ANNEX A.3

A.3 *An example of DCME SDL diagrams*

Note that in this annex, the intermediate trunk (IT) is indicated as IC and the bearer channel (BC) as SC. Also note that in this example the use of 24 and 16 kbit/s pre-assigned channels for maintenance are not included.

The diagrams are consistent with the CCITT functional specification and description language (SDL) defined in Recommendation Z.100.

In SDL representations of processes, transitions between states are assumed to occur instantaneously. Practical implementations must account for the time delays.

A.3.1 *Logic diagrams for DCME transmit side*

The logic diagrams in this section of the annex A.3 are supplemental to the description of the DCME transmit side structure given in § A.1. The transmit portion of the assignment procedures have been divided into two blocks:

a) IPS – Input processing and service request generation block

b) SRH – Service request handling block

A.3.1.1 *The IPS block*

The HSC process operates on an IC-by-IC basis. This means that the protocol described below will exist separately for each of the transmit ICs. The following signals are included in the block.

L1: Act, Inact

L2: Data-detect, Voice-detect

L3: Signal-detect

L4: Rxdata

L5: Transpreq, Transprel

L6: Process-reset

L10: Not used

L11: Not used

L12: Voice (Integer), Voiceinact (Integer), Data (Integer), Datainact (Integer), Transp (Integer), Discreq (Integer)

L17: Reset-act, Default-voice, Default-data,
Reset-signaldetect

The following states have been defined within the process included within the block:

– (HSC) hangover control and signal classification process (O,)

Voice-inactive, Voice-active, Voice-wait, Data-active, Data-inactive, Data-wait, Transp, Signalling-active, Signalling-wait, Preassigned, Voice-wait-hold, Signalling-active-hold, Voice-inactive-hold, Voice-active-hold, Signalling-wait-hold, Wait-for-data

The signals arriving from all functional units external to the block are assumed not to require IC addressing in order for the correct process instance to respond to the signal for its IC. There is one exception to this. Address information is required for the signals arriving from the Rx assignment block.

L1 – "Act" and "Inact" signals (for each IC) are received from the transmit activity detector. This unit is assumed to operate according to the following principles:

– The unit has two internal states (for each IC), Activity and No-Activity. At the transition from No-activity to Activity, an Act signal is sent. At the transition from Activity to No-activity, Inact is sent.

– The activity detector is reset to No-activity for an IC by a "Reset-act" signal for that IC.

L2 – "Data-detect" or "Voice-detect" signals are received from the data/speech discriminator for each IC. This unit is assumed to operate according to the following principles:

– All activity apart from data and 2100 Hz tones will be declared as voice.

– 2100 Hz tones and actual data transmissions will be declared as data.

– Inactivity will retain the previous declaration (data or speech) until a new declaration is made due to activity or reset signals.

– The Default-voice signal is used to reset the data/speech discriminator to voice.

– The Default-data signal is used to reset the data/speech discriminator to data.

– The unit has two internal states (data, voice). A signal is sent at the transition between these two states.

L3 – "Signaldetect" is received from the 2400 Hz tone detector for each IC. The unit is assumed to operate according to the following principles:

– The unit is assumed to have two states, Signal-detect and No-signal-detect. A signal will be sent at the transition between No-signal-detect to Signal-detect (see Note).

  Note – Detection of 2400 Hz causes the detector to transit to Signal-detect from No-Signal-detect. When the detector ceases to detect 2400 Hz, it should transit from Signal-detect to No-Signal-detect.

– The unit will be reset to No-Signal-detect by the reception of a Reset-signal-detect signal for that IC.

L4 – Rxdata is received from the receive side assignment handling process and indicates that the dataflag has been set for a certain IC.

L5 – Transpreq and Transprel arrive from the TCH. They will be issued after the TCH receives a request for connection or disconnection of a 64 kbit/s transparent channel from the local ISC or via the receive assignment messages from the remote DCME.

L12 – The HSC Process issues six different signals to the SRH block. These are:

– *Voice (Integer)* – Indicates a transition from voice-inactive to voice-active for an IC.

– *Voiceinact (Integer)* – Indicates a transition from voice-active to voice-inactive for an IC.

– *Data (Integer)* – Indicates a transition from data-inactive to data-active for an IC.

– *Datainact (Integer)* – Indicates a transition from data-active to data-inactive for an IC.

– *Transp (Integer)* – Indicates a transition from the previous state of an IC to a transparent condition.

– *Discreq (Integer)* – Indicates a transition from the transparent condition to voice-inactive for an IC.

L17 – Default-voice, Default-data, Reset-act, and Reset-signal-detect are reset signals.

L6 – The function of this signal is associated with map changes. The following assumptions are here made regarding the IPS block.

A map-changer-handler (MCH) will exist external to the assignment protocol. This process will control the configuration data for the IPS block. At start-up, this process will issue signals making it possible to configure the system correctly. The same will be done at a map change instant. The signal used at the map change instant is:

– *Process-reset* – The L6 signal will cause a termination of the process that receives it.

A.3.1.1.1 *Handling of the IPS processes at initialization*

A number of variables are passed from the map-change-handler to the HSC process when they are created by the map-change-handler. This occurs at system start-up or after a map change. These variables are:

– *ch* – The IC number associated with the process.

– *hot, sh, lh* – sh and lh are the hangover values associated with speech. The hangover is set to "sh" provided that the active signal duration preceding it is shorter than "hot". If this is not the case, "lh" is applied.

– *bhot, bsh, blh* – These are similar variables associated with periods of signalling.

– *dh* – This is the first hangover value nominally associated with data and is considered to be in the order of 14 seconds. It is operator settable.

– *dhs* – This is the second hangover value associated with data and is operator settable.

– *pre* – A Boolean variable which passes on the information whether the IC is pre-assigned or not.

– *RAGPID* – A process identifier variable which is used to address the signals to the SRH block to the correct process instance.

The process uses the following variables:

– *t1, t2* Time variables, stores the current time to be used for hangover handling.

– *d1* – Difference between two times.

– *already* – A Boolean variable used to check that the first data hangover value has only been applied once before starting to use the second hangover value.

– *ti, tia* – Timer variables.

