38

A)

1988 International Conference On Computer Processing of
Chinese And Oriental Language
Aug. 29, - Sept. 1, 1988
Tutorial Program, Moday, Aug. 29, 1988
Toronto, Canada

CHINESE CHARACTER DATA BASE (CCDB) AND CODING CHINESE, JAPANESE AND KOREAN MACHINE TRANSLATION

Jack Kai-tung Huang, C.T. Chang, C.C. Hsieh, C.C. Yang,
and S.S. Tseng
Chinese Character Analysis Group
Council for Cutural Planning and Development
Executive Yuan
Yang Ming Shan P.O.B. 1-123, Taipei, R.O.C.
Taipei, Republic of China

Abstract

This papaer covers the following subjecs of the Chinese Character Code for Information Interchange (CCCII) : Historical remarks of CCCII; The arrangement of the CCCII chinesec sets; The code structure of CCCII; Introduction of Chinese Character Data Base (CCDB) and a proposal of CCCII Machines; The Current status of CCCII; And the environment of multilinguist coding system based on CCCII and CCDB as a proposed hypermedia for language machine translation system.

A data base of Chinese character's attributes with multibytes coding system conversion mechanism will be presented. The structure of this data base system is based on CCCII and the first version of it was completed in 1983. Within this data base there were more than twenty attributes of the Chinese characters collected, in which it has cross reference mapping tables of CCCII with all the two-byte code such as GB 2312-80, JIS C 6226, KSC-5919, and the CNS and Big-5, the local codes in Taiwan. For convenience, CCCII provide coding spaces to JIS and Ksc with an added redundency third byte to treat them as a three bytes CCCII, and the CCDB provide a mechanism to convert them one to anothers. Also, the single byte coding system of ASCII and other ISO standard linguists can be put in CCCII with the same method as the two bytes code stated above.

The US Library of Congress anounce their standard for East Asian Character Code (EACC) used on the network of library information, those are thoroughly based on CCCII. EACC will become an ANSI standard currently.

CHINESE CHARACTER DATA BASE (CCDB) AND CODING CHINESE, JAPANESE AND KOREAN MACHINE TRANSLATION

1. Historical remarks of Chinese Character Code for Information Interchange (CCCII).

The Chinese Character Analysis Group (CCAG), a non-profit research organization, was formed on December 25, 1979. The main project of this group intends to establish the Chinese Character Code for Information Interchange. The purpose of this work is to provide an excellent and perfect environment for Chinese Information Processing System and the so called Chinese Computers. It may concern to those who do the academic research and the professional work in the field of Chinese Information processing internationally. This research work related to the subtopics as follows : collecting and categorizing the Chinese characters; Identification of the Chinese punctuations; Identification the structural Chinese character shapes; Pronunciation of the Chinese characters; Organization and deduction of the Chinese character structures; Collating sequence and Indexing methods of the Chinese characters; Attributes of the Character Data Base; and to research and develope the methods to teach Chinese with the aid of computers.

Because Chinese is an ideographical language and the Chinese characters are an open set, it is really difficult to collect exactly all the existed Chinese characters which are still in use today. However, in the COMPREHENSIVE CHINESE DICTIONARY by Dr. Chi-yun Chang, there are 49,906 characters, both ORTHOGRAPHIC and VARIANT forms are included. Some of the Chinese characters have one or more VARIANT forms, which have exactly the same pronunciation and the same meaning, but are different in their stroke structures. Between the VARIANT forms of a character, we usually choose one of them as on ORTHOGRAPHY, and they are interchangeable in writing, except when they are used as identifiers to name a person, places or things. The relationship between the ORTHOGRAPHIC and its VARIANT forms of a character in many cases has important implications. The SIMPLIFIED forms used in the mainland China are one of the VARIANT forms, all the VARIANT forms are collected, coded, and allocated from the second layer to twelfth layer in the coding scheme. Up to now, there were about 74,000 chinese characters collected and 53,000 of them were coded and published already.

In April of 1980, the CCAG published the First Edition of the Chinese character Code for Information Interchange which contained the complete coding structure of CCCII. It remains the same today. It coded 4,808 Most Frequently Used Character Set, and 41 phonetic (37) and tonal symbols (4) [1].

2. The arrangement of the CCCII Chinese characters

All orthographic forms of Chinese characters, which are estimated to be around 50,000 and hence could be placed within the first layer. They are classified into four sets based on their usage frequency (provided by the Ministry of Education, The Republic of China) as follows:

(1) Most Frequently Used Character Set: contains 4,808 characters.
(2) Next Frequently Used Character Set: contains 17,032 characters.
(3) Rarely Used Character Set: contains 20,583 characters.
(4) Suplemental Character Set: unknown number.

The order of Chinese characters for each character set mentioned above is arranged according to the established radical and stroke-count sequence. The three sorting keys are defined as follows:

(1) First sorting key : The natural sequence of radicals of the Kang Hsi Dictionary radical system. It contains 214 radicals ordered in ascending stroke-count sequence.

(2) Second sorting key : The ascending order of the stroke-count excluding the radical of the character.

(3) Third sorting key : The precedence of the strokes excluding the radical of the character. The arrangement of the precedence of strokes is defined as:

(a) A dot (dean), weight is 1.
(b) A horizontal stroke (herng), weight is 2.
(c) A vertical stroke (jyr), weight is 3.
(d) A stroke down from the upper right to lower left (piee). weight is 4.

(e) A stroke down from the upper left to lower right (nay).
weight is 5.

3. Code structure of the CCCII.

The CCCII and its communication system is based on the ISO 646 communiction 7-bit coding standard. It utilizes three 7-bit byte to represent a given Chinese character, figure 1. Its technique of code extension and the identifying location of the escape sequence are based on the ISO 2022.

This protocal provides a three dimensional space, i.e., 94*94*94 coding positions, for the Chinese characters used in computing and its development of the escape sequence. This method could encompass the coding for more than 50,000 Chinese characters. Meanwhile, it is also suggested that the same multiple byte coding scheme of the graphic symbols should be used for the symbols of the escape sequence.

It is planned that all the current Chinese characters, including all variant and simplified characters, will be coded with a three byte code. The frequency of usage and the classification of characters are used as the guiding principles.

The code, consisting of three 7-bit byte vectors, forms a finite three dimensional coding space, i.e., it provides a total 94*94*94 positions according to ISO 2022. It is called a space of 94 PLANES, with 94 sections in each plane, and 94 positions in each section, figure 2.

Since certain characters have variant and simplified forms, the coding spaces have been subdivided into layers so that the character codes of the variant and/or simplified form of a given character could be used to trace and to link back to the original characters and vise versa. Such relationships between the orthographic characters and their variant forms could in many cases, have important implications. The 94 planes are grouped into 16 layers. Each layer, from 1 through 15, is made of 6 consecutive planes, while the last layer (16th) has only 4 planes, figure 3.

Each Layer, thus, can have 53,016 (=94*94*6) coding spaces. However, the first Section of each Plane is reserved for the Control Codes that are required in handling Chinese full text process. This reduces the total coding space for Chinese characters in each Layer to 52,452 (=53,016-94*6) Positions. Section 2 to 15 of the first Plane of each Layer, except Layer 16, are reserved for User's area, there are 1,316 (=94*14) Positions in each Layer, and it has a total of 19,740 (=1,316*15) Positions in the whole coding space for user's usages. These areas are used for the user's collection of a special set of characters in the private field and/or privately designed new symbols of characters. Therefore, each Layer actually only has 51,136 (=52,452-1,316) usable coding spaces, figure 4.

