Format CALL IO(type,address[,...])

CALL IO(type,bits,cru-base,variable,variable [,...])

CALL IO(type,length,vdp-address[,...])

Description

The IO subprogram allows access to and control of any chip in the console or peripheral cards. The type refers to different access methods like playing sound from GROM or VDP memory automatically. The type can also specify reading or writing directly to a Control Register Unit (CRU) address. Thereby allowing direct chip control, or direct chip bypass if the user wishes. The IO subprogram is a Graphics Programming Language (GPL) command. So the function is exactly like GPL despite being run from the XB environment. As most of XB is written in GPL the user gains greater GPL like control. After all the Operating System is written in GPL for a good reason.*Note these docs are from GPL Manuals.

type		address specifications
0	----------	GROM sound list address.
1	----------	VDP sound list address.
2	----------	CRU input.
3	----------	CRU output.
4	----------	VDP address of Cassette write list.
5	----------	VDP address of Cassette read list.
6	----------	VDP address of Cassette verify list.

The length specifies the number of bytes. The length can be from -32768 to 32767 depending on the amount of VDP memory that is available. Of course a value of -32768 is HEX >8000 and 32767 is HEX >7FFF and VDP normally in a TI is only 16384 or HEX >4000 of VDP. So be careful or lock-up will result. The cru-base is the CRU address divided by 2 in decimal form as the command automatically doubles the value input. The CRU -base ranges from 0 to 8191 or HEX >0000 to >1FFF with a EVEN address for 8 bits or more being scanned. That means that a value of 8191 will lock-up the system as it is looking for a bit in 8192 that does not exist.