A.3.1.2 *The SRH block*

This block contains four different processes. The following signals are used within this block:

L7: Setcod (Integer, Integer, Boolean)

L8: Assign (Integer, Integer, Call-type, Integer)

L9: Addressmap-for-SCs

L10: Not used

L11: Not used

L12: Voice (Integer), Voiceinact (Integer), Data (Integer), Datainact (Integer), Transp (Integer), Discreq (Integer)

L13: Assign (Integer, Integer, Call-type, Integer), Reinsert (Integer), Remove (Integer), Seizesc (Integer), Integer, (Integer), Release (Integer), Releasesc (Integer), Seizebank (Integer)

L14: SC-bitmap

L15: Mode-map (Integer)

L16: Assign-enc (Integer, Integer, Call-type), Release-enc, Set-pre (Integer, Integer)

L18: Not used

L19: Trigger, Sync-Trigger

L20: Trigger

L21: Process-reset

L22: Process-reset

L23: Trigger

L24: Process-reset

L25: Trigger

L26: Process-reset

L200: Change (Integer)

L201: Sync-Alarm (Integer)

L300: Go-ahead

The processes of the SRH block and their states are the following:

a) *(RAG) request handling and assignment information generation process (0.2)*

No-messages-in-queue, Messages-in-queue, Wait-for-next, Wait-for- sync

b) *(SBC) SC bit map creation process (0.2)*

Wait

c) *(BMI) bit map implementation process (0.2)*

Wait

d) *(ENC) encoder control process (0.)*

Wait-for-signal

The service request handling uses the following symbols:

sat(nr)=bc – The array "sat" uses the transmit IC number to index the SC number to which it is connected. The array is initialized to zero for all the IC numbers used by the DCME at system start-up.

ic(bc)=nr – This array uses the SC number to index the IC number to which it is connected. This array is initialized as zero at system start-up.

typ(bc)=call-type – This array uses the SC number to index the type of connection that is connected to the SC number in question. The values of "call-type" are defined as follows:

"disc", "voiceavail", "voice", "dataavail", "data", "transp", "bank", "preassigned".

The array is initialized as "disc" for all channels at system start-up.

cod(nr)=cd – This array uses the IC number to index the physical coder number to which it is connected. All elements are initialized to zero at system start-up.

The signals used are explained as follows:

L7 – Setcod (Integer, Integer, Boolean) – This signal is issued by the ENC Process and causes the encoder associations to be made. The values delivered are the following:

IC number, Mode (3,4,5 or 8), and a Reset command.

This last variable will be TRUE if a Reset is to be made, FALSE otherwise.

L9 – Addressmap for SCs – Contains the bit addresses to use on the bearer. The signal is a delayed version of the SC-bitmap signal.

L12 – Voice (Integer), Voiceinact (Integer), Data (Integer), Datainact (Integer), Transp (Integer), Discreq (Integer).

L8, L13 –Assign (Integer, Integer, Call-type, Integer) – The variables included have the following meaning:

a) The first integer value gives the SC number to which the IC shall be connected.

b) The second integer value gives the IC number to which the SC shall be connected.

c) The third parameter (Call-type) contains the type of channel that is being assigned.

d) This third integer value contains the actual physical encoder to be used.

This signal is sent to the SBC process and to the environment.

L13 – Reinsert (Integer) – This signal is used to reinsert an SC into the voice list within the SBC process when an implicit disconnect of a data call has occurred.

L13 – Remove (Integer) – Removes an implicitly disconnected overload channel from the SBC overload channel list.

L13 – Seizesc (Integer, Integer, Integer) – Generates a fixed association between an SC number and an encoder number for a pre-assigned channel. The first variable contains the SC number; the second variable contains the encoder number to be used, and the third variable contains the mode (4/5/8).

L13 – Releasesc (Integer) – This signal releases a bit bank connection and is given to the SBC Process. The integer value identifies the SC to be released.

L13 – Seizebank (Integer) – This signal notifies the SBC Process that a certain SC has been seized as a bit bank. It is only used in association with initialization. The integer value indicates the SC that is used as a bit bank.

L13 – Release (Integer) – This signal updates the resource maps within the SBC Process.

L14 – SC bitmap – Contains the bitmap positions for the various channels. This is used to assemble the bearer channel from the output of the different encoders.

L15 – Mode Map (Integer) - This signal is issued by the SBC Process to the ENC Process in order for the correct encoder mode (3/4) to be set for voice connections. The variable contains the mode.

L16 – Set-pre (Integer, Integer) – Seizes a coder for a certain connection. The variables contained imply:

– mode (4/5/8), IC number.

L16 – Assign-enc (Integer, Integer, Call-type) – The variables included have the same meaning as the first three variables defined above for signal L8, L13 Assign (Integer, Integer, Call-type). The signal is sent to ENC process.

L16 – Release-enc – Causes the encoder, identified by the integer value, to release any connection it may have.

L19 – Trigger, Sync-trigger – Trigger signal occurs once every 2 msec period. The Sync-Trigger signal informs the process that the next 2 msec period is the first frame in the DCME multiframe structure. When the Sync-Trigger is present, the Trigger signal is suppressed.

L20, L23, L25 – Trigger – These signals are assumed to occur once every 2 msec period.

L21, L22, L24, L26 – Process-reset – This signal is generated by the map-change-handler in association with a map change and causes a termination of the process that receives it.

L200 – Change (Integer) – This signal arrives from the USM module and contains an IC number that is to be loaded into the Priority 0 Queue which is served by the RAG every nth frame.

L201 – Sync-Alarm (Integer) – This signal is sent if there seems to be a logical problem with the multiframe synchronization within the RAG. The Integer variable identifies the pool number for which the alarm has been raised.

L300 – Go-ahead (Pld, Pld, pre-assigned_list, pre-assigned_list, ic access_list, ic_access_list) – This signal is sent from the MCHA2 process to the MCHA1 process at traffic start up or at traffic reconfiguration. The signal contains information regarding the usage of the ILS and applied hangover times and hangover thresholds.

A.3.1.2.1 *The RAG Process*

The RAG Process is created by the Map-change-handler at system start-up or after a map change. Depending on the possible use of one or two pools, one or two instances of the process will be created. The Map-change-handler delivers a number of parameters to the process. The functions of these parameters are explained below:

- *b* – This integer variable contains the total number of 4-bit samples that are contained within the pool.
- *no* – This integer value contains the total number of SCs in the normal range in the pool that are not pre-assigned.
- *pre(i)* – This array contains the IC numbers of pre-assigned channels.
- *cdlist* – This list contains the physical encoder-numbers that the process may choose from when establishing a connection. The encoders that are to be used for pre-assigned connections are not included (see Note).