The usage of these 16 Layers is described as follows:

(1) Layer 1,i.e., (Plane 1 through 6) is used to designate the Graphic and Chinese Orthographic Characters of the:

- (a) Arithmetic and ASCII symbols (Plane 1, Section 2-3).
- (b) 35 Chinese punctuation marks (Plane 1, Section 11).
- (c) 214 Radicals (Plane 1 Section 12-14).
- (e) 41 Chinese numerical characters, 37 Chinese Phonetic Symbols and 4 Tone Marks (Plane 1, Section 15).
  [special note: up to this point, it's in the User's area of the first Layer].
- (f) The 4,808 Most Frequently Used Chinese Characters (Plane 1, Section 16-67, Hexadecimal values starts from 213031 to 216330.
- (g) The 17,032 Next Frequently Used Chinese Characters (Plane 1, Section 65 to Plane 3, Section 64, Hexadecimal values starts from 216421 to 236072).
- (h) 20,583 Rarely Used Characters (Plane 3, Section 65 to Plane 6, Section 5, Hexadecimal values starts from 236121 to 262543).

(2) Layer 2, i.e., Planes 7 through 12, is used for designating the Simplified Forms of Chinese characters which are used in the Chinese

(3) Layer 3 through 14 are used to designate the Variant Forms of the Chinese characters that appeared in Layer 1. The handling of these Variant Forms will be discussed in the following sections. Currently, only up to the 12th Layers are used for Variants. The JIS C 6226 characters are treated as Variant Forms and have been placed in Layer 13. Layer 14 is reserved for the Chinese characters used in Korean.

(4) Layer 15th is reserved for other usage.

(5) The 16th Layer is the coding area for other languages which are most closely correlated to the Chinese language. Also, the Shifted JIS C 6226 Kanji characters are placed in Plane 91st of this Layer. Plane 92nd is reserved for the Korean KIPS. The 93rd Plane is reserved for the supplementary Chinese characters. The last Plane (94) will be used to contain the non-Chinese characters of CB2 (Character Block 2).

4. Current Status of CCCII

(1) The Most Frequently Used Character Set:

April 15,1980, the First Edition of the Chinese Character Code For Information Interchange was published and the complete coding structure of CCCII was announced by CCAG.

(2) Next Frequently used Character Set:

In October of 1982, Second Edition of the CCAG's work were published [2]. It contains 33,544 characters, in which 21,885 were Orthographic characters and 11,660 Variant characters, which in turn contained 3,752 Simplified characters. A separate book on the Variant characters [3], and a printed form of the Chinese Character Data Base were also published at the same time [4].

(3) Rarely Used Character Set:

In February of 1987, The CCAG released the latest updated revision in a similar format to the earlier books [5]. However, in this set, it contains 20,583 Rarely Used Characters. This brings the total published number of characters to 53,940.

Above, in total, there were 42,423 Orthographic Forms of Chinese Characters coded in the CCCII.

In Plane 6, there still 8,366 coding spaces (i.e., 89 Sections) remainds unsed, these are reserved for extention of the Orthographic characters.

Up to this point , 42,423 Orthographic Forms and 11,517 Variant Forms, 53,940 Chinese characters coded in total were coded in the CCCII by the CCAG.

The Chinese Character Analysis Group has planned to release the most comprehensive set of publications, which will contain information of more than 74,000 characters CCAG collected, and analyzed throughtout these many years. No one can be absolutely sure that this will contain "all" Chinese characters, but it will be the most complete job ever accomplished in this field.

Based on preliminary survey and statistics, the collection made by Census Automation Project of Taiwan, R.O.C.. There are at least more than 12,000 to 18,000 Chinese characters that were estimated by the Project, are outside the collection of 48,174 characters collected by the Ministry of Education previously. Now these characters are under analysis by the CCAG, and will be coded and included in the future Variant Characters and the Rarely Used Character Set of CCCII.

Recently, we have learned that the CCCII/EACC (East Asian Character Code) is in the process of becoming an ANSI standard.

5. Chinese Character Data Base (CCDB).

Chinese Character Data Base (CCDB), shown as figure 6, is a data base together with necessary software that was developed based on the structure of

CCCII in order to make the application of CCCII easy and effecient. CCDB has three main functions:

(1) To provide Chinese character constructions, phonetic transcription, dot matrix configurations of Chinese characters, various correlated input methods and various existed corresponding Chinese character coding systems [6], please see figure 7.

The Chinese character Data Base (CCDB) provides more than 20 attribute indexed files of the CCCII with cross-references, any character can be searched by following methods, such as:

-Pronunciation
-Radicals
-Stroke-count
-Four Corner Code
-Three Corner Code
-Telegraph Code
-Big-5 Code
-Dragon Code
-General Chinese Character Standard Interchange Code (the old Version of Oct., 1983 only, but not the Version of Mar., 1986 and the correlated Version of Association Code of Jan., 1988)

(2) To provide controlling and searching machanism of transcription table between correspponding lexicographic character fonts and phonetic symbols.

(3) To provide the interchanging function among various corresponding multi-bytes coding systems, for example, from 3-byte CCCII to 2-byte JIS C 6226 or from 2-byte JIS C 6226 to 3-byte CCCII, etc.

The CCDB file structure is mainly organized by two portions: data files and indexing files, the data files subdivided into 16 LAYERS. All files are grouped into seven file-groups as described below:

(1) G1 (files of Layer 1) : designates the data files of Orthographic FORM of Chinese characters.

(2) G2 (files of Layer 2) : designates the data and searching files of Simplified Forms of Variant Forms of Chinese characters.

(3) G3 (files of Layer 3-12) : designates the data and searching files of other Variant Forms of Chinese characters appearing in Layer 1. Currently, only five selected Variant Forms are vertically allocated from Layer 3 to 7.

(4) G4 (files of Layer 13) : the data and searching files of Japanese Kana and 6,353 Kanji of JIS C 6226 are allocated in Layer 13.

(5) G5 : Inverted lists of Chinese lexiographic fonts, phonetic symbols, and the other atributes of Chinese characters.

(6) G6 : Indexed tables of various Chinese character input methods, e.g., 3-corner code, Dragon input method, etc.

(7) G7 : Cross-reference tables of various existed relevant coding systems.

The publications of CCDB released by CCAG on:

-Second Edition, October, 1985.
-Revised Edition, May 1985.
-Edition of CCDB for Rarely Used Characters, May, 1987.

6. The searching mechanism of CCDB

The Chinese characters stored in CCDB are hired R94, a 16-bit hexadecimal condensed code of CCCII. The file groups separated by a Layer Pointer(LP). To calculate R94 and LP, CCDB provide formulas as:

R94=[(B3-33) mod 6]*94*94+(B2-34)*94+(B1-33), and
LP=(B3-33)/6.

The searching mechanism of CCDB uses three different addressing modes: please see figure 8.

