

Transferring control to auxiliary memory

You can use the built-in routine named XFER to transfer control to and from program segments in auxiliary memory. You must set up three parameters before using XFER: the address of the routine you are transferring to, the direction of the transfer (main to auxiliary or auxiliary to main), and which page zero and stack you want to use.

Table 4-10
Parameters for XFER routine

Name or location	Parameter passed
Carry	1 = Transfer from main to auxiliary memory 0 = Transfer from auxiliary to main memory
Overflow	1 = Use page zero and stack in auxiliary memory 0 = Use page zero and stack in main memory
\$03ED	Program starting address, low-order byte
\$03EE	Program starting address, high-order byte

Note: The X, Y, and A parameters are preserved by XFER.

Put the transfer address into the two bytes at locations \$03ED and \$03EE, with the low-order byte first, as usual. The direction of the transfer is controlled by the carry bit: set the carry bit to transfer to a program in auxiliary memory; clear the carry bit to transfer to a program in main memory. Use the overflow bit to select which page zero and stack you want to use: clear the overflow bit to use the main memory; set the overflow bit to use the auxiliary memory.

After you have set up the parameters, pass control to the XFER routine by a jump instruction, rather than a subroutine call. XFER saves the accumulator and the transfer address on the current stack, then sets up the soft switches for the parameters you have selected and jumps to the new program.

Warning

It is the programmer's responsibility to save the current stack pointer at \$0100 in auxiliary memory and the alternate stack pointer at \$0101 in auxiliary memory before calling XFER and to restore them after regaining control. Failure to do so will cause program errors.