  Note – A variable instance of type "list" contains a list of integer numbers which can be accessed separately.
- *presc(i)* – This array contains the SCs to which the pre-assigned ICs should be connected. In the case where the pre-assigned IC is 64 kbit/s, only the even numbered SC is contained within the array.
- *premode(i)* – This array contains the mode (4/5/8) associated with each pre-assigned IC.
- *sclist* – This list contains the SCs which can be used by the process. The pre-assigned SC numbers are not included.
- *ptot* – This integer contains the total number of pre-assigned ICs that should be handled by the process.
- *sel(i)* – This array contains the encoder numbers that are to be used for the pre-assigned ICs.
- *bitbank(i)* – This array, with a maximum of 12 entries, contains the SC numbers to be used for bit banks. The SC numbers are maintained in numerically ascending order. At start-up, the array will contain the SC numbers required to handle pre-assigned 40 kbit/s channels.
- *btot* – This integer value contains the total number of bit banks required at any given moment to handle the number of data calls that are connected. At start-up, the variable will contain the number of bit banks required to handle pre-assigned 40 kbit/s channels.
- *sq* – This Boolean variable contains the value TRUE if the optional USM information is to be handled by the RAG Process.
- *n* – This integer value contains the periodicity of the optional USM information handling as the number of frames.
- *ENCPID(i)* – This array uses encoder numbers as indexes and identifies the process identifier applicable to the encoder number process instance.
- *pnr* – This integer variable identifies the pool number of the RAG process instance.
- *s* – This integer variable defines the lowest number of bits/sample permitted. Its value is either 3, for 3-bit encoding, or 2, for 2-bit encoding.

The process uses a number of different procedures and variables. The procedures are only included within the diagrams as procedure calls and the variables as names. Their meaning is described below in order of appearance:

- *Rm* – This variable is TRUE if an SC number is to be removed from an SBC resource list, otherwise FALSE.
- *Prev* – This variable is TRUE if a previous connection exists for another type of call for that IC, otherwise FALSE.
- *Reins* – This variable is TRUE if a SC number is to be reinserted in an SBC resource list, otherwise FALSE.
- *Rethere* – This variable is TRUE if a bit bank has to be created as a function of changing an IC already connected to an SC to "data", FALSE otherwise.

– *Return1* – This variable is TRUE if a re-assignment due to a connection of a transparent call is in progress, otherwise FALSE.

– *Return2* – This variable is TRUE if a re-assignment due to a connection of a transparent call is in progress, otherwise FALSE.

– *i* – Counter.

– *again* – This variable is TRUE if a refreshment message for the current SC should not be generated, otherwise FALSE.

– *r* – Counter.

– *r1* – Counter.

– *nr* – This integer variable stores the IC number associated with an incoming request.

– *ovlr* – This variable is TRUE if an overload channel is to be refreshed, otherwise FALSE.

– *f* – A local counter used to keep track of the frame number within the multiframe number, set to 0 after the reception of a sync trigger signal.

– *Store"X" (nr)* – This procedure stores the variable "nr" at the bottom of the priority queue marked "X".

– *req in queue (nr)* – This array is indexed by the IC numbers and stores the value 0 for a given index if there are no requests for that IC in any of the queues 2 to 5. It stores the value 1 if there is a request for that IC in any of the queues 2 to 5.

– *pr"X"count* – A variable which stores the number of requests that exist in priority queue "X".

– *req in discqueue(nr)* – This array is indexed by the IC numbers and stores the value 0 for a given index if there are no requests for that IC in queue 1. It stores the value 1 if there is a request for that IC in queue 1.

– *Remove from RAG queue (nr, more)* – This procedure removes any request for the IC "nr" from any of the queues 2 to 5. The pr"x"count variable in the queue is updated accordingly. The procedure also stores the value TRUE in the variable More if there is at least one request in any of the five queues after the removal has been performed, otherwise More is set to FALSE.

– *Additional-messages (more)* - This procedure checks if there are any messages remaining in queues 1 through 5. If this is the case, the variable "more" is set to TRUE, otherwise FALSE.

– *Read"X" (nr)* – This procedure reads the IC at the top of the "X" queue and delivers this value in the variable "nr".

– *Pop"X" (pr"X"count, more)* – This procedure removes the IC value at the top of the queue and pops the queue by one step. It updates the pr"X" count variable for that queue and delivers the value TRUE in the variable More if there is at least one request in any of the queues 1 through 5 after this operation is performed, otherwise More is set to FALSE.

– *Count data (difference)* – This procedure checks the number of 40 kbit/s pre-assigned channels and 40 kbit/s data channels that exist and compares it to the number of bit banks that are in use. If it is possible to delete a bit bank, the variable "difference" is given as TRUE, otherwise FALSE.

– *Count (nt, nd, nb, nv)* – This procedure checks the "typ" array and delivers the number of transparent calls currently being handled in the variable "nt", the number of data calls currently being handled in the variable "nd" and the number of bit banks in use in the variable "nb" and the number of voice calls currently being handled in the variable "nv".

– *d* – This variable is used to store the mean number of bits per voice call that the handling of an additional call would result in, or the total number of bits in the frame left for use after handling a request.

– *Search transp (bc, nr, cd, nrv1, nrv2, bcv1, bcv2, nrv3, nrv4, bcv3, bcv4, success)* – This procedure searches the array "typ" for a possible place to connect a transparent call. There are ten possibilities which are searched for in descending order of priority. It cannot be guaranteed that the search will find at least one of these possibilities even when the check of available bits has been successfully passed. If the search fails the variable success is given as FALSE, otherwise TRUE. The procedure delivers the results of the search in a number of parameters that indicate if action is to be taken. The variables have the following meaning:

  1) *bc* – The even SC number which the transparent call is to be assigned to.

  2) *bc + 1* – The SC number just above that will also be used by the transparent call (a derived variable).

  3) *nr* – The IC number containing the transparent call.

  4) *cd* – The encoder number picked by the procedure from the pool of available encoders. It should be noted that there may be a need to pick-up the encoder from one of the selected channels if this is declared "voiceavail" or "dataavail". Specifically, if "cod(nr)" is not equal to 0, cd = cod(nr).

  5) *nrv1* – The IC already connected to "bc".

  6) *nrv2* – The IC already connected to "bc+1".

  7) *bcv1* – The SC that "nrv1" is to be re-assigned to.

  8) *bcv2* – The SC that "nrv2" is to be re-assigned to.

  9) *nrv3* – The IC already connected to "bcv1".

  10) *nrv4* – The IC already connected to "bcv2".

  11) *bcv3* – An overload SC that "nrv3" is to be re-assigned to.

  12) *bcv4* – An overload SC that "nrv4" is to be re-assigned to.

  13) *success* – Result of the search (TRUE or FALSE).

– *k* – The SC that the IC was previously connected to before changing to a transparent/data/voice call.

– *tk* – A temporary variable used to store the value of "k".

– *tnr* – A temporary variable used to store the value of "nr".

