THE INTERNATIONAL TELEGRAPH AND TELEPHONE CONSULTATIVE COMMITTEE

CCITT

SIXTH PLENARY ASSEMBLY

GENEVA, 27 SEPTEMBER - 8 OCTOBER 1976

ORANGE BOOK

VOLUME VI.4

PROGRAMMING LANGUAGES FOR STORED-PROGRAMME CONTROL EXCHANGES

Published by the
INTERNATIONAL TELECOMMUNICATION UNION
GENEVA, 1977

ISBN 92-61-00421-0

FUNCTIONAL SPECIFICATION AND DESCRIPTION LANGUAGE (SDL)

TABLE OF CONTENTS

Recommendation Z.101

1. GENERAL EXPLANATION OF THE SPECIFICATION AND DESCRIPTION LANGUAGE (SDL)

This Recommendation deals with the presentation of the functional specification and of the description of the internal logic processes in stored programmed control (SPD) telephone exchanges.

1.1 *Introduction*

1.1.1 *Methods of presentation*

The methods of presentation of functional specifications and of descriptions of internal logic processes in SPC telephone exchanges can be subdivided into the following categories:

- narrative methods (natural language and numerical information supported by drawings and lists etc.);
- formalized presentation methods.

The narrative methods, which can be used to a large extent for both specifications and descriptions of SPC telephone switching systems, need no standardization by the CCITT.

Considering the formalized methods of presentation, the subject of this Recommendation is a graphical method, based on state transition diagrams, using the symbols and rules of the Functional Specification and Description Language (SDL) described in the following sections. (It may be noted that some processes of an SPC switching system may require specifications and/or descriptions by methods other than in this Recommendation.) Wherever appropriate the symbols of the SDL have been taken from the ISO standard for flow charts (ISO/R 1028-1969).

1.1.2 *General objective*

The objective of the SDL is to provide a standardized method of presentation:

- that is easy to learn, to use and to interpret in relation to the needs of operating organizations;
- that provides unambiguous specifications and/or descriptions for tendering and ordering;
- that provides the capability of meaningful comparisons between competitive types of SPC telephone exchanges;
- that is open-ended to be extended to cover new developments.

1.1.3 *Area of application*

The main area of application covers all types of SPC telephone switching systems. Within these systems the following functions are included amongst others:

- call processing (e.g. call handling, routing, signalling, metering, etc.);
- maintenance and fault treatment (e.g. alarms, automatic fault clearing, configuration control, routine tests, etc.);
- system control (e.g. overload control, modification and extension procedures, etc.).

1.2 *Framework*

1.2.1 *Basic definitions*

a) To clarify the meaning of terms used in the SDL, a number of definitions are given below.

b) Some of the terms defined below have been in use in other fields and will have connotations related to those fields. Care should therefore be exercised by those concerned with the SDL that their use and understanding of such terms is in accord with the definitions contained in this section.

1.2.2 *Specifications and descriptions*

a) The requirements of a system are defined in a *specification* of that system and the implementation of those requirements in a *description* of the system.

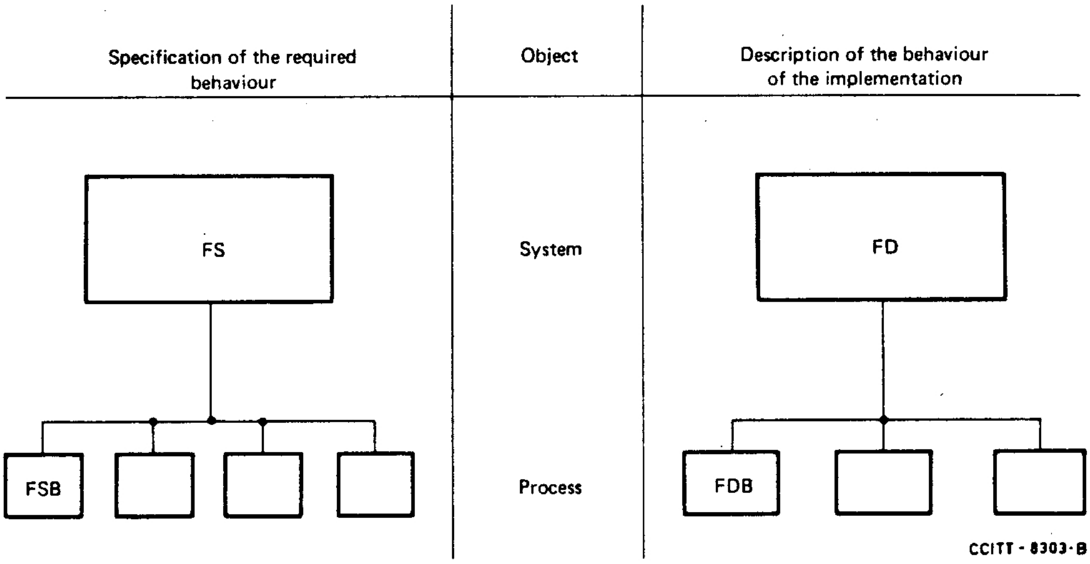
b) Both *specifications* and *descriptions* consist of two parts: the former comprises *general parameters* required of the system and the *functional specifications* (FS) of its required behaviour; the latter comprises *general parameters* of the system as implemented and the *functional description* (FD) of its actual behaviour.

c) The *general parameters* in both cases relate to such matters as temperature limits, transmission limits, construction, exchange capacity, grade of service, etc.

1.2.3 *Functional specifications and functional descriptions* (See Figure 1/Z.101)

The *functional specification* (FS) of a system is a specification of the total functional requirements of that system from all significant points of view.

The *functional description* (FD) describes the actual behaviour of the implementation of those functional requirements in terms of the internal structure and logic processes within the system.


Specification of the required behaviour
Object
Description of the behaviour of the implementation
FS
System
FD
FSB
Process
FDB
CCITT - 8303-B

FS = functional specification
FD = functional description
FSB = functional specification block
FDB = functional description block

Note. – The partitioning of an FS into FSBs for a particular system does not necessarily correspond to the partitioning of the FD into FDBs for the same system.

FIGURE 1/Z.101 – **Partitioning**

1.2.4 *Functional specification blocks and functional description blocks*

a) *functional specification blocks* (FSB) and *functional description blocks* (FDB), as tools for specification or description, are both entities of manageable size and relevant internal relationship. (See Figure 1/Z.101.)

A functional specification block specifies the desired behaviour of one or more processes.

A functional description block describes the means by which the required behaviour of processes is achieved.

b) The behaviour of a process is described in terms of *inputs, states, transitions, decisions, tasks* and *outputs.*

c) FSBs and/or FDBs might relate to such processes as call handling, traffic recording, signalling, switchboard operations, man-machine procedures, queueing, etc.

d) FSBs and FDBs are formed by partitioning their parent FSs and FDs. There may be more than one way of partitioning any FS or FD.

e) Both FSBs and FDBs may be partitioned further to form subsidiary FSBs and FDBs so that the hierarchical structure is extended downwards.

f) The partitioning of an FS into FSBs for a particular system does not necessarily correspond to the partitioning of the FD into FDBs for the same system. The boundaries of FSBs and FDBs do not necessarily correspond at any hierarchical level.

g) At any given hierarchical level a process appears only in one block.

h) Every process specified or described by a *block* should have a well-defined boundary across which *signals* pass as explicitly indicated. Interfaces between FSBs, and between FDBs, should similarly be well defined.

