### *LOADING A PROGRAM FROM THE KEYBOARD*

1. Touch 'NORMAL RESET' key
2. Clear keyboard register
3. Key-in the address of the first instruction
4. Touch the 'LOAD ADDR' key
5. Key-in the first instruction
6. Touch the 'LOAD STORE' key
7. Touch the 'INCR ADDR' key
8. Key-in the next instruction
9. Touch the 'LOAD STORE' key
10. If any more instructions, repeat from 7
11. Load data stores in the same way (enter negative sign last)

### *RUNNING A PROGRAM*

1. Touch 'NORMAL RESET' key
2. Clear keyboard register
3. Key-in the address of the first instruction
4. Touch the 'LOAD IAR' key
5. Touch the stage keys 'A' - 'B' - 'C' or 'RUN'
6. Machine will stop if
   a) 'STOP' key is touched
   b) An input or display instruction is obeyed
   c) An overflow occurs or there is an invalid instruction

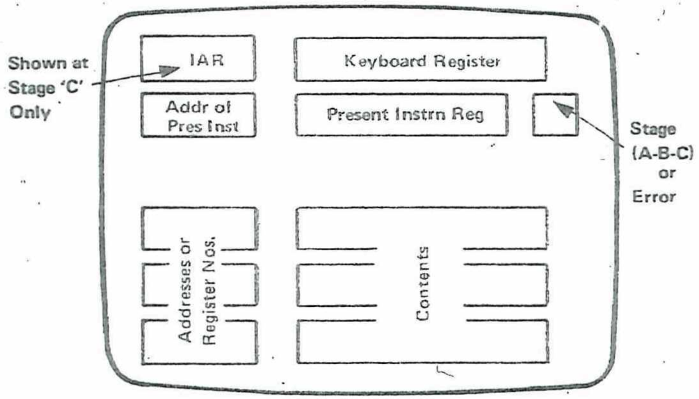
| *REASON FOR STOPPAGE* | *ACTION* |
|---|---|
| 'STOP' or stage keys touched | Touch 'A' - 'B' - 'C' or 'RUN' |
| Input instruction | Key-in required data and touch 'RUN' |
| Output instruction | To continue a program, touch 'RUN' |
| Overflow or invalid instruction | Correct program and start again |

### *OBEYING INSTRUCTIONS FROM THE KEYBOARD*

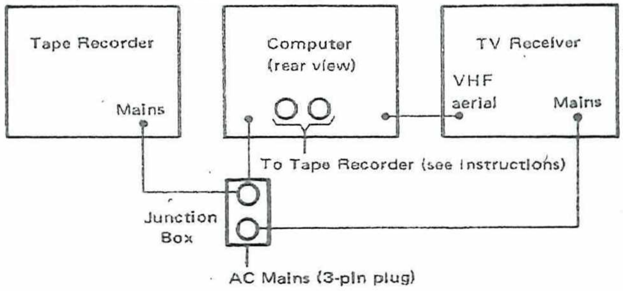
1. Touch 'KI RESET' key
2. Clear keyboard register
3. Key-in one 8-digit or two 4-digit instructions
4. Touch 'A', 'B' or 'C' as required

If two 4-digit instructions are entered, they will be obeyed in sequence.