– *Check for additional bit bank (new)* – This procedure checks if a bit bank is required if one additional data call was to be handled by the bearer. If this is the case, the variable "new" is set to 1, otherwise to 0. It should be noted that if at least one SC declared "data-avail" exists, "new" is always equal to 0.

– *Make room in bitbankarray (nw, bc, bitbank)* – This procedure handles the bitbankarray in order to make it possible to inset "bc" in its correct position, thus keeping the SC numbers used for bit banks in ascending numerical order. The entry which should have the value "bc" associated with it is given in the variable "nw".

– *Search data (bc, nr, cd, nrv, bcv, data success)* – This procedure searches the "typ" array for possible places to connect a data call. There are four possibilities which are searched for in descending order of priority. It also checks the need for an overload channel re-assignment to handle a data connection request. If an overload channel re-assignment is not needed, the variable "data success" is given the value TRUE, otherwise it is given the value FALSE. The procedure will deliver its results in variables having the following meaning:

  1) *bc* – The SC number that the data call is to be connected to.

  2) *nr* – The IC number that is to be connected as a data call.

  3) *cd* – The encoder number picked by the procedure. It should be noted that there may be a need to pick an encoder freed by the use of a channel declared "Voiceavail" or "Dataavail". Specifically, if cod(nr) is not equal to 0, cd = cod(nr).

4) *nrv* – The IC number previously connected to "bc".

5) *bcv* – The SC number that "nrv" is to be re-assigned to. This number is always an overload SC number.

6) *data success* – Result of the check for an overload channel re-assignment requirement (TRUE not needed, FALSE needed).

– *Search voice (bc, nr, cd, nrv)* – This procedure searches the "typ" array for possible places to connect a voice call. There are three possibilities which are searched for in descending order of priority.

The results of the procedure are delivered in variables having the following meaning:

1) *bc* – The SC number that the voice call is to be assigned to.

2) *nr* – The IC number containing the voice call.

3) *cd* – The encoder number picked by the procedure. It should be noted that there may be a need to pick an encoder freed as a consequence of handling that request. Specifically, if cod(nr) is not equal to 0, cd = cod(nr).

4) *nrv* – The IC number previously connected to "bc".

– *SBCPID* – The process identifier variable which is used to address the signals to the correct SBC process.

– *Check_overload_reassignment_when_data (data_success)* – This procedure checks the need for an overload re-assignment to handle a data connection request. If an overload channel re-assignment is not needed, the variable data_success is given the value TRUE, otherwise it is given the value FALSE.

– *Check_overload_reassignment_when_transp (nr,success)* – This procedure checks the need for an overload re-assignment to handle a transparent connection request. If an overload channel re-assignment is not needed, the variable data_success is given the value TRUE, otherwise FALSE. It is noted that if "nr" is "voice" or "voice_avail", two voice channels may be taken from an overload channel creating voice channel pool, and if "nr" is "data" or "data_avail", only one channel may be taken from the pool.

A.3.1.2.2 *The SBC Process*

This process is created by the RAG Process and receives at its creation three parameters as input. These are:

– *sclist* – The current list of SC numbers that is used by this pool. The pre-assigned SCs are not included within this list.

– *bt* – The total number of 4-bit samples within the pool. This number determines the maximum number of SC numbers that can exist, namely:

bt + Integer [bt/3]

This is required for handling the various maps and arrays within the process.

– *ENCPID (i)* – This array uses encoder numbers as indexes and identifies the process identifier applicable to that encoder number process instance.

The following internal resource maps exist within the process:

Voicelist, Overloadlist, Datalist, Transplist, Preassign40list, Preassign64list, Preassign32list, Banklist, Coder (Integer).

The functions and the internal rules governing their use are contained within the specification.

The following parameters and procedure calls are made use of within the process:

– *Generate maps* – This procedure takes the input parameters and generates the various lists and arrays in accordance with the rules regarding initialization of these lists and arrays.

– *Change coder array (cod)* – This procedure goes through the coder array until it finds a SC indexing the coder number "cod". This SC number has its coder number set to zero.

– *Include in voicelist and extract (b)* – This procedure takes the SC number "b" and includes it in the appropriate place in the voicelist and removes it out of any other list that it might exist in. If it finds the SC number in question within the transplist, this should be extracted after which the SC number "b+1" should also be inserted into the voicelist. If "b" is already included in the voicelist, no action shall be taken.

– *Delete overload (b)* – This procedure removes SC number "b" from the overloadlist. If "b" is not included in the overloadlist, no action shall be taken.

– *Generate address mode (bit map)* – This procedure generates the bitpositions and the various modes associated with the encoders. It uses the "ic" variable defined by the received Assign signal as a pointer. This is done in accordance with the rules outlined within the specification. The output is put into the signals mode-map and SC-bitmap.

– *Preassign40 (b,cod)* – This procedure will include SC number "b" in the preassign40list and set the coder array entry for "b" to "cod".

– *Preassign64 (b,cod)* – This procedure will include SC number "b" in the preassign64list and set the coder array entry for "b" and "b+1" to "cod".

– *Preassign32 (b,cd)* – This procedure is used to insert the number "b" into preassign32list upon reception of the signal "seizesc" containing a mode set to 4. It also sets Coder(b) to "cod".

– *Include in banklist and extract (b)* – This procedure will include SC number "b" in the banklist and extract SC number "b" from any other list it might exist in. If already included within the banklist, no action shall be taken.

– *Included in banklist (b,included)* – This procedure checks if SC number "b" is included within banklist or not. If this is the case, the variable "Included" is set to TRUE, otherwise to FALSE.

– *Included in datalist (b,included)* – This procedure checks if SC number "b" is included in the datalist. If this is the case, the variable "Included" is set to TRUE, otherwise to FALSE.

– *Included in transplit (b,included)* – This procedure checks if SC number "b" is included in transplist. If this is the case, the variable "Included" is set to TRUE, otherwise to FALSE.

– *Included in datalist and extract (b)* – This procedure includes the SC number "b" in the datalist and extracts it from any other list it might exist in. If "b" already included in the datalist, no action shall be taken.

– *Update coder association (b,cod)* – This procedure sets the coder array entry for "b" to "cod". It also checks to see if "cod" is associated with any other SC numbers in the array. Should this be the case, the entries for those SC are set to zero.

– *Update coder association for transp (b,cod)* – This procedure sets the entries for "b" and "b+1" to "cod" in the coder array. It also checks the array to see if "cod" is given as an entry for other SC numbers. If this is the case, these entries are set to zero.

– *Include in transplist and extract (b)* – This procedure will extract SC numbers "b" and "b+1" from any list they might exist in and insert SC number "b" into the transplist. If "b" is already included in the transplist, no action shall be taken.