(1) Direct addressing : If LP=0, the R94 code is the storing address of the Orthographic Form of Chinese characters in Layer 1 in the environment of CCCII.

(2) Indirect addressing : If LP=1, the R94 code is the address of the storing address of the Simplified Form of Chinese characters in LAYER 2. If LP=12, R94 code is the indirect address of Japanese Kana and Kanji of JIS C 6226 allocated in Layer 13 in the environment of CCCII. If LP=13, the R94 code is the indirect address of KIPS or KSC-5916 Korean characters which to be allocated in the environment of CCCII.

(3) Indirect-displacement addressing : If LP=2 to 11, the R94 code is the address of a Base, denoted by B, and the storing address of the Variant Forms of chinese characters in Layer 3 through Layer 12 in the environment of CCCII are calculated by the formula m=B+LP-2, where LP must less than or equal to the upper boundary of LP(L). The upper boundary of LP and B are different from character to character, so it should be pre-determinated when the character was stored.

The above mentioned three addressing modes are also applicable to the currently reserved Planes in Layer 15 to 16 of CCCII. The CCDB provides this mechanism for multi-lingual data processing as a universal coding system as long as those languages were coded within the scope of CCCII. One important and common feature of CCCII and CCDB we would like to point out is that both mentioned systems are open systems in nature, subjecty to increment of appending of relevant searching and data files without further modification of the systems.

7. Code-format Switchers of CCDB

The CCDB is based on a 3-byte CCCII. TO enhance the efficiency of data communication and storage, the 3-byte code can be transformed to 2-byte or 1-byte codes in accordance with the usage of control codes defined by CCCII [7]. Those sets of control codes are called code-format switchers. CCDB provides the control processing algorithm of code-format switchers as shown in Appendix A.

8. Introduction to the CCCII-machine

For the goal of implementation of a universal coding system, the CCCII-machine was designed that based on the structure of CCCII. The CCCII-machine system constructed by three main components, namely; (1) Kernel, (2) Transmission Unit and (3) Controller. Please see figure 9.

(1) Kernel : CCDB is the Kernel in fact. By means of the proper routine calls, the information of CCDB will be accessed. Some flags tell the states of activities.

(2) Transmission unit : The transmission unit consists of the CORE (COde REceiver) and COTRA (COde TRAsmitter). It serves as an interface between CCDB and data communication devices. The CORE and COTRA are constructed by specific data structure and routines. Under the proper routine calls, the CORE and COTRA will be activated. Please refer to figure 10 and 11.

(3) Controller : Controller is the principal control program of CCCII-machine. Because CCCII-machine may be implemented on a terminal, PC, or a host computer, then the controller must provide different functions in the different places of different applications. Thus, the controller is application dependence.

9. CORE (COde REceiver) of CCCII-machine:

Refer to figure 10 structure of CORE, the CORE receives a character stream sending from the data transmission device. The character stream usually is an mixture of the control codes of ISO-646, the control codes of CCCII, the 3-byte and/or 2-byte and/or single byte codes of CCCII. The responsibilities of CORE are to pick up those and give them the proper treatment and process, especially, the single byte or 2-byte codes should be transferred back to 3-byte representations. The data structure of CORE is explained as follows:

RI :	7-bit input register.
RB3 :	7-bit register stores B3 of CCCII.
RB2 :	7-bit register stores B2 of CCCII.
RB1 :	7-bit register stores B1 of CCCII.
SD3 :	A stack to keep the B3 default values for PSL/PSU/PSR control codes under 2-byte condition.
SP :	Stack point for SD3.
RD3 and RD2:	7-bit registers store B3 and B2 default valuses for SSU code under 1-byte condition.
S :	state register, value 0 indicates the 3-byte condition, value 1 indicates the 2-byte condition and value 2 indicates the 1-byte condition.
C :	Modula 3 register for byte position control, if 0 indicates next byte into RI is B3. 1 indicates next byte into RI is B2. 2 indicates next byte into RI is B1.
F :	Register for error status.

The routine componets of CORE composed of a main control routine and a group of correlated subroutines. The flowchart of the main control routine is shown in figure 12. The main control routine is an infinite-loop by nature. Once the controller starts, the main control routine of CORE will endlessly operate.

The main control routine is also served as an interrupt-driven routine. It is liable to become idle after received a new byte and processed. It will wake-up when next byte enters the RI. To stop the main control routine of CORE, it must be operated by means of system break. The details of design of idle, wake-up and break are implementation-and-user dependent.

Error-handling functions are included in each subroutine. when a error occurs, the status of CORE will be handled by F register of CORE, the controller will take over after. A set of routines which wrote in PASCAL is shown in Appendix B.

10. <u>COTRA (COde TRAnsmitter) of CCCII-machine:</u>

Same as CORE, the COTRA is constructed by two components; the data structure and the routines as shown in figure 11 Structure of COTRA. In the data structure, except the S is defined slight differently from CORE, the definitions of RB3, RB2 and RB1 are the same as in CORE.

S is designed to control the number of bytes of CCCII to transmit out. When the controller calls COTRA it requests to transmit a CCCII. Functions of S are as follows:

(1) If S=0, then transmit all the RB3, RB2, and RB1.
(2) If S=1, then transmit the RB2 and RB1.
(3) If S=2, then transmit the RB1.

It is necessary to utilize routine calls to modify the contents of S. If the content of S is changed, COTRA will transmit corresponding control codes and parameter(if needed). as shown figure 13.

There is no main control routine for COTRA routines, because COTRA is totally passive and is completely controlled by controller. A set of programs for COTRA routines which were written in PASCAL is shown in Appendix c.

11. A guide to use the CCCII and CCDB

As mentioned above, the CCCII was only a universal collection of Chinese characters with their ID codes. The CCDB consists of data file groups of the attributes of the CCCII's characters and indexing files of those data files. How to use them depends on the user's working field. In order to save the storage space and improve the retrieval efficiency, the special user may select a special subset of characters in accordance with their requirements from the universal set of CCCII. EACC, for example, is a subset of CCCII selected by the Research Library Group of the States used to catalog the Chinese, Japaness, and Korean books for Library Bibleography.

Characters is the media to record the information and the language. The users working in the field of Chinese information processing, the natural language processing, the full text Chinese document processing, the Chinese office automation, the research of Chinese CAI, the study of the knowledge base of Chinese characters, and so forth all need to use the CCDB and CCCII. It is obviously that the users all know how to use them.

12. Conclusion

It is hard to explain that what is "language machine translation", because it need not only the whole set of characters, but also need the words or vocabularies, the rules of word structures, the meaning and it's category structure of the characters and vocabularies, and the semantics and grammar of a sentence. More than that, it needs the algorithms to analysis the language's architectures, the technique of Artificial Intelligence, and other knowledge and skill. However, the CCDB and CCCII were the basic and fundamental media for this sort of works. Talking about language machine translation between the Chinese and other languages, it can not forget the CCDB and CCCII at all.

It was learned that the EACC is in process been an ANSI standard. This means that the result of CCAG will be appreciated internationally.

13. References:

1. Chinese Character Analysis Group, Chinese Character Code for Information Interchange, Volume 1, April, 1980.

2. Chinese Character Analysis Group, Chinese Character Code for Information Interchange, Second Edition, October, 1982.

3. Chinese Character Analysis Group, Variant Forms of Chinese Character Code for Information Interchange, Volume II, Second Edition, December, 1982.

