516-16
ADH
12/17/68

516 ASSEMBLER AND POST PROCESSOR FOR UNSEGMENTED (e.g., SYSTEM) PROGRAMS

A GMAP assembler is available for DDP-516 unsegmented programs (such as hard-core system subroutines). This assembler produces a relocatable binary deck, in GMAP format. Several such decks may be loaded and linked by a companion GE post processor, which produces an absolute binary DDP-516 deck, ready to load and execute.

1. GMAP Assembler

The deck set-up for running a 516 assembly is as follows:

```
1          8          16
$          IDENT      [usual information]
$          GMAP       NGMAC
[optional  TTL, TTLS, LBL cards]
$          SELECT     GEDISK/DDP-516/SYSTEM/GMAP
[symbolic  deck]
$          ENDJØB
```

Several programs may be assembled in one run; the $ GMAP and $ SELEC cards must be supplied with each program.

The symbolic deck is punched in standard GMAP format. It must concl with an END card. The standard DAP-16 mnemonics may be used for the memory-reference, shift/rotate, and generic instructions. The input, output instructions are not coded by giving (say) an INA with a 10-bit address field; instead, all input/output instructions are given distinct names, just like OTK (which is really one of the SMK instructions). See 516-2 for the mnemonics assigned to Dept. 1353 IO hardware. Indirect addressing and indexing are specified in GMAP format:

```
1          8          16
           LDA        X          [no modification]
           LDA        X,1        [tag]
           LDA        X,*        [flag]
           LDA        X,1*       [tag and flag]
```

A program may have one ENTRY point; if a program has no SYMDEF's, it must have an ENTRY point. An ENTRY point becomes the address on the transfer card of the absolute binary deck produced by the post processor. It is defined by the macro

```
1          8          16
           ENTRY      name
```

where name is a relocatable symbol, or a relocatable expression composed of predefined terms (i.e., name must satisfy the rules for the variable field of FEQU).

An ASCII macro has been defined which converts the GE 64-character set into the middle half of the 128-character ASCII set. The symbolic format is as follows:

```
1         8          16
          ASCII      (ab,cd,ef,...)
```

where a goes into the left half of the first word, d goes into the right half of the second word. (The leading bit of these 8-bit characters is 0.) ASCII literals may be defined as follows:

```
1         8          16
          LDA        =Hab
```

A TVDEF macro has been defined to simplify the coding of transfer vectors. The general coding format is as follows:

```
1         8          16
          TVDEF       (name list)[,op,arg]
```

The first argument is either a single name or list of names. All names on the list are SYMDEF'd by the macro, to make them accessible externally. If the second and third subfields are missing, then each name is the name of a one-word storage location; the macro expands into:

```
1         8          16
name1     BSS         1
name2     BSS         1
...
```

Otherwise, the names are synonyms for the location containing the data word specified by the second and third subfields. For example,

```
1         8          16
          TVDEF      (A,B),ØCT,17
```

expands into

```
1         8          16
A         EQU         *
B         EQU         *
          ØCT         17
```

One further special case arises if the second (op) subfield is ADDR, for address. In this case, the third subfield is SYMREF'd, because it is assumed to be an external symbol. If an internal addres is desired, the DAC mnemonic may be used instead of ADDR in the second subfield.

An ERRØR macro has been defined:

```
1       8         16
        ERRØR     number
```

expands into

```
1       8         16
        JST       ERRØR
        DEC       number
```

The symbol .RZERØ has been defined as a relocatable 0.

2. GE Post Processor

The deck set-up for post processing is as follows:

```
1       8         16
$       IDENT     [usual information]
$       SELECT    GEDISK/DDP-516/SYSTEM/POSTPROC
[binary decks and control cards]
$       ENDJØB
```

It is possible to post process several times in one run; each time, the $ SELECT card precedes the binary and control deck.

The binary decks are those produced by the GMAP assembler, with all non-binary cards removed (hence, remove the $ OBJECT and $ DKEND cards).

The control cards are punched in free-field format, that is, the first subfield may begin anywhere, and following subfields are separate by commas or blanks (or both). Only the first 72 columns of each card are used. The first subfield is the control word. The 2 control card types are described next.