– *Included in voicelist (b,included)* – This procedure checks if SC number "b" is included within voicelist. If this is the case, the variable "Included" is set to TRUE, otherwise FALSE.

– *Included in overloadlist (b,included)* – This procedure checks to see if SC number "b" is included within the overloadlist. If this is the case, the variable "Included" is set to TRUE, otherwise to FALSE.

– *Include in overloadlist (b)* – This procedure includes SC number "b" in the overloadlist. If "b" is already included in the overloadlist, no action shall be taken.

– *i* – counter.

– *md* – This variable indicates the mode (4/5/8) received in the signal siezesc.

– *ic* – The IC number received in the signal Assign.

– *typ* – The Call-type received in the signal Assign.

– *mode(i)* – This array contains the mode of each connection (2/3/4/5/8). The list is updated every DCME frame by the procedure Generate-address-mode.

– *BMIPID* – A process identifier variable which is used to address signals to the correct BMI process instance.

A.3.1.2.3 *ENC Process*

There are as many instances of this process as there are encoders. The processes are created by the Map-change-handler at system start-up. The following variables and procedures are used within the process.

– *change* – This variable stores the value TRUE if data for the encoder has been received since the last Trigger signal. Otherwise FALSE is stored.

– *ic* – The IC number that the unit currently is connected to. Initialized as zero.

– *mode* – The mode of the encoder (2/3/4/5/8 bits/sample). Initialized as zero.

– *reset-coder* – This variable stores the value TRUE if a reset of the coder is to be made, otherwise it stores the value FALSE.

– *fic* – This variable stores a future IC connection for the encoder.

– *fmode* – This variable stores the future mode for the encoder.

– *cic* – This variable stores a current IC connection for the encoder.

– *cmode* – This variable stores a current mode for the encoder.

– *amd* – This variable stores the received mode from the SBC process.

– *cd* – This variable stores the encoder number addressed by the signals Release-enc and Assign-enc.

– *Store(fic, fmode)* – This procedure stores the values of the parameters included at the bottom of a queue. At initialization, this queue shall store 0 for all variables in all positions.

– *Retrieve (cic,cmode)* – This procedure takes the values stored three DCME frames ago from the top of the queue and delivers the results in the variables "cic", and "cmode". The values at the lower queue positions are moved up one place.

– *b* – SC number contained in an Assign-enc signal.

– *nr* – IC number contained in an Assign-enc signal.

– *typ* – The type of connection contained in an Assign-enc signal.

– *md* – The mode contained in a Set-pre signal.

  It should be noted that a setcod signal containing ic=0, mode=0 and Reset-coder=False should not have any impact on the encoder being addressed.

A.3.1.2.4 *The BMI Process*

This process is created at system start-up and only delays the signal by three DCME frames. It contains the following internal procedure calls:

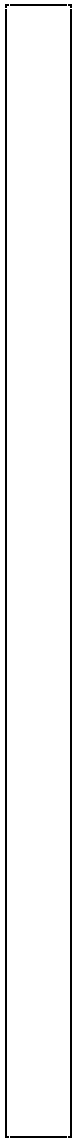
– *Store (bit map)* – This procedure takes the information contained within the SC-bitmap signal and places this information at the bottom of a queue. At initialization, the queue shall contain an all zeros address in all its queue positions.

– *Retrieve (bit map)* – This procedure extracts the information queued three DCME frames ago and delivers this to the signal Addressmap-for SCs. The values at the lower positions in the queue are moved up one place.

– It should be noted that an all zeros address in the signal addressmap-for-SCs should not cause any connections to be made between encoder outputs and the bearer.

FIGURE (sans numéro)

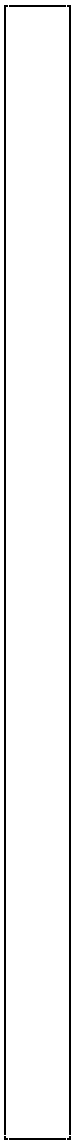
Manuscrit: **COM XV-R 39 -E/F/S** Page **175**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE (sans numéro)

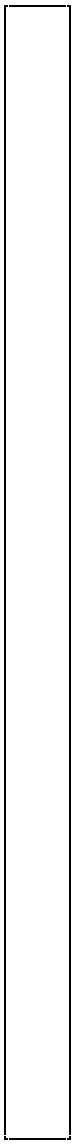
Manuscrit: **COM XV-R 39 -E/F/S** Page **176**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE (sans numéro)

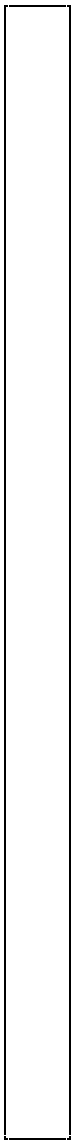
Manuscrit: **COM XV-R 39 -E/F/S** Page **177**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE (sans numéro)

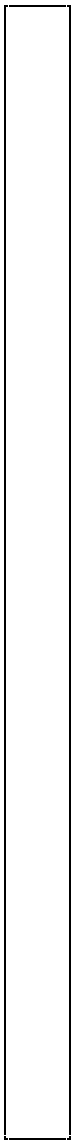
Manuscrit: **COM XV-R 39 -E/F/S** Page **178**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE (sans numéro)

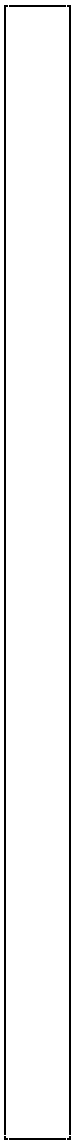
Manuscrit: **COM XV-R 39 -E/F/S** Page **179**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE (sans numéro)

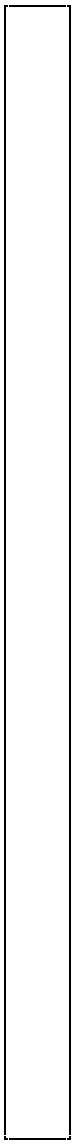
Manuscrit: **COM XV-R 39 -E/F/S** Page **180**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE (sans numéro)

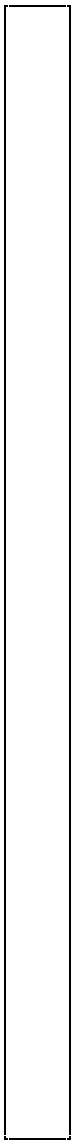
Manuscrit: **COM XV-R 39 -E/F/S** Page **181**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE (sans numéro)

