as each 4-bit hex digit will result in an 8-bit ASCII equivalent. To convert a full byte, it must first be separated into 4 bit chunks (handled by the first routine) which are then converted by the second sub. RE.1 and RE.0 hold the finished products upon return. In this case no illegal byte checks are needed. It is not possible to represent hex digits greater than F or less than 0 with 4 bits. (This should be obvious, but it is awfully easy to forget that, when analyzed, we are really dealing with binary numbers all the time. Hex and Octal are products of the mind -- our minds -- and no binary computer has ever seen an "F" or an "A" and wouldn't know what to do with one if it did.)

POINT TO INSTRUCTION

This shorty takes advantage of the fact that all instructions begin at position #7 on any line. Counting from left to right, if the last 4 bits of RA equalling 0 means that RA points to the first character of a line, then if the last 4 bits of RA = 6, RA must point to the 7th byte of that 16-byte section. (If RA.0 = A0, then RA.0 = A6 = the seventh byte counting A0 as the first byte.) This routine strips off the last 4 bits of RA, replacing those bits with a 6. No matter where RA