LABEL name

where name, with 1-6 characters, defines the label to be punched in columns 72-78 of the binary output deck; columns 79-80 provide the usual sequential numbering. This card may be placed anywhere in the deck. If it is omitted, the default label is DDP516.

ORIGIN location

where location is an octal number, sets the loading origin for the following binary decks. The default origin is 0 for the first deck, and the first location after the preceding deck for all others.

Note: A blank control word is treated as a comment indicator; the card is listed but ignored.

The loader provides automatic inter-sector linkage when required. Linkages will be inserted into sector 0, starting at the top (777_8) and working down to location 100_8. Before a new linkage word is generated, a search is made of all data in locations 100_8-777_8, and if an existing data word is identical to the required linkage word, the existing word is used instead of generating another word.

The post processor provides a printout of the control cards, all external references and all data words.

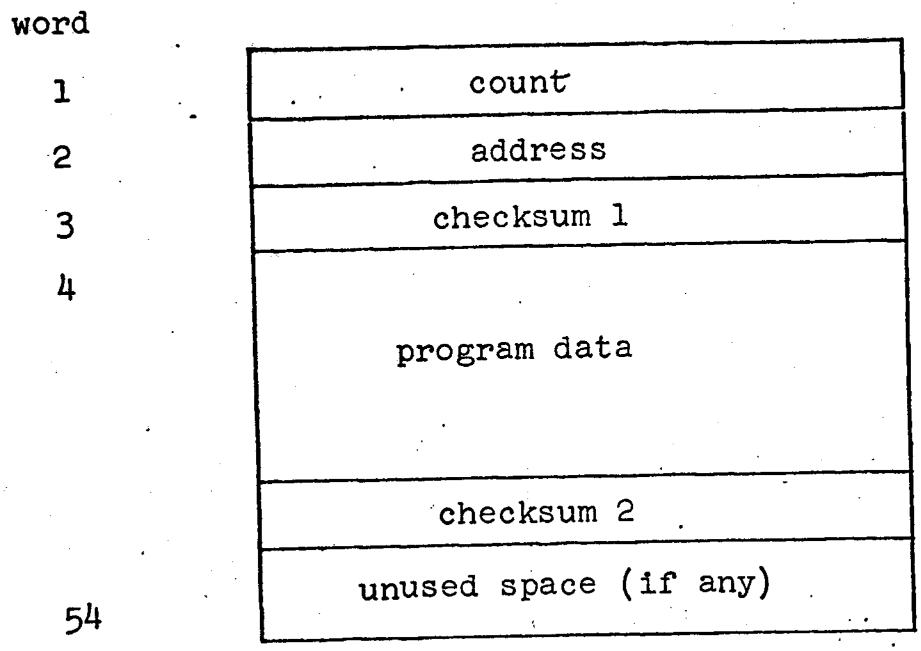
3. Absolute Binary Card Format

Data is punched in the first 72 columns of the binary output cards. Identification and sequence information is punched in columns 73-80, as described above. The data is packed onto the card, 3 words to 4 columns, as shown:

Row Column	12		9
4k-3	upper 4 bits word 3k	upper 4 bits word 3k-1	upper 4 bits word 3k-2
4k-2	lower 12 bits, word 3k-2		
4k-1	lower 12 bits, word 3k-1		
4k	lower 12 bits, word 3k		

where k ranges over 1-18, column number ranges over 1-72, and word number ranges over 1-54.

A word-oriented description of card format is clearly easier to provide than a column-oriented description:



The count (first) word is the 1's complement of the number of words of program data on this card. If the count is -1 (0 data words), then this is a transfer card rather than a data card. The address (second) word for a data card is the BES address of the data block, i.e., the address +1 of the last program data word on this card; also, the tag bit is set in the address word. In the case of a binary transfer card, the address is the transfer address; the tag bit is off in this case. The checksums are the negatives (2's complement) of the sums (2's complement) of their respective domains: checksum 1 checks words 1, 2, and checksum 2 checks the program data (words 4-54 maximum).

One important consequence of the chosen card format is that rows 6-9 of the first column will always be punched; the top 8 rows of the first column may or may not be punched.