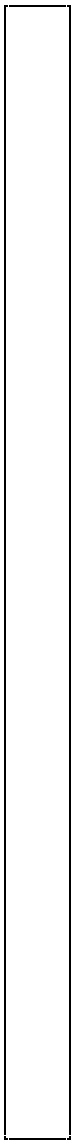
Manuscrit: **COM XV-R 39 -E/F/S** Page **182**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE (sans numéro)

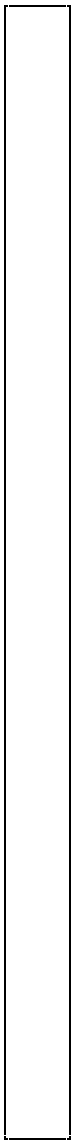
Manuscrit: **COM XV-R 39 -E/F/S** Page **183**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1503400-90)**

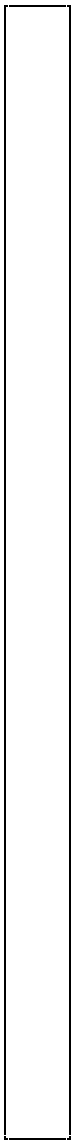
Manuscrit: **COM XV-R 39 -E/F/S** Page **184**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(1503410-90)**

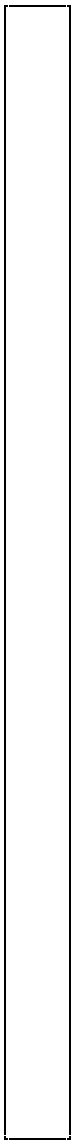
Manuscrit: **COM XV-R 39 -E/F/S** Page **185**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1503420-90)**

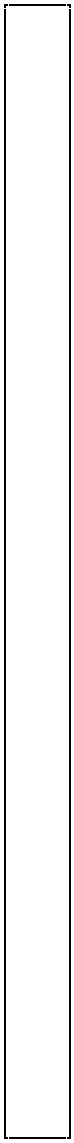
Manuscrit: **COM XV-R 39 -E/F/S** Page **186**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1503430-90)**

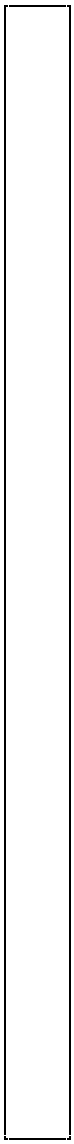
Manuscrit: **COM XV-R 39 -E/F/S** Page **187**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1503440-90)**

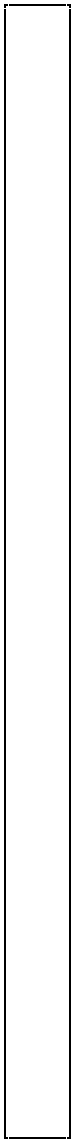
Manuscrit: **COM XV-R 39 -E/F/S** Page **188**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1503450-90)**

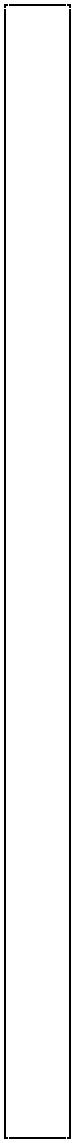
Manuscrit: **COM XV-R 39 -E/F/S** Page **189**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1503460-90)**

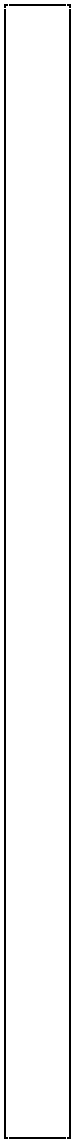
Manuscrit: **COM XV-R 39 -E/F/S** Page **190**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1503470-90)**

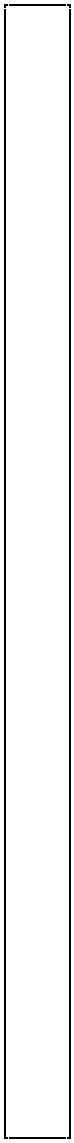
Manuscrit: **COM XV-R 39 -E/F/S** Page **191**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1503480-90)**

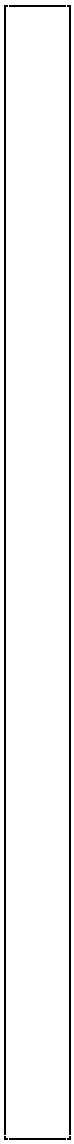
Manuscrit: **COM XV-R 39 -E/F/S** Page **192**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1503490-90)**

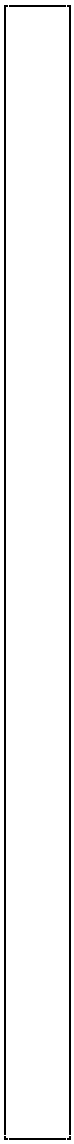
Manuscrit: **COM XV-R 39 -E/F/S** Page **193**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1503500-90)**

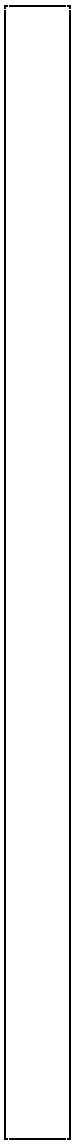
Manuscrit: **COM XV-R 39 -E/F/S** Page **194**



MONTAGE: Monter les figures dans l´anglais uniquement.

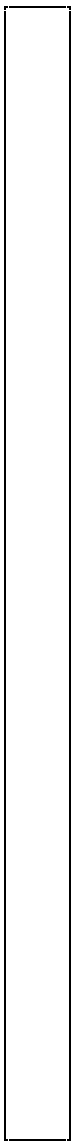
FIGURE **(T1503510-90)**

Manuscrit: **COM XV-R 39 -E/F/S** Page **195**



MONTAGE: Monter les figures dans l´anglais uniquement.

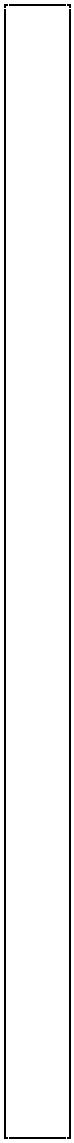
FIGURE **(T1503520-90)**



MONTAGE: Monter les figures dans l´anglais uniquement.

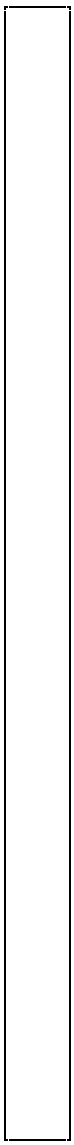
FIGURE **(T1503530-90)**

Manuscrit: **COM XV-R 39 -E/F/S** Page **197**



MONTAGE: Monter les figures dans l´anglais uniquement.

