G-CPU Instruction Set

Data Movement Instructions:

Machine Codes (hex)	Instruction	Operand	Description	# of States
00	TAB	none	Transfer A to B (inherent addressing)	2
01	TBA	none	Transfer B to A (inherent addressing)	2
02 mm	LDAA #data	8-bit data	Load A with immediate data (immediate addr.)	3
03 mm	LDAB #data	8-bit data	Load B with immediate data (immediate addr.)	3
04 ll hh	LDAA addr	16-bit address	Load A with data from memory location addr (extended addressing)	5
05 ll hh	LDAB addr	16-bit address	Load B with data from memory location addr (extended addressing)	5
06 ll hh	STAA addr	16-bit address	Store data in A to memory location addr (extended addressing)	5
07 ll hh	STAB addr	16-bit address	Store data in B to memory location addr (extended addressing)	5
08 ii jj	LDX #data	16-bit data	Load X with immediate data (immediate addr.)	4
09 ii jj	LDY #data	16-bit data	Load Y with immediate data (immediate addr.)	4
0A ll hh	LDX addr	16-bit addr	Load X with data from memory location addr. (extended addressing)	6
0B ll hh	LDY addr	16-bit addr	Load Y with data from memory location addr. (extended addressing)	6
0C dd	LDAA dd,X	8-bit displacement	Load A with data from memory location pointed to by X + dd (indexed addressing)	4
0D dd	LDAA dd,Y	8-bit displacement	Load A with data from memory location pointed to by Y + dd (indexed addressing)	4
0E dd	LDAB dd,X	8-bit displacement	Load B with data from memory location pointed to by X + dd (indexed addressing)	4
0F dd	LDAB dd,Y	8-bit displacement	Load B with data from memory location pointed to by Y + dd (indexed addressing)	4
10 dd	STAA dd,X	8-bit displacement	Store data in A to memory location pointed to by X + dd (indexed addressing)	4
11 dd	STAA dd,Y	8-bit displacement	Store data in A to memory location pointed to by Y + dd (indexed addressing)	4
12 dd	STAB dd,X	8-bit displacement	Store data in B to memory location pointed to by X + dd (indexed addressing)	4
13 dd	STAB dd,Y	8-bit displacement	Store data in B to memory location pointed to by Y + dd (indexed addressing)	4

G-CPU Instruction Set

ALU Related Instructions:

Machine Codes (hex)	Instruction	Operand	Description	# of States
14	SUM_BA	none	Sum A, B and place in A (inherent addressing)	2
15	SUM_AB	none	Sum A, B and place in B (inherent addressing)	2
16	AND_BA	none	AND A, B and place in A (inherent addressing)	2
17	AND_AB	none	AND A, B and place in B (inherent addressing)	2
18	OR_BA	none	OR A, B and place in A (inherent addressing)	2
19	OR_AB	none	OR A, B and place in B (inherent addressing)	2
1A	COMA	none	Complement contents in A (inherent addressing)	2
1B	COMB	none	Complement contents in B (inherent addressing)	2
1C	SHFA_L	none	Shift A left by one-bit (inherent addressing)	2
1D	SHFA_R	none	Shift A right by one-bit (inherent addressing)	2
1E	SHFB_L	none	Shift B left by one-bit (inherent addressing)	2
1F	SHFB_R	none	Shift B right by one-bit (inherent addressing)	2
30	INX	none	Increment X (inherent addressing)	2
31	INY	none	Increment Y (inherent addressing)	2

Branch Instructions:

Machine Codes (hex)	Instruction	Operand	Description	# of States
20 bb	BEQ	addrL	Branch if A = 0, i.e., Z Flag = 1 (absolute addressing)	3
21 bb	BNE	addrL	Branch if A ≠ 0, i.e., Z Flag = 0 (absolute addressing)	3
22 bb	BN	addrL	Branch if A is negative, i.e., N Flag = 1 (absolute addressing)	3
23 bb	BP	addrL	Branch if A is positive (or zero), i.e., N Flag = 0 (absolute addressing)	3

Special Notes

1. Z flag and N flag are only set and cleared by the contents in register A.
2. A branch is accomplished by moving the operand address "addr" to the lower byte of the PC. The upper byte of the PC remains unchanged after a branch.
3. The Branch Instructions use absolute addressing where only the low byte of the address is used as an operand. If the branch condition is met, the high byte of the PC is unchanged and the low byte takes the value of the operand (addrL).
4. Explanations of the operands shown in the Machine Codes:
   - `mm` — 8-bit immediate data value
   - `ii` — Low-order byte of a 16-bit data
   - `jj` — High-order byte of a 16-bit data
   - `ll` — Low-order byte of a 16-bit address
   - `hh` — High-order byte of a 16-bit address
   - `dd` — 8-bit displacement value
   - `bb` — Low-order byte of a 16-bit address for a branch instruction