy 0x1F into H, set Low to zero.
T20	A0	Copy High into H, copy Low into R, set O to zero, copy 0x20 into H, set Low to zero.
T21	A1	Copy High into H, copy Low into R, set O to zero, copy 0x21 into H, set Low to zero.
T22	A2	Copy High into H, copy Low into R, set O to zero, copy 0x22 into H, set Low to zero.
T23	A3	Copy High into H, copy Low into R, set O to zero, copy 0x23 into H, set Low to zero.
T24	A4	Copy High into H, copy Low into R, set O to zero, copy 0x24 into H, set Low to zero.
T25	A5	Copy High into H, copy Low into R, set O to zero, copy 0x25 into H, set Low to zero.
T26	A6	Copy High into H, copy Low into R, set O to zero, copy 0x26 into H, set Low to zero.
T27	A7	Copy High into H, copy Low into R, set O to zero, copy 0x27 into H, set Low to zero.
T28	A8	Copy High into H, copy Low into R, set O to zero, copy 0x28 into H, set Low to zero.
T29	A9	Copy High into H, copy Low into R, set O to zero, copy 0x29 into H, set Low to zero.

Mnemonic	Opcode (hexadecimal)	Operation (Low = low order byte of program counter, High = high order byte of program counter, xMx is an implied 16-bit memory address. Pseudo register N stands for the next opcode in the instruction stream, essentially a code literal.)
T2A	AA	Copy High into H, copy Low into R, set O to zero, copy 0x2A into H, set Low to zero.
T2B	AB	Copy High into H, copy Low into R, set O to zero, copy 0x2B into H, set Low to zero.
T2C	AC	Copy High into H, copy Low into R, set O to zero, copy 0x2C into H, set Low to zero.
T2D	AD	Copy High into H, copy Low into R, set O to zero, copy 0x2D into H, set Low to zero.
T2E	AE	Copy High into H, copy Low into R, set O to zero, copy 0x2E into H, set Low to zero.
T2F	AF	Copy High into H, copy Low into R, set O to zero, copy 0x2F into H, set Low to zero.
T30	B0	Copy High into H, copy Low into R, set O to zero, copy 0x30 into H, set Low to zero.
T31	B1	Copy High into H, copy Low into R, set O to zero, copy 0x31 into H, set Low to zero.
T32	B2	Copy High into H, copy Low into R, set O to zero, copy 0x32 into H, set Low to zero.
T33	B3	Copy High into H, copy Low into R, set O to zero, copy 0x33 into H, set Low to zero.
T34	B4	Copy High into H, copy Low into R, set O to zero, copy 0x34 into H, set Low to zero.
T35	B5	Copy High into H, copy Low into R, set O to zero, copy 0x35 into H, set Low to zero.
T36	B6	Copy High into H, copy Low into R, set O to zero, copy 0x36 into H, set Low to zero.
T37	B7	Copy High into H, copy Low into R, set O to zero, copy 0x37 into H, set Low to zero.
T38	B8	Copy High into H, copy Low into R, set O to zero, copy 0x38 into H, set Low to zero.
T39	B9	Copy High into H, copy Low into R, set O to zero, copy 0x39 into H, set Low to zero.
T3A	BA	Copy High into H, copy Low into R, set O to zero, copy 0x3A into H, set Low to zero.
T3B	BB	Copy High into H, copy Low into R, set O to zero, copy 0x3B into H, set Low to zero.
T3C	BC	Copy High into H, copy Low into R, set O to zero, copy 0x3C into H, set Low to zero.
T3D	BD	Copy High into H, copy Low into R, set O to zero, copy 0x3D into H, set Low to zero.
T3E	BE	Copy High into H, copy Low into R, set O to zero, copy 0x3E into H, set Low to zero.
T3F	BF	Copy High into H, copy Low into R, set O to zero, copy 0x3F into H, set Low to zero.
Group 3 — “ALU” Instructions (Opcode bits 7,6 = 1)		
Subgroup “GET-PUT” (opcode bits 5,4 = 00, 01, 10)		
G0A	C0	Copy A into Clip, copy value at memory address 0x80 into A. Increment Low by 1. If bit 7 of A is zero then set mXm to 256*H+A, else if bit 6 of A is zero then set xMx to 256*G+A, else set xMx to 256*L+A.