4. Chinese Character Analysis Group, Chinese character Data Base, Second Edition, October, 1982.

5. Chinese Character Analysis Group, Chinese Character Code for Information Interchange, Volume III, February, 1987.

6. The Chinese Character Analysis Group, Chinese Character Data Base, Revised Edition, may, 1985.

7. The Chinese Character Analysis Group, Explanation for Control Codes of CCCII, March 13 1986.

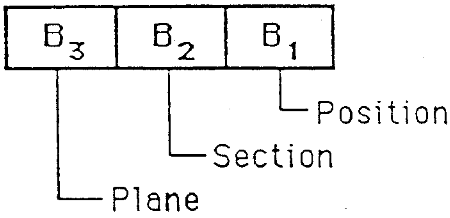


Figure 1 Three 7-bit Byte extension--- ISO 2022

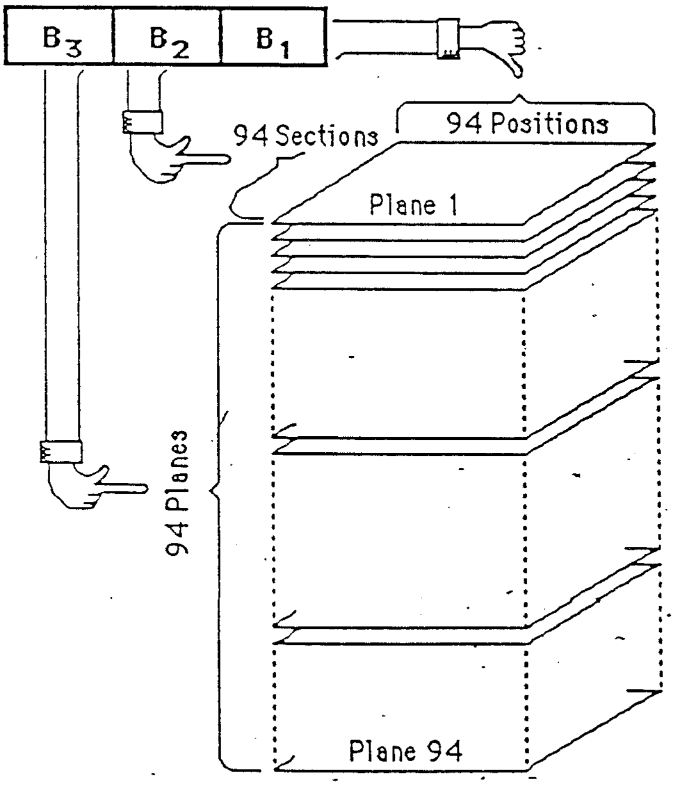


Figure 2 Three dimensional (94x94x94) Coding Structure

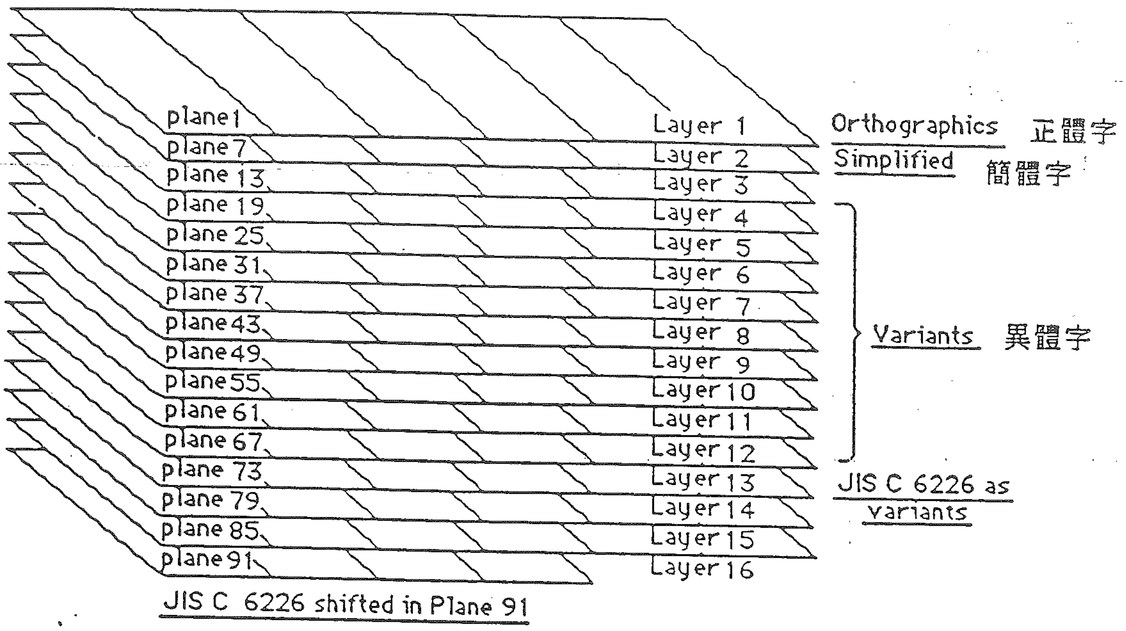


Figure 3 Sixteen LAYERS of the CCCII Coding Structure

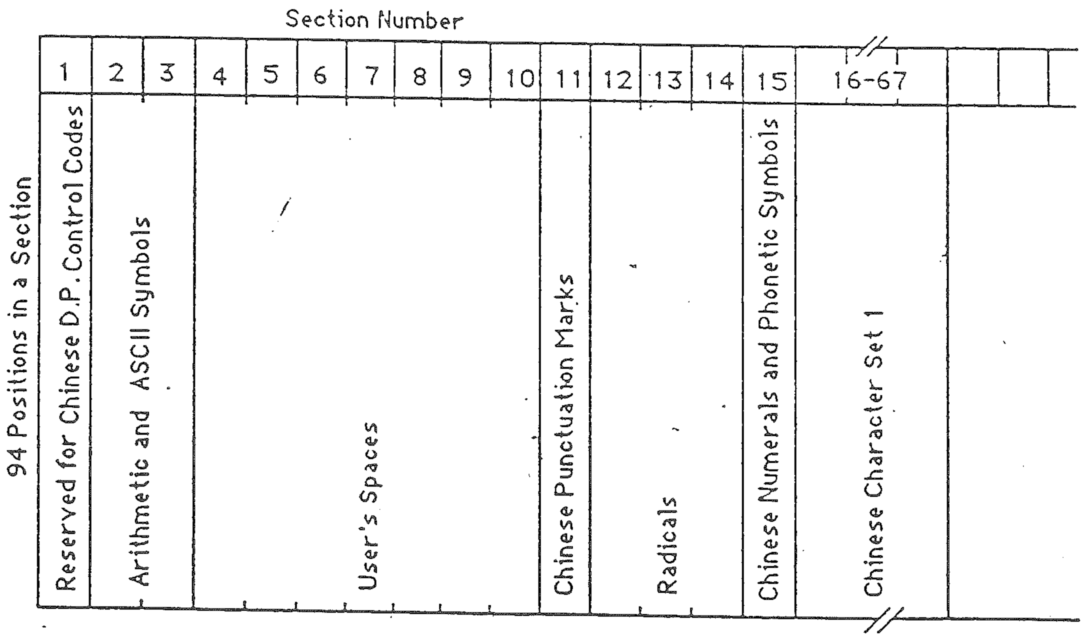


Figure 4 Structure of Each LAYER

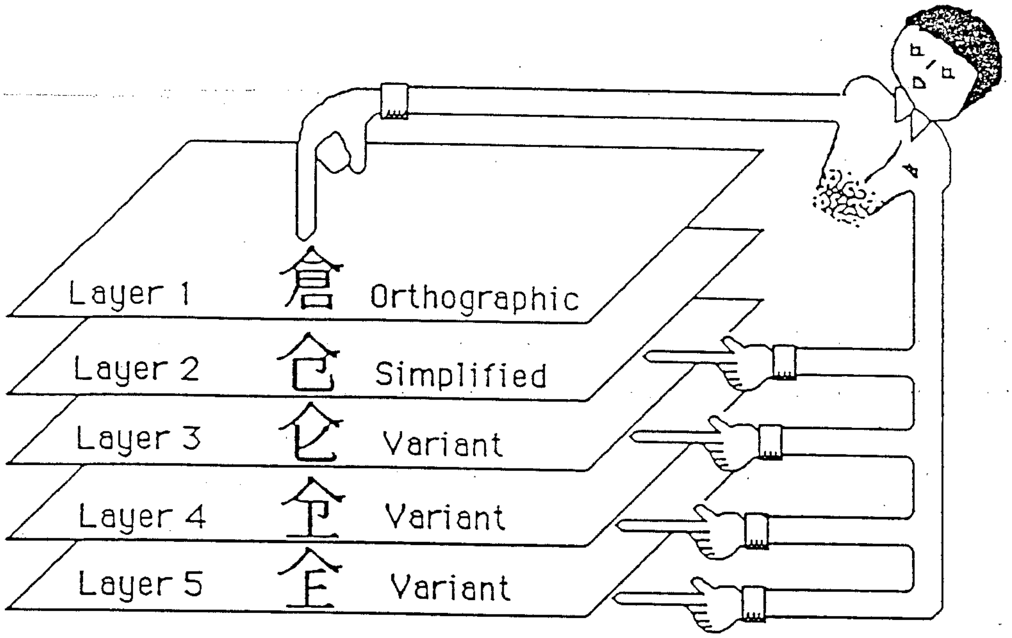


Figure 5 Relationship between Orthographic and Variant characters

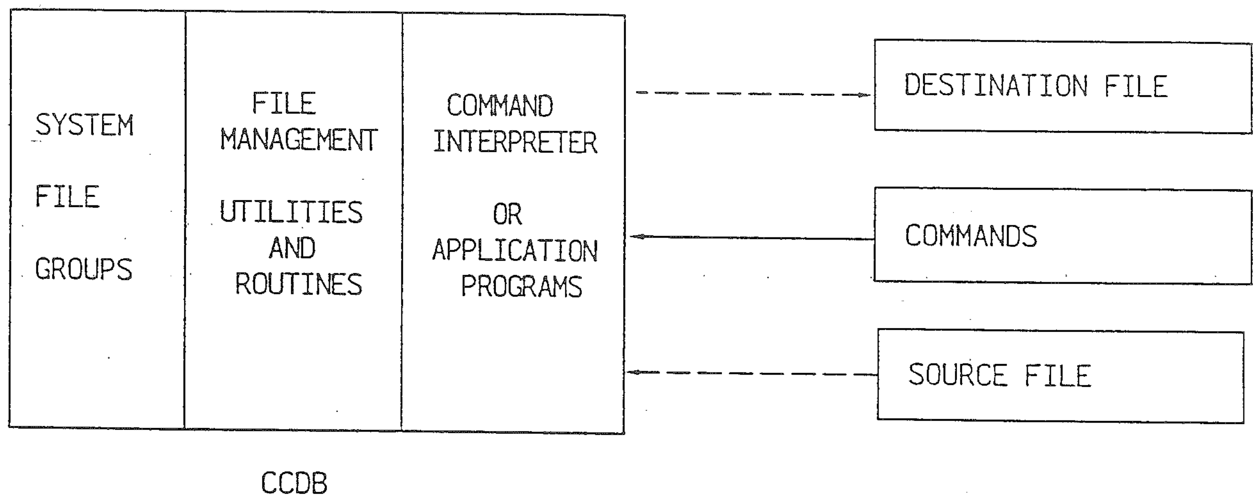
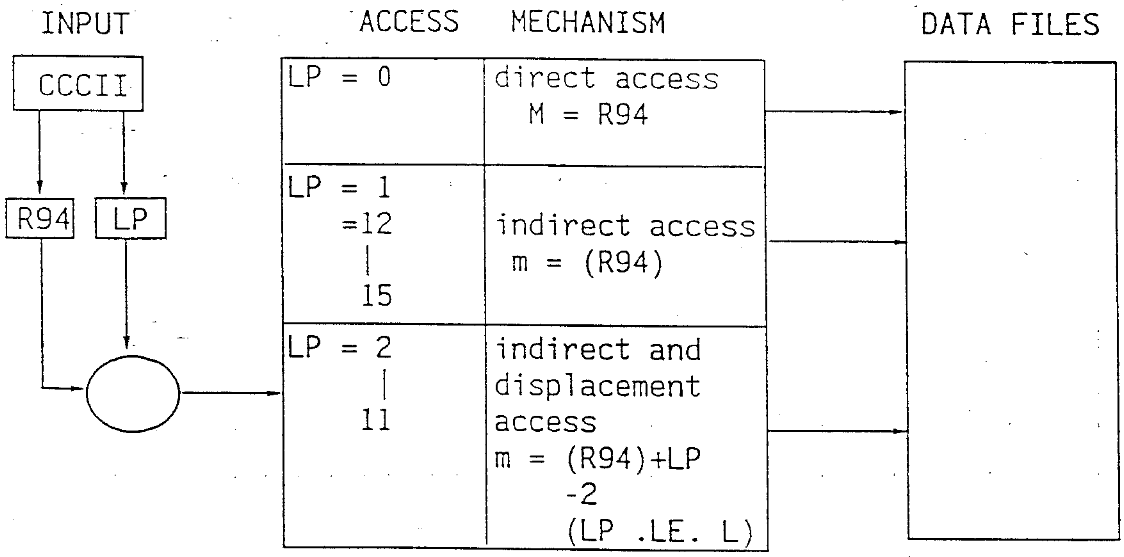