1.3 *Basic concepts for the SDL*

The SDL is based on the following definitions:

1.3.1 *Signals*

a) A *signal* is a flow of data conveying information to a process.

b) A *signal* may be either in hardware or in software form.

c) If the information flow is from a process described by a *block* to a process described by another *block* it is an *external signal.* If the flow is between processes described by the same *block* it is an *internal signal.*

1.3.2 *Inputs*

An *input* is an incoming *signal* which is recognized by a process. (It is not to be confused with input as applied to normal data processing.)

In accordance with the definition of *signals*, an input can be internal or external.

1.3.3 *States*

A *state* is a condition in which the action of a process is suspended awaiting an *input.*

1.3.4 *Transitions*

A *transition* is a sequence of actions which occurs when a process changes from one *state* to another in response to an *input.*

A process can be either in one of its states or in a transition at any one instant.

1.3.5 *Outputs*

An *output* is an action within a *transition* which generates a *signal* which in turn acts as an *input* elsewhere. (It is not to be confused with output as applied to normal data processing.)

In accordance with the definition of signals an output can be either internal or external.

1.3.6 *Decisions*

A *decision* is an action within a *transition* which asks a question to which the answer can be obtained at that instant and chooses one of several paths to continue the *transition.*

1.3.7 *Tasks*

A *task* is any action within a *transition* which is neither a *decision* nor an *output.*

Recomendation Z.102

2. SYMBOLS AND RULES

2.1 *General*

Each process represented consists of several states and the various transitions between them. An input will trigger the leaving of a state to travel along a transition, executing tasks, generating outputs and branching on decisions until another state is reached. The representations may be linear, with multiple appearances of a single state if convenient, or may be of mesh form or any combination of the two.

The concepts of state, input, task, output and decision are represented by their respective symbols. The appropriate interconnection of such symbols by flow lines represents the logical flow of a process.

2.2 *Symbols*

The recommended symbols appear in Figure 2/Z.102.

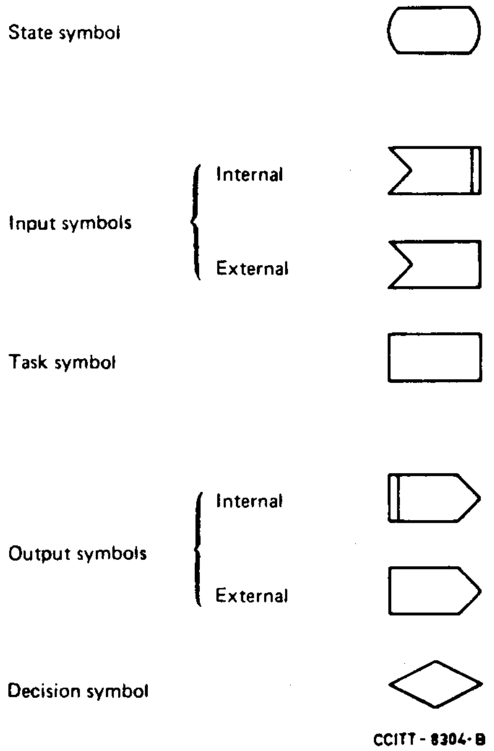


FIGURE 2/Z.102 – Recommended symbols

2.3 *Sequence rules*

Certain rules for the use of the symbols and their interconnections are required to ensure a valid representation of a process. For the purpose of these rules, *follow* means *follow immediately.*

2.3.1 A state symbol may only be followed by one or more input symbols.

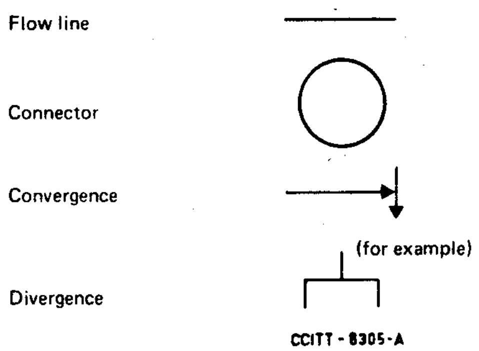
2.3.2 Each input symbol follows one and only one symbol which must be a state symbol.

2.3.3 Each input symbol is followed by one and only one symbol, which may be any symbol except another input symbol.

2.3.4 Each task or output symbol is followed by one and only one symbol, which may be any symbol except an input symbol.

2.3.5 A decision symbol must be followed by two or more symbols, which may not be input symbols.

2.4 *Flow lines and connectors*



CCITT - 8305-A

2.5 *Flow line rules*

2.5.1 Every symbol is connected to the symbol it follows by a solid flow line.

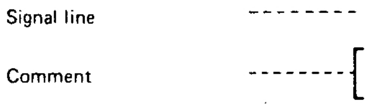
2.5.2 A solid flow line may be broken by a pair of associated connectors, with the flow assumed to be from the out-connector to its associated in-connector.

2.5.3 Where two or more symbols are followed by a single symbol the flow lines leading to that symbol converge. This convergence may appear as one flow line flowing into another or as more than one out-connector associated with a single in-connector.

2.5.4 Where a symbol is followed by two or more other symbols a flow line leading from that symbol may diverge into two or more flow lines.

2.5.5 Arrow heads are required whenever two flow lines converge and whenever a flow line enters an out-connector or a state symbol. Arrow heads are prohibited on flow lines entering input symbols.

2.6 *Annotations*



CCITT - 8306

2.7 *Annotation rules*

2.7.1 Where an output and an associated input symbol represent a signal from one process to another a dashed line from one symbol to the other may be included to indicate the association.