G1A	C1	Copy A into Clip, copy value at memory address 0x81 into A. Increment Low by 1. If bit 7 of A is zero then set mXm to 256*H+A, else if bit 6 of A is zero then set xMx to 256*G+A, else set xMx to 256*L+A.
G2A	C2	Copy A into Clip, copy value at memory address 0x82 into A. Increment Low by 1. If bit 7 of A is zero then set mXm to 256*H+A, else if bit 6 of A is zero then set xMx to 256*G+A, else set xMx to 256*L+A.
G3A	C3	Copy A into Clip, copy value at memory address 0x83 into A. Increment Low by 1. If bit 7 of A is zero then set mXm to 256*H+A, else if bit 6 of A is zero then set xMx to 256*G+A, else set xMx to 256*L+A.
L0A	C4	Copy A into Clip, copy value at memory address 0xC0 into A. Increment Low by 1. If bit 7 of A is zero then set mXm to 256*H+A, else if bit 6 of A is zero then set xMx to 256*G+A, else set xMx to 256*L+A.
L1A	C5	Copy A into Clip, copy value at memory address 0xC1 into A. Increment Low by 1. If bit 7 of A is zero then set mXm to 256*H+A, else if bit 6 of A is zero then set xMx to 256*G+A, else set xMx to 256*L+A.

Mnemonic	Opcode (hexadecimal)	Operation (Low = low order byte of program counter, High = high order byte of program counter, xMx is an implied 16-bit memory address. Pseudo register N stands for the next opcode in the instruction stream, essentially a code literal.)
L2A	C6	Copy A into Clip, copy value at memory address 0xC2 into A. Increment Low by 1. If bit 7 of A is zero then set mXm to 256*H+A, else if bit 6 of A is zero then set xMx to 256*G+A, else set xMx to 256*L+A.
L3A	C7	Copy A into Clip, copy value at memory address 0xC3 into A. Increment Low by 1. If bit 7 of A is zero then set mXm to 256*H+A, else if bit 6 of A is zero then set xMx to 256*G+A, else set xMx to 256*L+A.
AG0	C8	Copy A into memory at address 0x80. Increment Low by 1.
AG1	C9	Copy A into memory at address 0x81. Increment Low by 1.
AG2	CA	Copy A into memory at address 0x82. Increment Low by 1.
AG3	CB	Copy A into memory at address 0x83. Increment Low by 1.
AL0	CC	Copy A into memory at address 0xC0. Increment Low by 1.
AL1	CD	Copy A into memory at address 0xC1. Increment Low by 1.
AL2	CE	Copy A into memory at address 0xC2. Increment Low by 1.
AL3	CF	Copy A into memory at address 0xC3. Increment Low by 1.
G0B	D0	Copy B into Clip, copy value at memory address 0x80 into B. Increment Low by 1. If bit 7 of B is zero then set mXm to 256*H+B, else if bit 6 of B is zero then set xMx to 256*G+B, else set xMx to 256*L+B.
G1B	D1	Copy B into Clip, copy value at memory address 0x81 into B. Increment Low by 1. If bit 7 of B is zero then set mXm to 256*H+B, else if bit 6 of B is zero then set xMx to 256*G+B, else set xMx to 256*L+B.
G2B	D2	Copy B into Clip, copy value at memory address 0x82 into B. Increment Low by 1. If bit 7 of B is zero then set mXm to 256*H+B, else if bit 6 of B is zero then set xMx to 256*G+B, else set xMx to 256*L+B.
G3B	D3	Copy B into Clip, copy value at memory address 0x83 into B. Increment Low by 1. If bit 7 of B is zero then set mXm to 256*H+B, else if bit 6 of B is zero then set xMx to 256*G+B, else set xMx to 256*L+B.
L0B	D4	Copy B into Clip, copy value at memory address 0xC0 into B. Increment Low by 1. If bit 7 of B is zero then set mXm to 256*H+B, else if bit 6 of B is zero then set xMx to 256*G+B, else set xMx to 256*L+B.
L1B	D5	Copy B into Clip, copy value at memory address 0xC1 into B. Increment Low by 1. If bit 7 of B is zero then set mXm to 256*H+B, else if bit 6 of B is zero then set xMx to 256*G+B, else set xMx to 256*L+B.