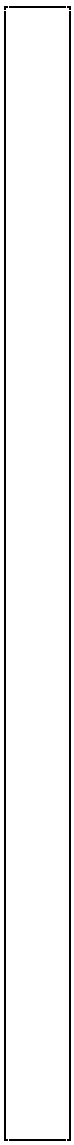
FIGURE **(T1503540-90)**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1503550-90)**

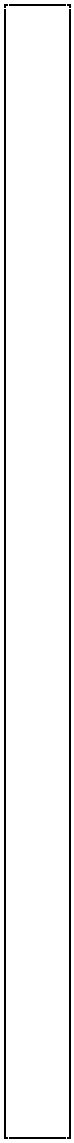
Manuscrit: **COM XV-R 39 -E/F/S** Page **199**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(sans numero)**

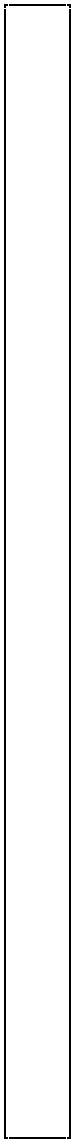
Manuscrit: **COM XV-R 39 -E/F/S** Page **200**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(sans numero)**

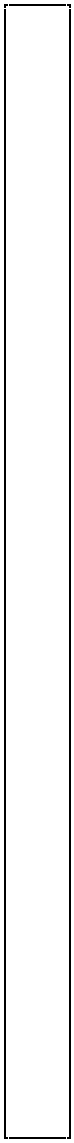
Manuscrit: **COM XV-R 39 -E/F/S** Page **201**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1503560-90)**

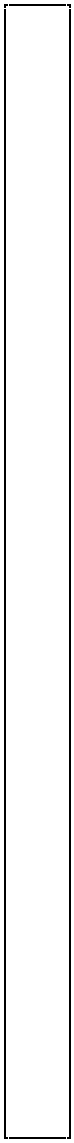
Manuscrit: **COM XV-R 39 -E/F/S** Page **202**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1503570-90)**

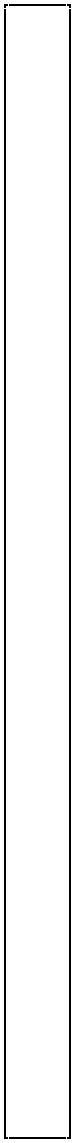
Manuscrit: **COM XV-R 39 -E/F/S** Page **203**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1503580-90)**

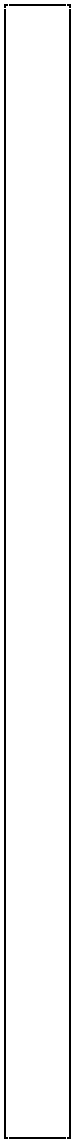
Manuscrit: **COM XV-R 39 -E/F/S** Page **204**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1503590-90)**

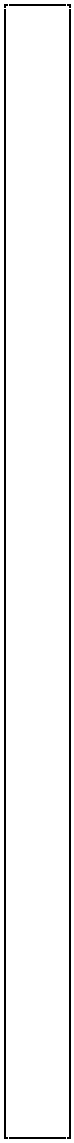
Manuscrit: **COM XV-R 39 -E/F/S** Page **205**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1503600-90)**

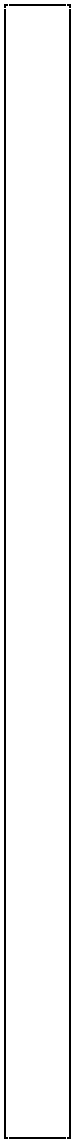
Manuscrit: **COM XV-R 39 -E/F/S** Page **206**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1503610-90)**

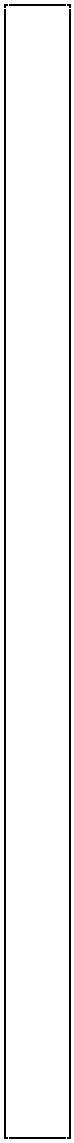
Manuscrit: **COM XV-R 39 -E/F/S** Page **207**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1503620-90)**

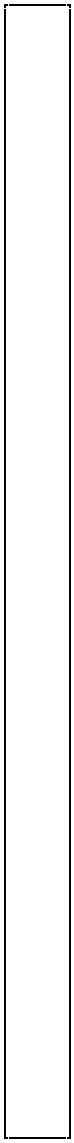
Manuscrit: **COM XV-R 39 -E/F/S** Page **208**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1503630-90)**

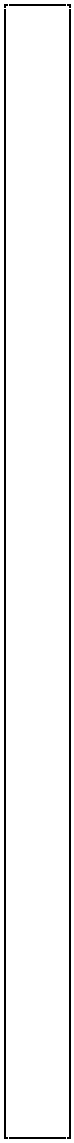
Manuscrit: **COM XV-R 39 -E/F/S** Page **209**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1503640-90)**

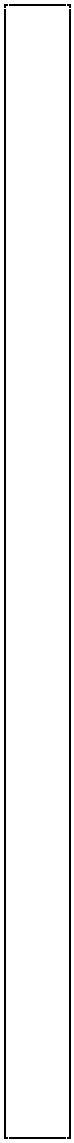
Manuscrit: **COM XV-R 39 -E/F/S** Page **210**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1503650-90)**

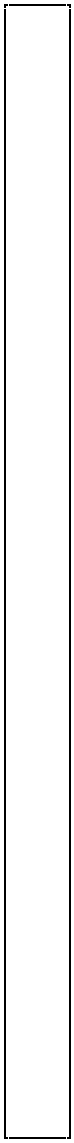
Manuscrit: **COM XV-R 39 -E/F/S** Page **211**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1503660-90)**

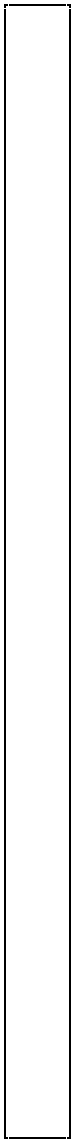
Manuscrit: **COM XV-R 39 -E/F/S** Page **212**



MONTAGE: Monter les figures dans l´anglais uniquement.

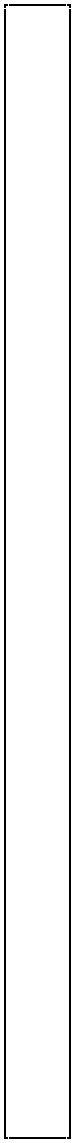
FIGURE **(T1503670-90)**

Manuscrit: **COM XV-R 39 -E/F/S** Page **213**



MONTAGE: Monter les figures dans l´anglais uniquement.