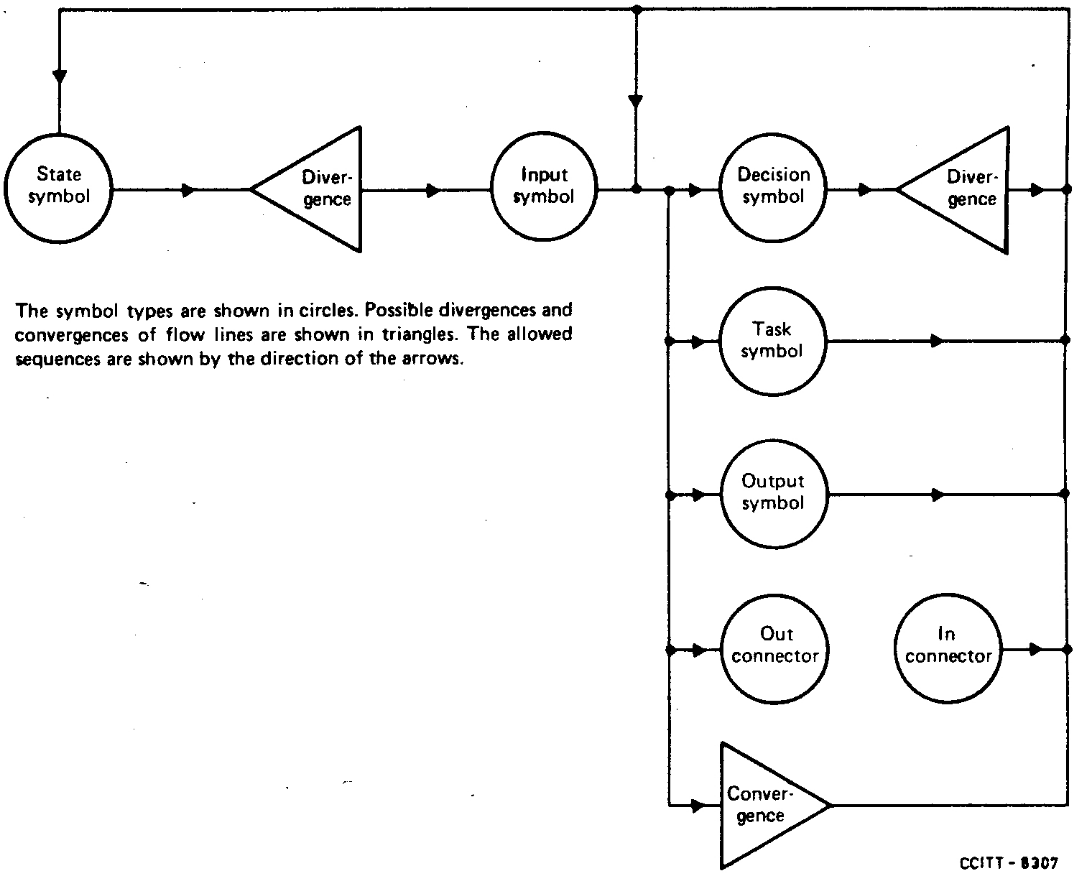
These dashed lines representing signals may diverge, converge or be broken by connectors.

2.7.2 Comments may be enclosed by a single square bracket connected by a dashed line to any symbol or flow line.

2.8 *Connectivity diagram*

Figure 3/Z.102 summarizes the sequence rules and flow line rules stated in the text.

Note. – Divergence and convergence both include the trivial case of a single continuity flow line.



The symbol types are shown in circles. Possible divergences and convergences of flow lines are shown in triangles. The allowed sequences are shown by the direction of the arrows.

FIGURE 3/Z.102 – Connectivity diagram

2.9 *Draughting conventions*

2.9.1 All symbol boxes of the same type shall preferably be of the same size within any one diagram.

2.9.2 Mirror images of input and output symbols are allowed.

2.9.3 Flow lines are horizontal or vertical and have sharp corners.

2.9.4 Flow lines that cross have no logical relationship.

2.9.5 The preferred aspect ratio of symbols is 2:1.

2.9.6 The text associated with the symbols should be placed within those symbols where practicable.

Recommendation Z.103

3. OPTIONAL USE OF PICTORIAL ELEMENTS WITHIN STATE SYMBOLS [1]

3.1 *General*

3.1.1 The use of pictorial elements within a state symbol forms an optional part of the SDL.

Such pictorial elements can provide advantages when applied to certain functional specifications and functional descriptions, resulting in a more compact and less verbal diagram.

3.1.2 With pictorial elements each state is represented by a state symbol containing a state picture and a state identifier (normally consisting of a state number and a state title) with the format shown in Figure 4/Z.103.

3.1.3 The total processing involved when going from one state to the following state is that required to effect the changes in the state pictures, together with the processing indicated in any decisions, outputs or tasks appearing in the transition between the states.

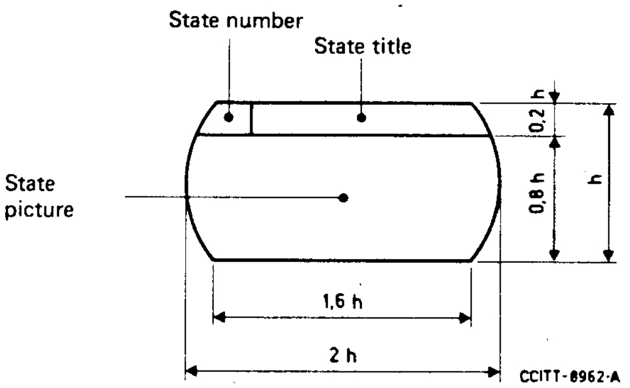


FIGURE 4/Z.103 – Recommended format for a state symbol with pictorial elements

Recommendation Z.104 (Has not yet been defined)

[1] The standardization of symbols for pictorial elements is the subject of further study in new Question 7/XI. The preliminary results from the Study Period 1973-1976 are presented in Annex 2 to the new Question 7/XI, Contribution COM XI-No. 1.

ANNEX TO RECOMMENDATIONS Z.101-Z.103

Examples of the use of SDL

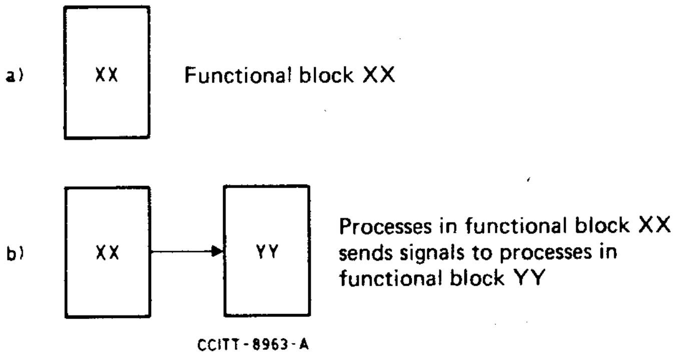
A. *General introduction*

A.1 *General*

The examples in this section are intended to illustrate the versatility of the SDL in application to the specification (before design) and the description (after design) of several processes typical of SPC switching systems.

A.2 *Functional block interactions*

To provide a framework in which each example can be understood, the concepts of Recommendation Z.101 have been applied in the form of a simple diagram of the interactions between functional blocks for each example. The interpretation of the diagrams is as follows:


a)
XX
Functional block XX
b)
XX
YY
Processes in functional block XX sends signals to processes in functional block YY
CCITT-8963-A

A.3 These examples have been designed to show the use of the SDL, and are not international specifications.

