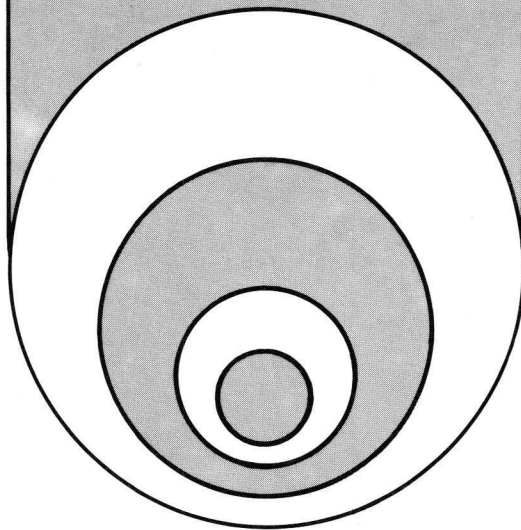


**signetics**

MOS  
MICROPROCESSOR



ABSOLUTE  
OBJECT  
FORMAT  
SS51

(REVISION NO. 1)

REVISION NO. 1

### INTRODUCTION

The format for absolute code produced for the 2650 is described in this application note.

The absolute object code is formatted into blocks. The first character of every block is a colon. Inside of a block, all the characters are hexadecimal, i.e., 0 to 9 or A to F, inclusive. Only non-printing ASCII control characters may occur within an interblock gap. These are the characters in the first two columns (columns 0 and 1) of the ASCII standard code table. A CR/LF is used within the interblock gap to reset the TTY or terminal after each block.

Each block is independent. For example, paper tape can be positioned prior to any block and a load started. The loading of absolute object code will be halted by:

- A BCC error on the address + count fields
- A BCC error on the data field
- An incorrect block length
- A non-hex character within the block

The block length field contains the number of bytes of actual data which is half the number of hex characters in the data field. While the size of the data field can range from 2 to 510 characters, a standard size of 60 characters has been established so that the tape may be easily generated and read on a variety of terminals and systems. A block length of zero indicates an End of File (EOF) block. The address field of an EOF block contains the start address of the loaded program.

The Block Control Character is 8 bits formed from the actual bytes and not from the ASCII characters. The bytes

are in turn exclusive or'ed to the BCC byte, and then the BCC byte is left rotated one bit. It appears as two hex characters. Both the address and count fields and the data field are followed by a BCC character pair. The BCC prevents storing data at an invalid memory address or storing bad data into memory.

EXAMPLE: An object tape that loads ten bytes starting at location 500  
:05000A3C0455B024FFF01F05040030  
:000000

### FORMAT

1. Interblock gap of any non-printing characters including spaces
2. Start of block character; a colon
3. Address field; four hex characters
4. Count field; two hex characters in range 0 to 1E
5. BCC for address and count fields; two hex characters
6. Data field; twice the value in the count field which is the number of memory locations loaded by the current block
7. BCC for the data field; two hex characters

### EXAMPLE OF OBJECT FORMAT

:05000A3C0455B024FFF01F05040030

② ③ ④ ⑤                      ⑥                      ⑦

- 2 – Start of block character (colon)
- 3 – Starting address for block (H'0500')
- 4 – Number of bytes in block (H'0A' = 10)
- 5 – BCC byte for fields 3 and 4 (H'3C')
- 6 – Data, two characters per byte
- 7 – BCC byte for field 6 (H'30')