Figure 6 Software Structure of CCDB

CCCII	RAD	STK	EXP	VFM	PHC	FNT	ETL	OTC

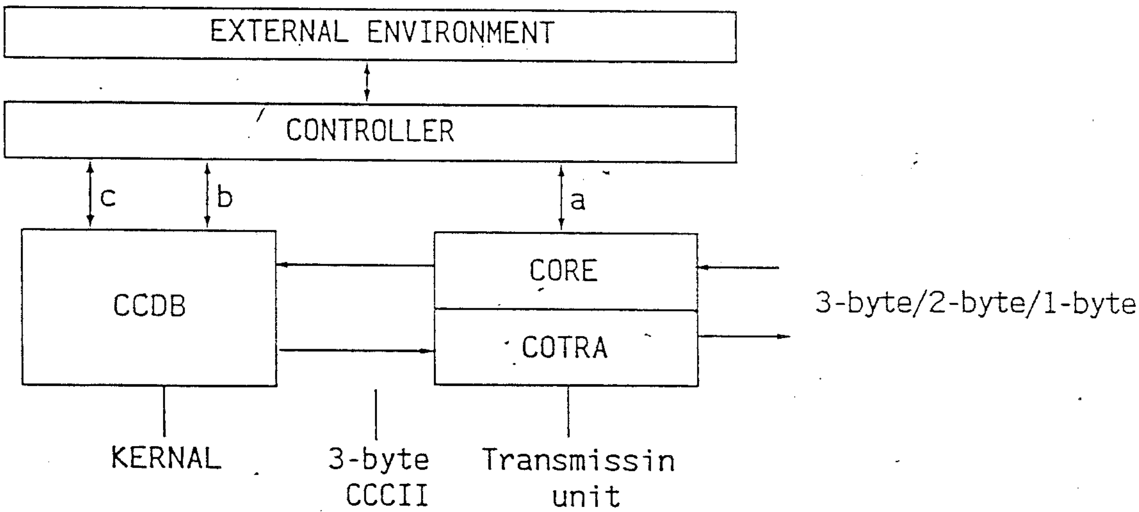
CCCII : Chinese Character Code for Information Interchange
RAD : Radical which a character belongs to
STK : Total stroke count
EXP : Component expression of a character
VFM : Variant forms of a character
PHC : Phonetic symbols of a character's pronounciation
FNT : Dot matrix representation of the FONT of a character
ETL : External or Indexing code of a character
OTC : Codes of different coding system

Figure 7 The Attributes provided by CCDB



LP : Layer indicator, from 0 to 15
L : The upper boundary of LP
m : The effective address of the character to be accessed
R94 : Condensed code of CCCII

Figure 8 The Access Mechanism of CCDB



a. The routine calls from controller to CORE/COTRA and STATUS from CORE/COTRA back to controller
b. The routine calls from controller to CCDB and the STATUS from CCDB back to controller
c. Data channels