B. *Figures and examples*:

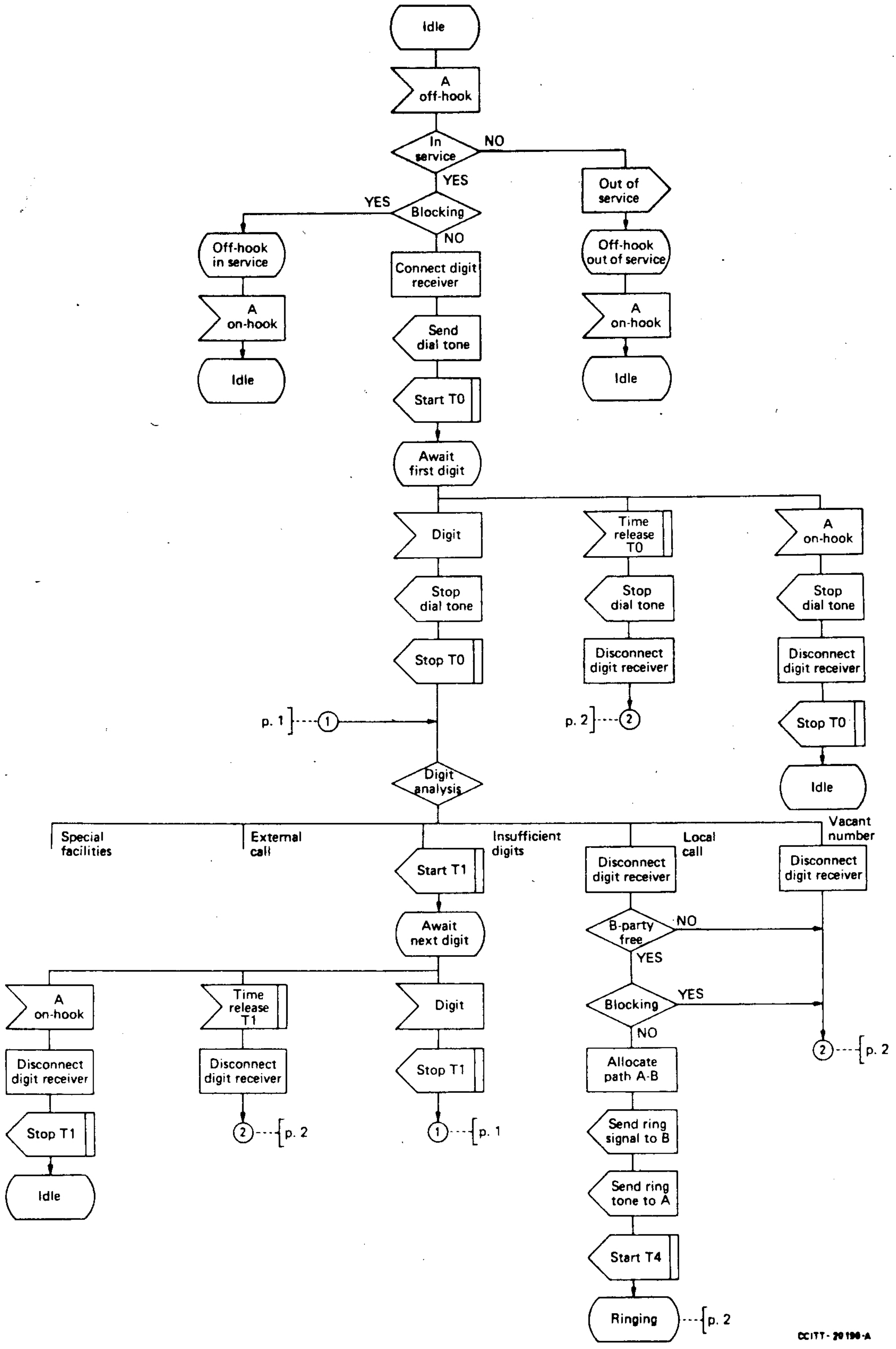


FIGURE 1 – Example 1 of a local call handling process, page 1

For notes to this Figure, see end of this Annex.

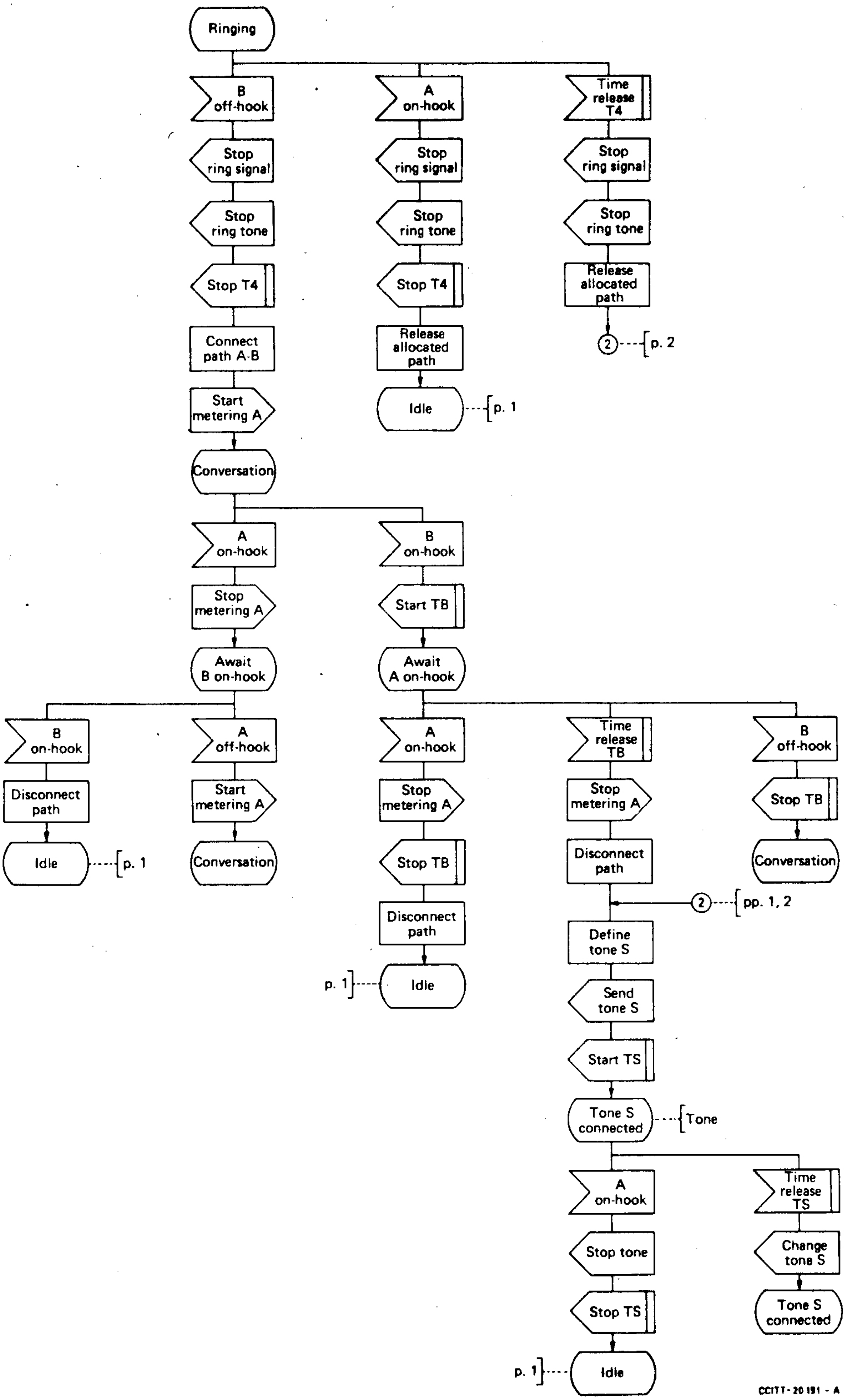


FIGURE 1 – **Example 1 of a local handling process, page 2**

For notes to this Figure, see end of this Annex.