L2B	D6	Copy B into Clip, copy value at memory address 0xC2 into B. Increment Low by 1. If bit 7 of B is zero then set mXm to 256*H+B, else if bit 6 of B is zero then set xMx to 256*G+B, else set xMx to 256*L+B.
L3B	D7	Copy B into Clip, copy value at memory address 0xC3 into B. Increment Low by 1. If bit 7 of B is zero then set mXm to 256*H+B, else if bit 6 of B is zero then set xMx to 256*G+B, else set xMx to 256*L+B.
BG0	D8	Copy B into memory at address 0x80. Increment Low by 1.
BG1	D9	Copy B into memory at address 0x81. Increment Low by 1.
BG2	DA	Copy B into memory at address 0x82. Increment Low by 1.
BG3	DB	Copy B into memory at address 0x83. Increment Low by 1.
BL0	DC	Copy B into memory at address 0xC0. Increment Low by 1.
BL1	DD	Copy B into memory at address 0xC1. Increment Low by 1.
BL2	DE	Copy B into memory at address 0xC2. Increment Low by 1.

Mnemonic	Opcode (hexadecimal)	Operation (Low = low order byte of program counter, High = high order byte of program counter, xMx is an implied 16-bit memory address. Pseudo register N stands for the next opcode in the instruction stream, essentially a code literal.)
BL3	DF	Copy B into memory at address 0xC3. Increment Low by 1.
G0R	E0	Copy value at memory address 0x80 into R, set O to 0. Increment Low by 1.
G1R	E1	Copy value at memory address 0x81 into R, set O to 0. Increment Low by 1.
G2R	E2	Copy value at memory address 0x82 into R, set O to 0. Increment Low by 1.
G3R	E3	Copy value at memory address 0x83 into R, set O to 0. Increment Low by 1.
L0R	E4	Copy value at memory address 0xC0 into R, set O to 0. Increment Low by 1.
L1R	E5	Copy value at memory address 0xC1 into R, set O to 0. Increment Low by 1.
L2R	E6	Copy value at memory address 0xC2 into R, set O to 0. Increment Low by 1.
L3R	E7	Copy value at memory address 0xC3 into R, set O to 0. Increment Low by 1.
RG0	E8	Copy R + O into memory at address 0x80. Increment Low by 1.
RG1	E9	Copy R + O into memory at address 0x81. Increment Low by 1.
RG2	EA	Copy R + O into memory at address 0x82. Increment Low by 1.
RG3	EB	Copy R + O into memory at address 0x83. Increment Low by 1.
RL0	EC	Copy R + O into memory at address 0xC0. Increment Low by 1.
RL1	ED	Copy R + O into memory at address 0xC1. Increment Low by 1.
RL2	EE	Copy R + O into memory at address 0xC2. Increment Low by 1.
RL3	EF	Copy R + O into memory at address 0xC3. Increment Low by 1.
Subgroup “MAP” (opcode bits 5,4 = 11)		
IDA	F0	Store A into R, set O to 0. Increment Low by 1.
IDB	F1	Store B into R, set O to 0. Increment Low by 1.
OCA	F2	Store one's complement of A into R, set O to 0. Increment Low by 1.
OCB	F3	Store one's complement of B into R, set O to 0. Increment Low by 1.
SLA	F4	Store result of shift-left A into R, set O to 0. Increment Low by 1.
SLB	F5	Store result of shift-left B into R, set O to 0. Increment Low by 1.
SRA	F6	Store result of logical shift-right of A into R, set O to 0. Increment Low by 1.
SRB	F7	Store result of logical shift-right of B into R, set O to 0. Increment Low by 1.
AND	F8	Store result of A logical-AND B into R, set O to 0. Increment Low by 1.
IOR	F9	Store result of A logical-OR B into R, set O to 0. Increment Low by 1.
EOR	FA	Store result of A logical-XOR B into R, set O to 0. Increment Low by 1.
ADD	FB	Store 8-bit result of A plus B into R, set O to 0. Increment Low by 1.
CAR	FC	Store carry bit (0 or 1) of A plus B into R, set O to 0. Increment Low by 1.
ALB	FD	Store 255 into R if A less than B, else store 0, set O to 0. Increment Low by 1.
AEB	FE	Store 255 into R if A equals B, else store 0, set O to 0. Increment Low by 1.
AGB	FF	Store 255 into R if A greater than B, else store 0, set O to 0. Increment Low by 1.