Figure 9 The organization of CCCII Machine

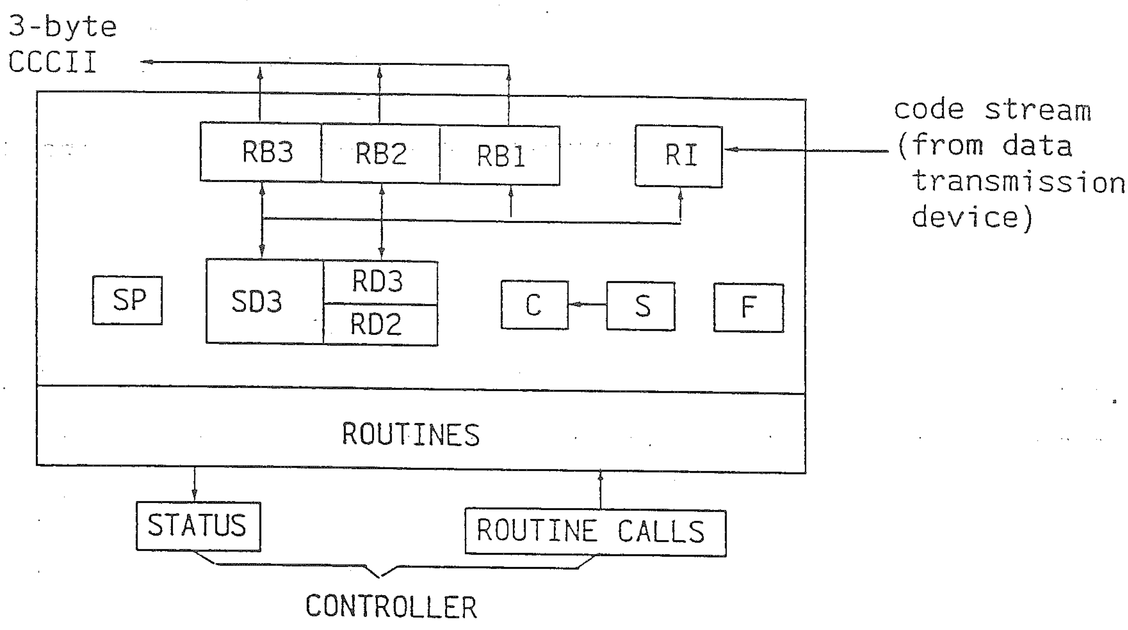


Figure 10 The Structure of CORE

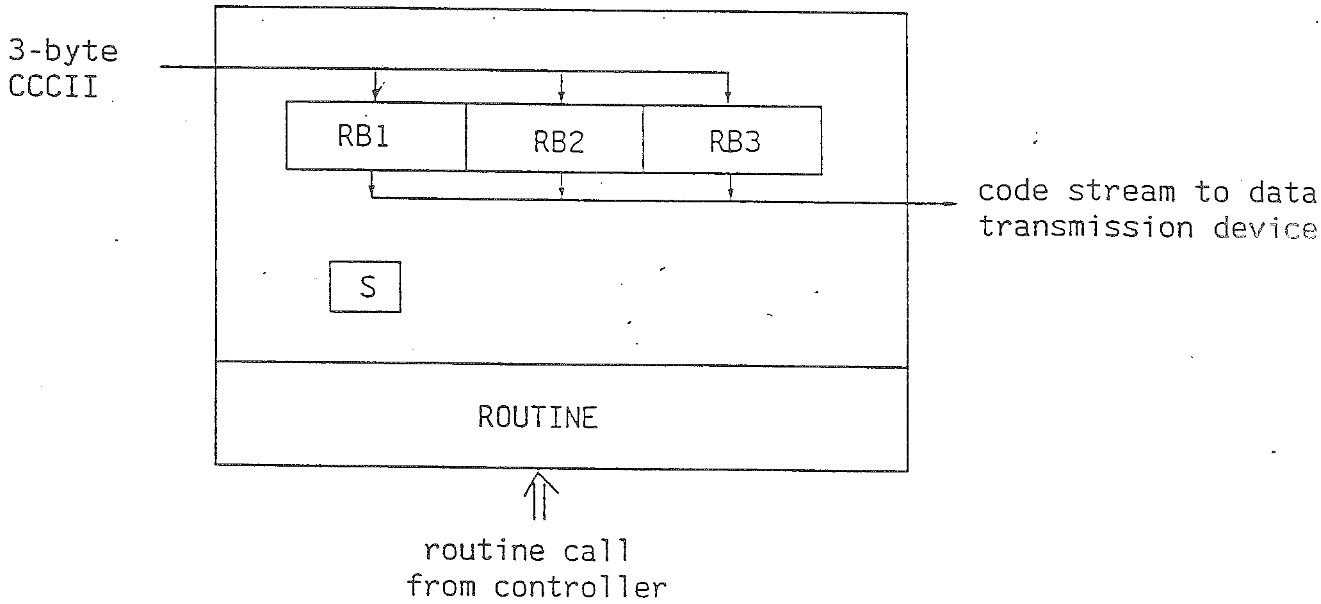


FIgure 11 The Structure of CONTA

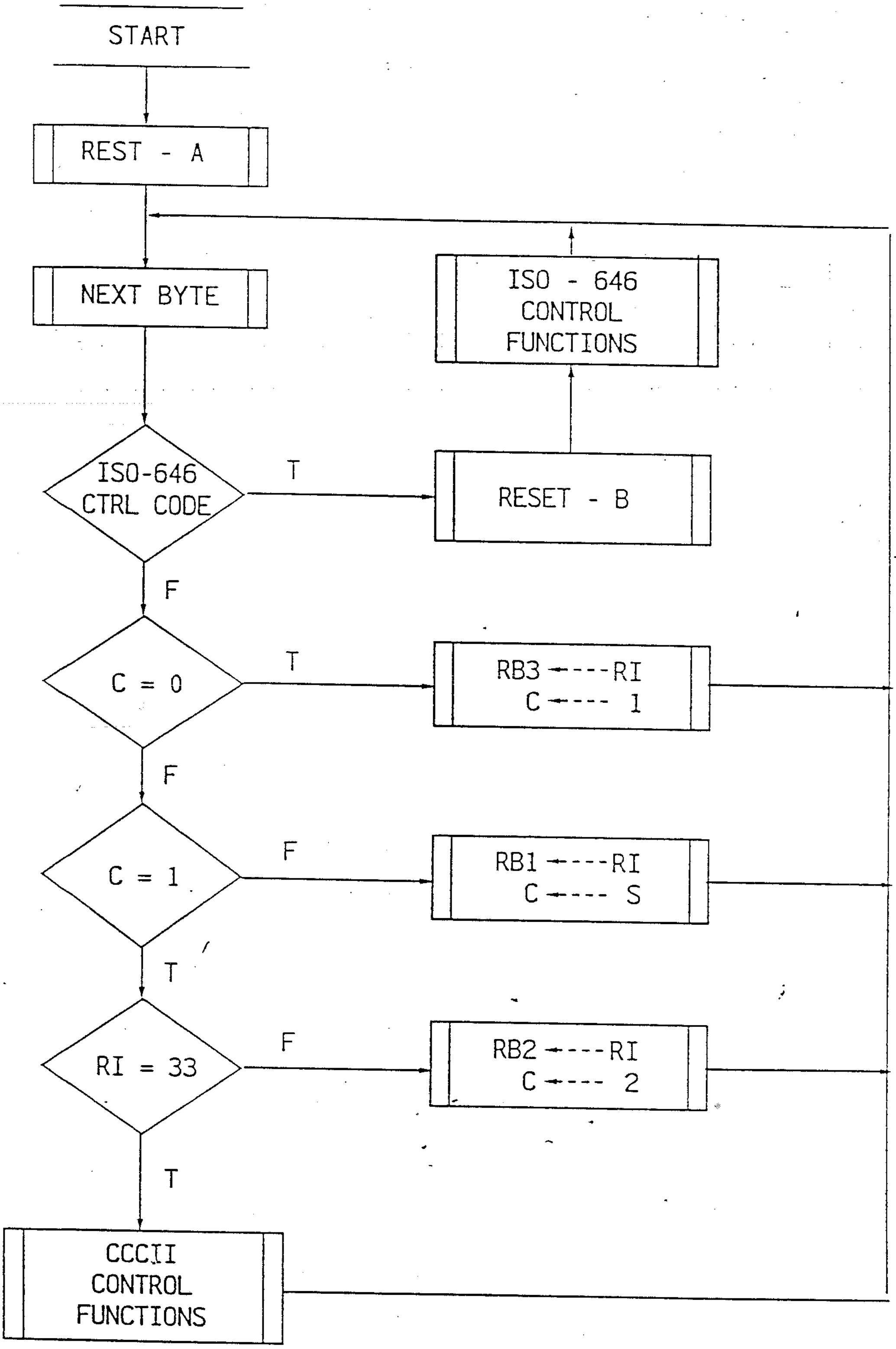


Figure 12 The Main Control Routine of CORE

S	OUTPUT	DESCRIPTION
0 ---► 1	PSL / PSU with parameter	3-byte -----► 2-byte
0 ---► 2	SSL with parameter	3-byte -----► 1-byte
1 ---► 0	HOM	2-byte -----► 3-byte
1 ---► 2	SSL with parameter	2-byte -----► 1-byte
2 ---► 0	ESC 2/4 4/1	1-byte -----► 3-byte

Figure 13 The relation between control code and S

Appendix A

*PSL (Plane-shift, locked) : B1=112

To set the B3 (plane) as default value, all the following CCCII's will have the same B3 and thus omitted. This will produce 2-byte (B2 and B1) CCCII's. PSL must be followed by 2-byte parameter, in which the first byte is ignored and the second byte is used as the plane indicator. The control function of PSL is explained as follow :

[B3],B2,B1;	[B3],33,112;	X,B3';	B2,B1
3-byte CCCII string or 2-byte CCCII string with old default value of B3	PSL	parameter	2-byte CCCII string with B3' default value

*PSR (Plane-shift, locked) : B1=113

Pop the last old default value of B3 from stack, if stack is blank, the PSR will be not reacted. The PSR has no parameter follows after, and it is accepted only under the case of 2-byte condition, if under the case of 3-byte condition, the PSR will be rejected.

*PSU (Plane-shift, unlocked) : B1=114

Similar to PSL. The different is that PSL will change the B3 permanently (or until another PSL with ddifferent plane value). But the PSU will only change the following character. The PSU is accepted only under the case of 2-byte condition, if under the case of 3-byte condition, the PSU will be rejected. The function of PSU is explained as follows:

B2,B1;	33,114;	X,B3';	B2,B1;	B2,B1+
2-byte CCCII	PSU	parameter	2-byte	2-byte CCCII

stream with default B3+		CCCII with (with B3')	stream with default B3+

*SSL (Section/Plane-shift, locked) : B1=115

Use to set the default values of B3 and B2. All the following CCCII's. SSL is applicable to 3-byte and 2-byte condition, the SSL must be followed by a 2-byte parameter, the first byte indicates the default value of B2, and the second byte indicates the default value of B1. The functions of SSL is explained as follows:

[B2],B2,B1;	[B3],33,115;	B2',B3';	B1
3-byte CCCII stream or 2-byte CCCII stream with old default value of B3	SSL	parameter	1-byte CCCII stream with default B3' and default B2

*SSU (Section/Plane-shift, unlocked) : B1=116

Similar to SSL, but only react on the character that followed after. SSU only applies under 2-byte condition, SSU will be rejected if under 3-byte condition. The functions of SSU is explained as follows:

B2,B1;+	33,116	B2',B3''	B1	B2,B1;+
2-byte CCCII stream with default B3+	SSU	parameter	1-byte CCCII with (with B3' and B2')	2-byte CCCII stream with default B3+

*HOM (Home) : B1=119

Hom is for deleting the defined B3 default value, to recover back to 3-byte or 2-byte, once the CCCII 3 bytes is reduced to single byte, it will need the ESC 2/4 4/1 to go back to 3 bytes codes.

Notes :

1. For all CCCII control codes, B3 is ot active and B2=33.
2. All the symbol used above, their definitions are as follows:

    , Used to delimit the byte, not appear in the code stream.