### *LAYOUT OF TV DISPLAY*



### *ELECTRICAL CONNECTIONS*



IBM UNITED KINGDOM LTD

EXPERIMENTAL

SCHOOLS COMPUTER

## INSTRUCTION

## CARD

INSTRUCTION CODES →

← BRIEF OPERATING INSTRUCTIONS

## BASIC INSTRUCTIONS (4 digits)

General Form

| | | | |
|---|---|---|---|
| a | b | c | d |

a b is the 2-digit function code

### *REGISTER INSTRUCTIONS*

| *Register and digit* | | | *2 Registers* | | |
|---|---|---|---|---|---|
| c is a register number<br>d is a decimal digit<br>CODES | | | c and d are register numbers<br>CODES | | |
| a | b | | a | b | |
| 0 | 0 | $(R_c) \leftarrow (R_c) + d$ | 1 | 0 | $(R_c) \leftarrow (R_c) + (R_d)$ |
| 0 | 1 | $(R_c) \leftarrow (R_c) - d$ | 1 | 1 | $(R_c) \leftarrow (R_c) - (R_d)$ |
| 0 | 2 | $(R_c) \leftarrow d - (R_c)$ | 1 | 2 | $(R_c) \leftarrow (R_d) - (R_c)$ |
| 0 | 3 | $(R_c) \leftarrow d$ | 1 | 3 | $(R_c) \leftarrow (R_d)$ |
| 0 | 4 | INVALID CODE | 1 | 4 | $(R_c) \leftarrow$ R H 6 dig $(R_d)$ |
| 0 | 5 | Test $(R_c)$ for d | 1 | 5 | INVALID CODE |
| 0 | 6 | Left shift $(R_c)$ by d | 1 | 6 | Left shift $(R_c)$ by $\lvert(R_d)\rvert$ |
| 0 | 7 | Right shift $(R_c)$ by d | 1 | 7 | Right shift $(R_c)$ by $\lvert(R_d)\rvert$ |
| 0 | 8 | INVALID CODE | 1 | 8 | INVALID CODE |
| 0 | 9 | INVALID CODE | 1 | 9 | Display $(R_c)$ & $(R_d)$ |

*Conditions for code 05*

If the condition is fulfilled, then the CONTROL LATCH (CL) is set at 1, otherwise 0

| *Value of d* | *Condition* |
|---|---|
| 0 | $(R_c) = 0$ |
| 1 | $(R_c) > 0$ |
| 2 | $(R_c) < 0$ |
| 3 | L H dig $(R_c) = 0$ |
| 4 | R H dig $(R_c) = 0$ |

### *SINGLE-ADDRESS INSTRUCTIONS*

a determines the type of addressing
b determines the function
cd gives the 2-digit base address

CODES (For direct addressing)

| a | b | |
|---|---|---|
| 2 | 0 | $(R_0, R_1) \leftarrow (cd)$ |
| 2 | 1 | $(cd) \leftarrow (R_0, R_1)$ as data |
| 2 | 2 | $(cd) \leftarrow (R_0, R_1)$ as instruction |
| 2 | 3 | INVALID CODE |
| 2 | 4 | Branch & link to cd unconditionally |
| 2 | 5 | Branch & link to cd if (CL) = 1 |
| 2 | 6 | Branch & link to cd if (CL) = 0 |
| 2 | 7 | $(cd) \leftarrow (R_{LINK})$ |
| 2 | 8 | Input to cd |
| 2 | 9 | Display (cd) |

| *Digit a* | *Type of addressing* | | *True address* |
|---|---|---|---|
| 2 | Direct | | cd |
| 3 | Modified by | $(R_3)$ | cd + $(R_3)$ |
| 4 | | $(R_4)$ | cd + $(R_4)$ |
| 5 | | $(R_5)$ | cd + $(R_5)$ |
| 6 | Indirect | | (cd) |

## THREE-ADDRESS INSTRUCTIONS (8 digits)

General form :

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| a | b | c | d | e | f | g | h |

a b is the 2-digit function code

cd, ef, and gh are the three 2-digit addresses

CODES (For direct addressing)

| a | b | |
|---|---|---|
| 7 | 0 | (cd) ← (ef) + (gh) |
| 7 | 1 | (cd) ← (ef) - (gh) |
| 7 | 2 | (cd) ← (ef) x (gh) |
| 7 | 3 | (cd) ← (ef) ÷ (gh) |
| 7 | 4 | Branch to cd if (ef) = (gh) |
| 7 | 5 | Branch to cd if (ef) > (gh) |
| 7 | 6 | Branch to cd if $\lvert(ef)\rvert > \lvert(gh)\rvert$ |
| 7 | 7 | If (gh) ≠ 0 branch to cd & store link address in ef. |
| 7 | 8 | Input to cd; display (ef) & (gh) |
| 7 | 9 | Display (cd), (ef), (gh) |

The type of addressing is determined by the digit a :

| *Digit a* | *Type of addressing* | *True addresses* |
|---|---|---|
| 7 | Direct | cd, ef, gh |
| 8 | Modified by $(R_3)$, $(R_4)$, $(R_5)$ respectively | cd + $(R_3)$, ef + $(R_4)$, gh + $(R_5)$ |
| 9 | Indirect | (cd), (ef), (gh) |