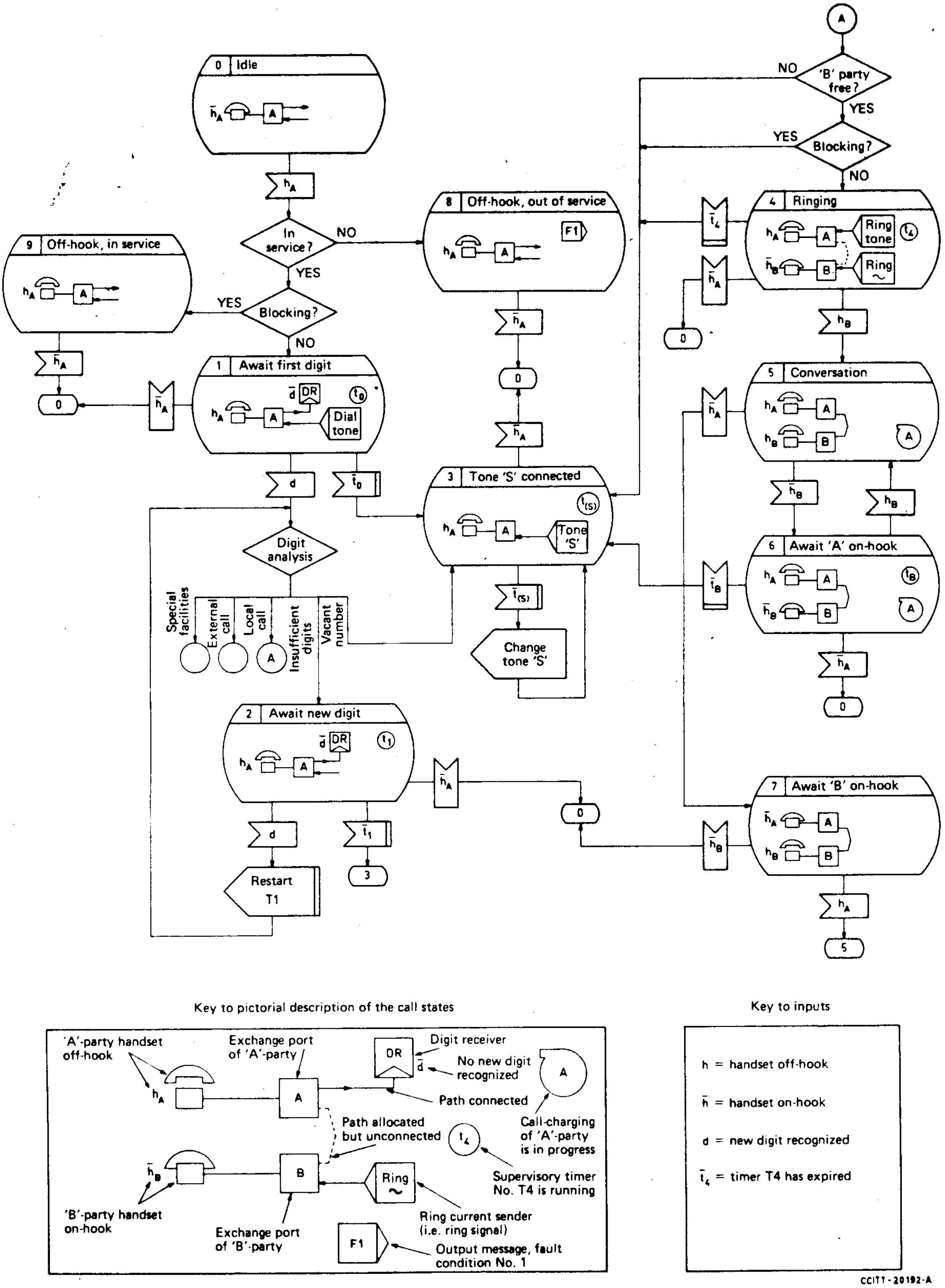
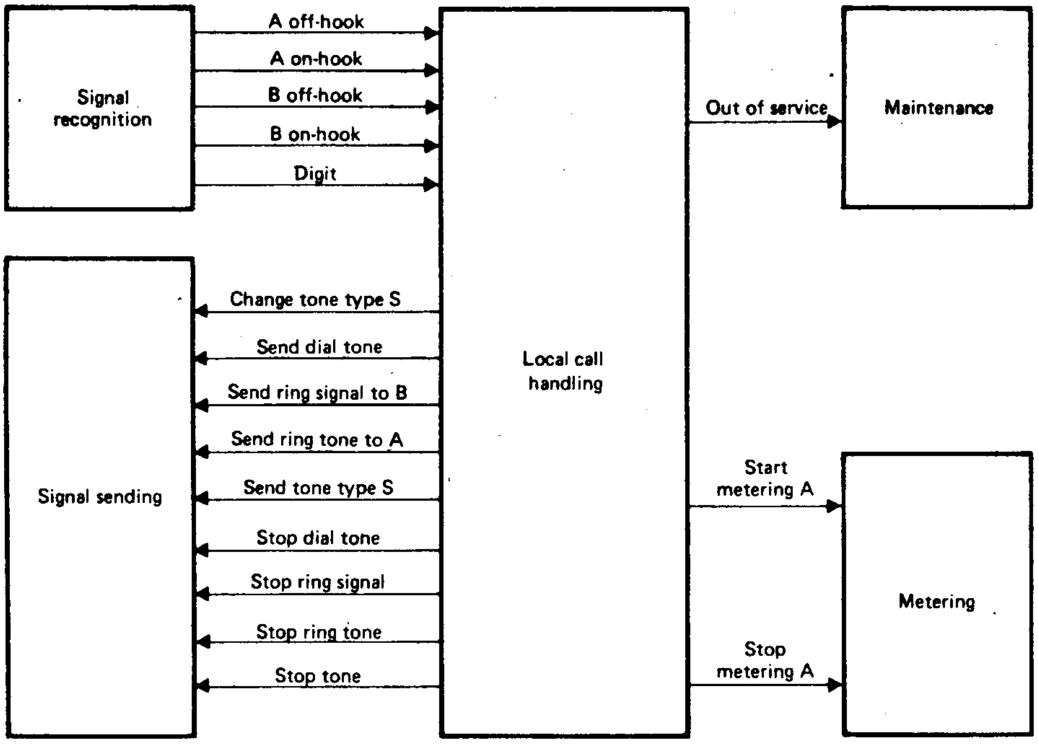