    ; Used to delimit the CCCII code, not appear in the code stream.

    [ ] Used to express B3 is not existed under 2-byte condition.

    _ Used to express CCCII stream, as the number of CCCII codes=0,1,2,...;

    X Used to express the byte value is 34-126.

    B3' and B2' Used each express B3 and B2 default value respectively.

    + Used to express 2-byte CCCII stream with specified default B3+.

```
procedure F_RESET;
  begin
    F:=NO_ERROR
  end;

procedure RESET_A;
  begin
    S:=0;
    C:=0;
    SP:=BOTTOM;
    F_RESET
  end;

procedure RESET_B;
  begin
    if C<>S then F:=IMCOMPLETE_CCCII;
    C:=S
  end;

procedure RESET_C;
  begin
    C:=S;
    F:=PARAMETER_ERROR
  end;

procedure NEXT_BYTE;
  begin
    repeat WAIT_LOOP { CORE idle }
    until EVENT(RI) { next byte into RI }
  end;

procedure ACCEPT_BYTE;
  begin
    NEXT_BYTE;
    if RI<33 or RI>126
      then F:=BYTE_ERROR
      else F:=NO_ERROR
  end;

procedure PSL;
  begin
    ACCEPT_BYTE;
    ACCEPT_BYTE;
    if F=NO_ERROR
      then
        begin
          RB3:=RI;
          SD3[SP]:=RI;
          SP:=SP+1;
          S:=1;
          C:=1
        end
      else RESET_C;
  end;

procedure PSR;
  begin
    if SP>BOTTOM
      then
        begin
          case S of
            0        :
F:=PSR_REJECT;
            1 : begin
                  SP:=SP-1;
                  RB3:=SD3[SP];
                  C:=S;
                  F_RESET
                end
          end
      else F:=STACK_EMPTY
  end;

procedure PSU;
 begin
   ACCEPT_BYTE;
```

```
  ACCEPT_BYTE;
  if F=NO_ERROR
    then
      begin
        RB3:=RI;
        ACCEPT_BYTE;
        if F=NO_ERROR
          then
            begin
              RB2:=RI;
              ACCEPT_BYTE;
              if F=NO_ERROR
                then
                  begin
                    RB1:=RI;
                    write(RB3,RB2,RB1);
                    RB3:=SD3[SP-1]
                  end
                else F:=PSU_FAIL
            end
          else F:=PSU_FAIL
      end
    else F:=PSU_FAIL;
  C:=S
end;

procedure SSL;
  begin
    ACCEPT_BYTE;
    if F=NO_ERROR
      then
        begin
          RB2:=RI;
          RD2:=RI;
          ACCEPT_BYTE;
          if F=NO_ERROR
            then
              begin
                RB3:=RI;
                RD3:=RI;
```

```
                S:=2;
                C:=2
              end
            else RESET_C
        end
      else RESET_C
  end;

procedure SSU;
  begin
    ACCEPT_BYTE;
    if F=NO_ERROR
      then
        begin
          RB2:=RI;
          ACCEPT_BYTE;
          if F:=NO_ERROR
            then
              begin
                RB3:=RI;
                ACCEPT_BYTE;
                if F=NO_ERROR
                  then
                    begin
                      RB1:=RI;
                      write(RB3,RB2,RB1);
                      RB3:=RD3;
                      RD2:=RD2
                    end
                  else F:=SSU_FAIL
              end
            else F:=SSU_FAIL
        end
      else F:=SSU_FAIL
    C:=S
  end;
```

Appendix C.

```
procedure HOM;
  begin
    RESET_A
  end;
  begin
    ACCEPT_BYTE;
    if F=NO_ERROR
      then
        begin
          case RI of
            33 : ...... ;
                  :
                  :
            112 : PSL;
            113 : PSR;
            114 : PSU;
            115 : SSL;
            116 : SSU;
                  :
            119 : HOM;
                  :
                  :
            126 : ......
          end
  end;
```

```
procedure ESC_SEQ;
  begin
    NEXT_BYTE;
    if RI=36
      then
        begin
          NEXT_BYTE;
          if RI=65 then RESET_A
                   else ......
        end
      else ......
  end;
```

```
procedure ISO646_CONTROL;
  begin
    if RI=127
      then DELETE
      else
        begin
          case RI of
            0 : NULL;
                  :
                  :
            27 : ESC_SEQ;
                  :
                  :
            32 : SPACE
          end
  end;
```

```
procedure CORE;
  begin
    RESET_A;
    repeat
      NEXT_BYTE;
      if RI<=32 or RI=127
        then
          begin
            RESET_B;
            ISO646_CONTROL
          end
        else
          begin
            case C of
              0 : begin
                    RB3:=RI; C:=1
                  end;
              1 : if RI=33
                    then CCCII_CONTROL
                    else
                      begin
                        RB2:=RI; C:=2
                      end;
              2 : begin
                    RB1:=1; C:=S;
```

```
                write(RB3,RB2,RB1)
              end
          end
    until SYSTEM_BREAK
  end;


procedure CORE;
  begin
    RESET_A;
    repeat
      NEXT_BYTE;
      if RI<=32 or RI=127
        then
          begin
            RESET_B;
            ISO646_CONYROL
          end
        else
          begin
            case C of
              0 : begin
                    RB3:=RI; C:=1
                  end;
              1 : if RI=33
                    then CCCII_CONTROL
                    else
                      begin
                        RB2:=RI; C:=2
                      end
              2 : begin
                    RB1:=1; C:=S;
                    write(RB3,RB2,RB1)
                  end
          end
    until SYSTEM_BREAK
  end;


procedure CORTA;
  begin
    case S of
      0 : write(RB3,RB2,RB1);
      1 : write(RB2,RB1);
      2 : write(RB1)
    end
  end;


procedure OUT_LOCK_1(B3,B2);
  begig
    case S of
      0 : begin
            write(33,33,115); { SSL code }
            write(B2,B3)  { parameter }
          end;
      1 : begin
            write(33,115); { SSL code }
            write(B2,B3) { parameter }
          end
    end;
    S:=2
  end;


procedure OUT_LOCK_2(B3);
  begin
    case S of
      0 : begin
            write(33,33,112); { PSL code }
            write(34,B3); { parameter }
            S:=1
          end;
      1 : begin
            write(33,112); { PSL code }
            write(34,B3) { parameter }
          end
    end
  end;


procedure OUT_UNLOCK_1(B3,B2,B1)
  begin
    if S=1
```

```
      then
        begin
          write(33,116); { SSU code }
          write(B2,B3,B1)
        end
  end;


procedure OUT_UNLOCK_2(B3,B2,B1);
  begin
    case S of
      0 : begin
            write(33,33,114); { PSL code }
            write(34,B3,B2,B1) { parameter & 2-byte code }
          end;
      1 : begin
            write(33,114); { PSU code }
            write(34,B3,B2,B1) { parameter & 2-byte code }
          end
    end
  end;


procedure OUT_LOCK_3;
  begin
    case S of
      0 : write(33,119); { HOM code }
      1 : write(27,36,65) { ESC 2/4 4/1 }
    end
  end;


procedure PSR_CODE;
  begin
    write(33,113) { PSR code }
  end;
```