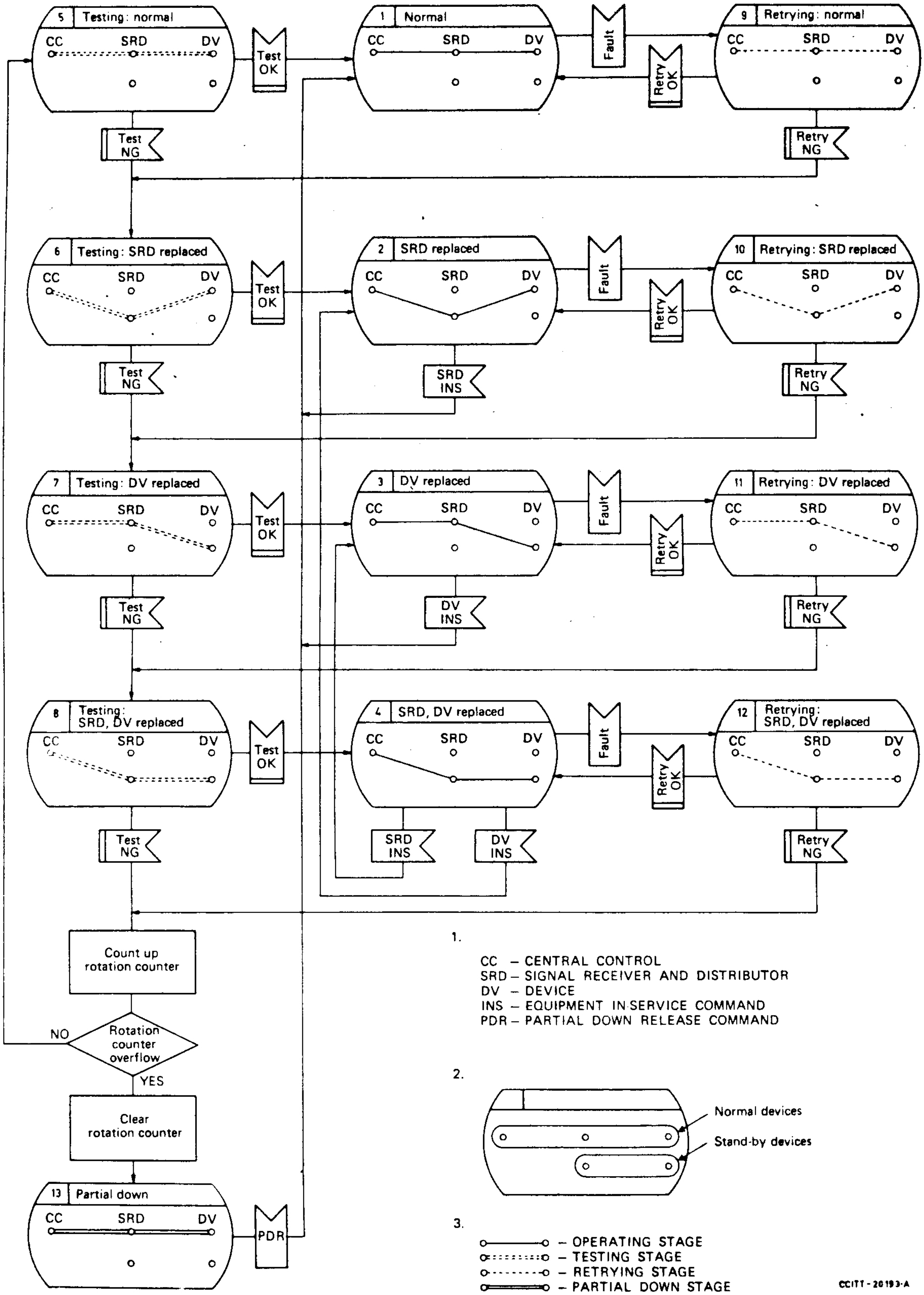
FIGURE 2 – Example 2 of a local call handling process, using state pictures

For notes to this Figure, see end of this Annex.



CCITT - 8964

FIGURE 3 – **Functional block interaction, for Examples 1 and 2**



1.

CC – CENTRAL CONTROL
SRD – SIGNAL RECEIVER AND DISTRIBUTOR
DV – DEVICE
INS – EQUIPMENT IN-SERVICE COMMAND
PDR – PARTIAL DOWN RELEASE COMMAND

2.

3.

– OPERATING STAGE
– TESTING STAGE
– RETRYING STAGE
– PARTIAL DOWN STAGE

FIGURE 4 – Example 3, a system configuration process during fault conditions using state pictures

For notes to this Figure, see end of this Annex.

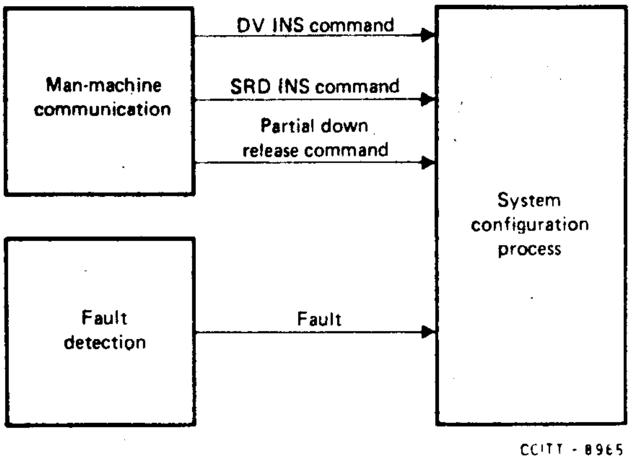

Man-machine communication
DV INS command
SRD INS command
Partial down release command
System configuration process
Fault detection
Fault
CCITT - 8965

FIGURE 5 – Functional block interaction for Example 3

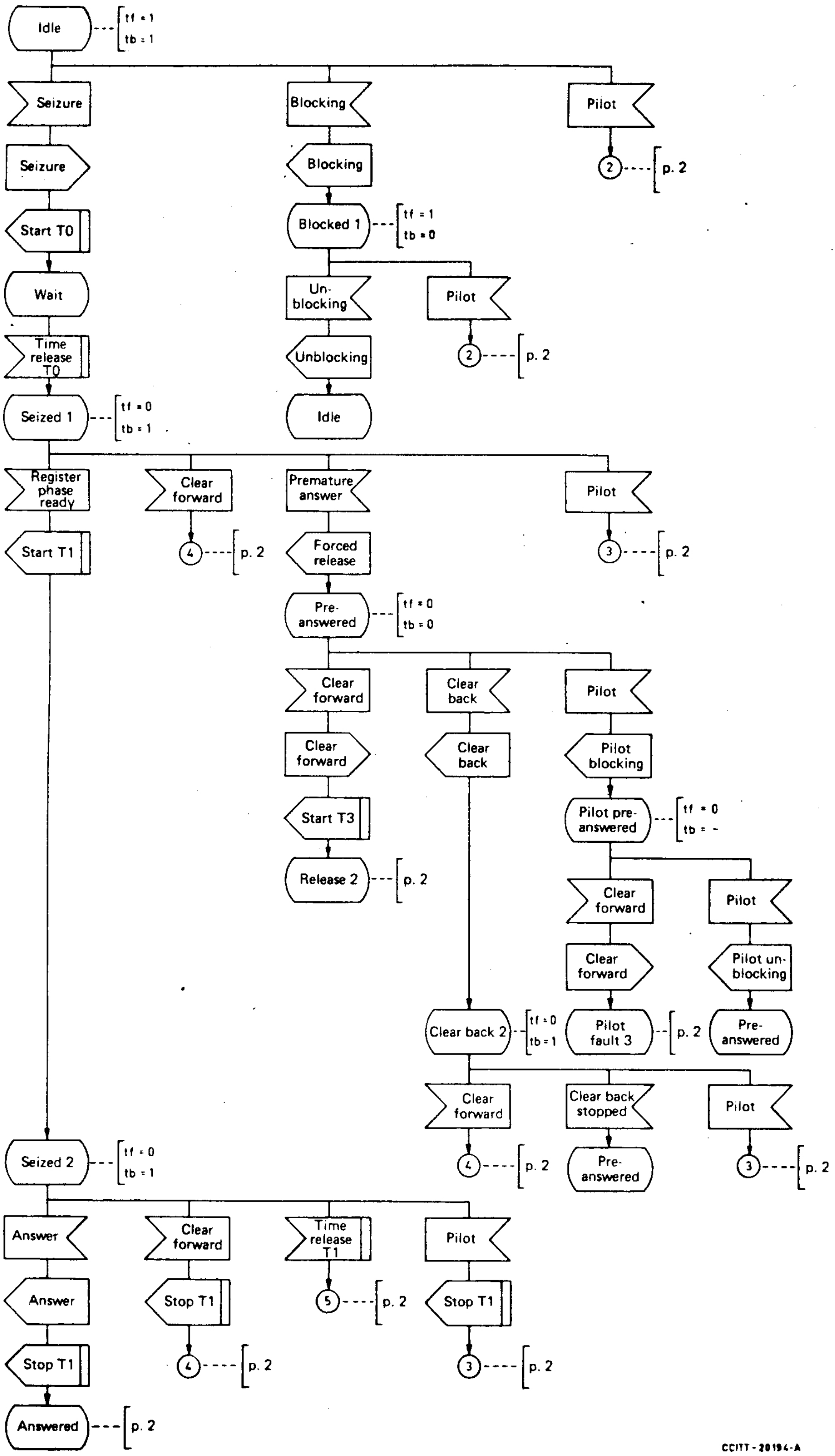


FIGURE 6 – Example 4, an R2 outgoing line-signalling process, page 1

For notes to this Figure, see end of this Annex.

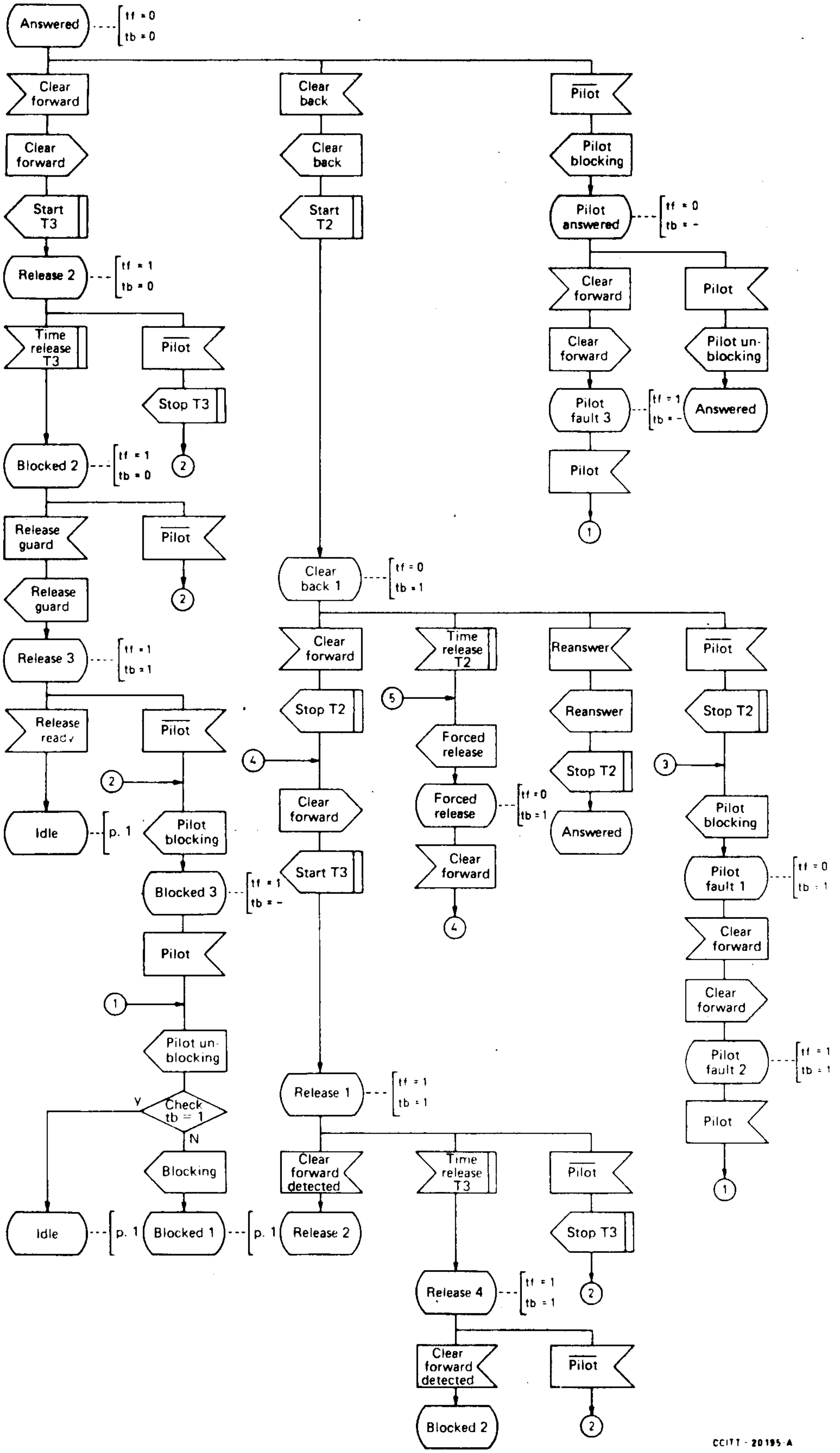


FIGURE 6 – **Example 4, an R2 outgoing line-signalling process, page 2**

For notes to this Figure, see end of this Annex.

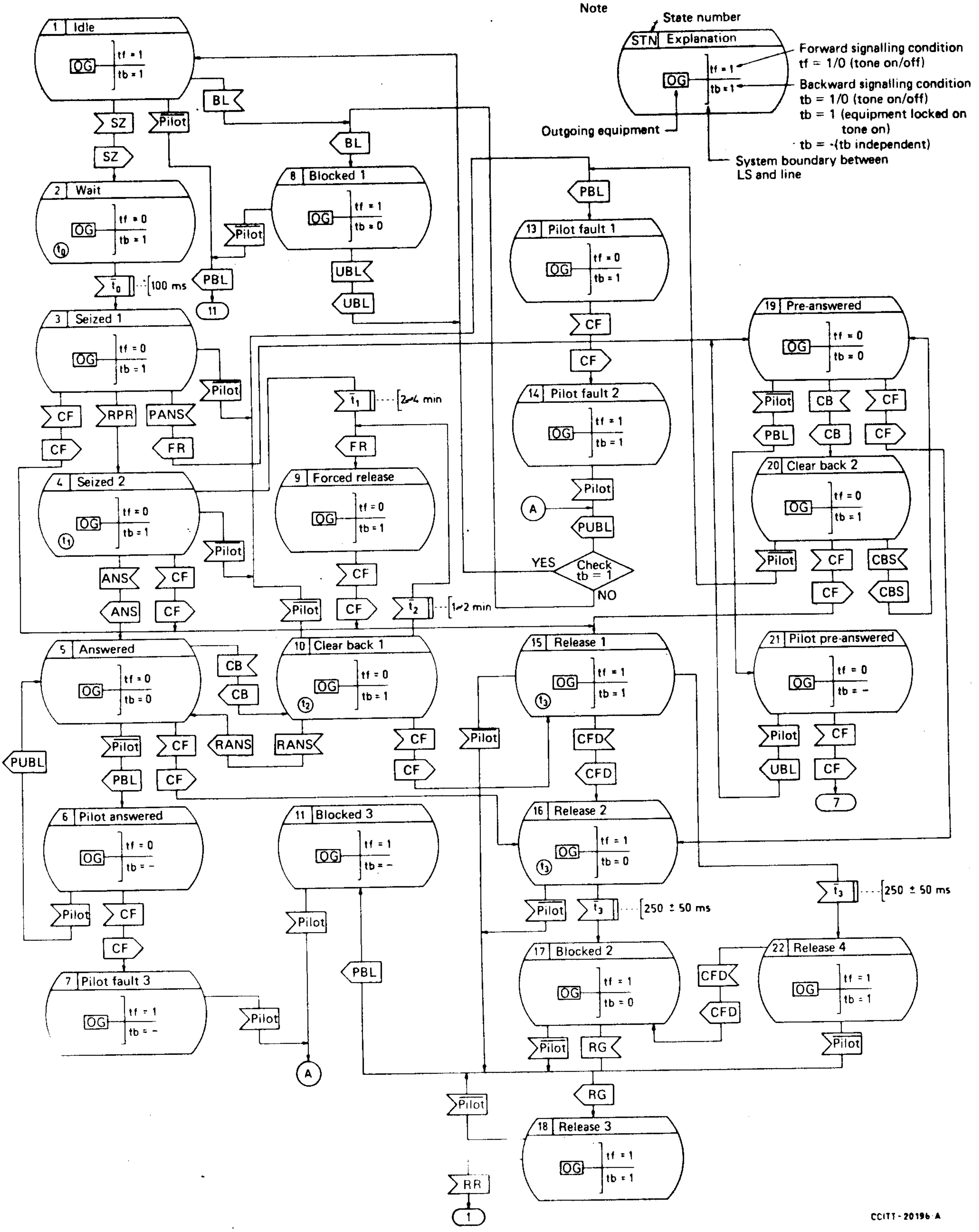
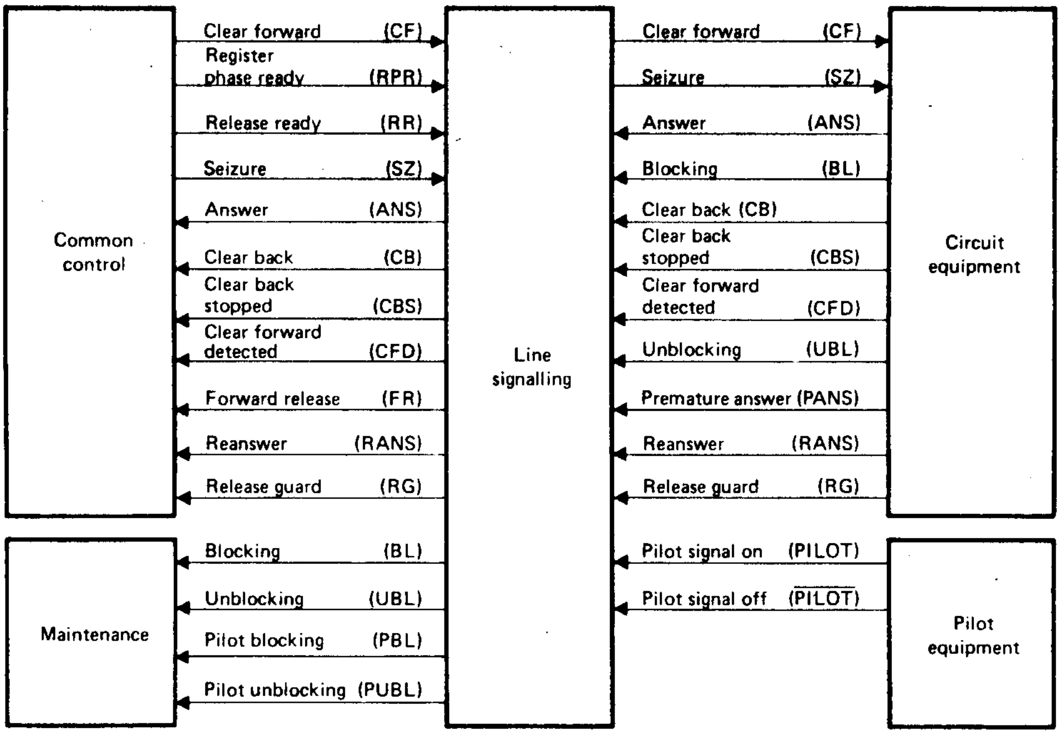


FIGURE 7 – Example 5, an R2 outgoing line-signalling process using state pictures

For notes to this Figure, see end of this Annex.



CCITT-8966-A

FIGURE 8 – Functional block interaction for Examples 4 and 5

Notes to Figure 1 (Example 1)

Note 1. – This example has been designed to show the use of the SDL, and is not an international specification for call handling.

Note 2. – The use of the *blocking* decisions in the SDL diagram accounts for the possibility of there being no free device or free path before device connection or path connection.

Note 3. – The service tones are considered to be of various types S = 1, 2, 3, ... that are not identified separately in Figure 1.

Notes to Figure 2 (Example 2)

Note 1. – This example has been designed to show the use of the SDL, and is not an international specification for local call handling.

Note 2. – The use of the *blocking* decisions in the SDL diagram accounts for the possibility of there being no free device or free path before device connection or path connection.

Note 3. – The service tones are considered to be of various types S = 1, 2, 3, ... that are not identified separately in Figure 2.

Note 4. – Each state is shown only once in full, i.e. with its state number, state title and state picture. All other appearances of the state are shown by a diminished size state symbol containing the state number alone as sufficient identification.

Note 5. – The functional block interactions of this Example 2 are shown in Figure 3.

Note to Figure 4 (Example 3) – This Example has been designed to show the use of the SDL, and is not an international specification.

Notes to Figure 6 (Example 4)

Note 1. – This example has been designed to show the use of the SDL, and is not an international specification.

Note 2. – The input and output symbols have been drawn in such a way that their orientation indicates the direction of the signals between the appropriate FSBs. Common control and maintenance is depicted to the left; circuit equipment and pilot equipment is depicted on the right.

Note 3. – The abbreviation tf = 1 (tf = 0) indicates that the forward tone is on (off). tb refers to the backward tone.

Note 4. – The functional block interactions of this Example 4 are shown in Figure 8.

Notes to Figure 7 (Example 5)

Note 1. – This example has been designed to show the use of the SDL and is not an international specification.

Note 2. – The functional block interactions of this Example 5 are shown in Figure 8.