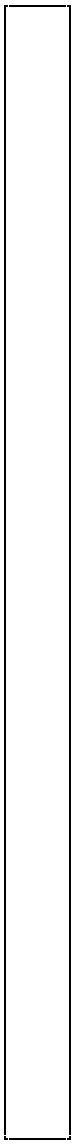
FIGURE **(T1503680-90)**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1503690-90)**

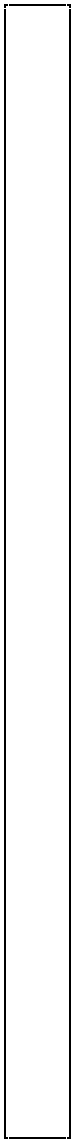
Manuscrit: **COM XV-R 39 -E/F/S** Page **215**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1503700-90)**

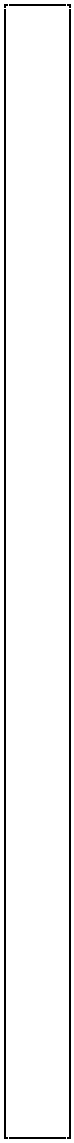
Manuscrit: **COM XV-R 39 -E/F/S** Page **216**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1503710-90)**

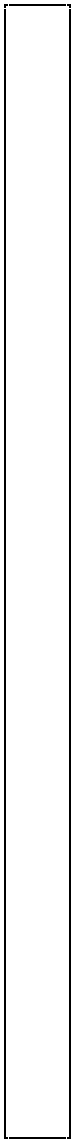
Manuscrit: **COM XV-R 39 -E/F/S** Page **217**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1503720-90)**

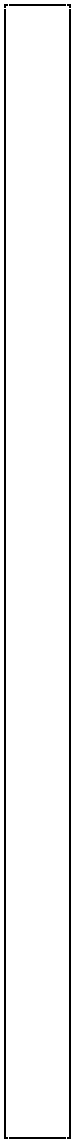
Manuscrit: **COM XV-R 39 -E/F/S** Page **218**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1503730-90)**

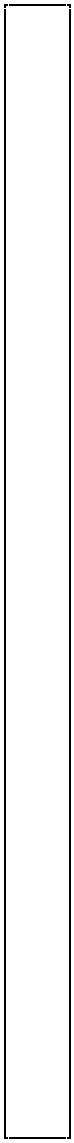
Manuscrit: **COM XV-R 39 -E/F/S** Page **219**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1503740-90)**

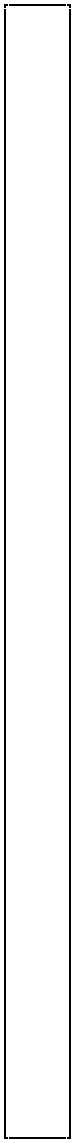
Manuscrit: **COM XV-R 39 -E/F/S** Page **220**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1503750-90)**

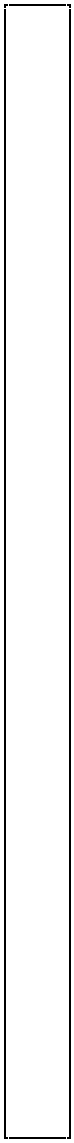
Manuscrit: **COM XV-R 39 -E/F/S** Page **221**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1503760-90)**

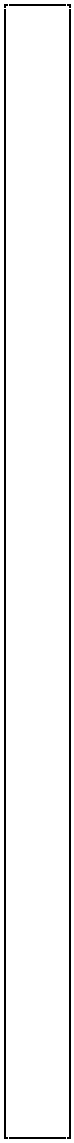
Manuscrit: **COM XV-R 39 -E/F/S** Page **222**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1503770-90)**

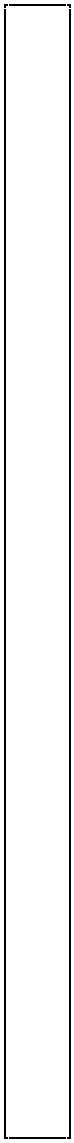
Manuscrit: **COM XV-R 39 -E/F/S** Page **223**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1503780-90)**

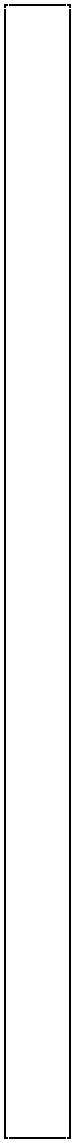
Manuscrit: **COM XV-R 39 -E/F/S** Page **224**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1503790-90)**

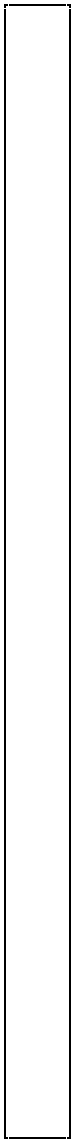
Manuscrit: **COM XV-R 39 -E/F/S** Page **225**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1503800-90)**

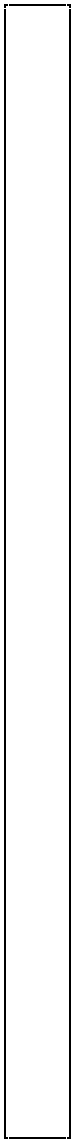
Manuscrit: **COM XV-R 39 -E/F/S** Page **226**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1503810-90)**

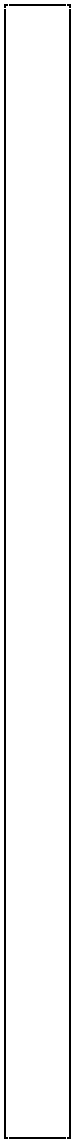
Manuscrit: **COM XV-R 39 -E/F/S** Page **227**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1503820-90)**

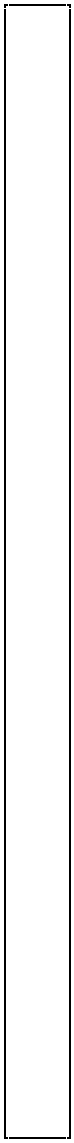
Manuscrit: **COM XV-R 39 -E/F/S** Page **228**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1503831-90)**

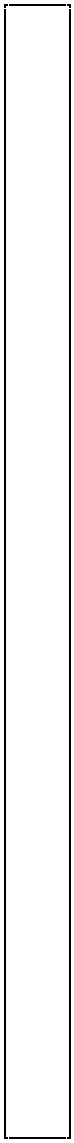
Manuscrit: **COM XV-R 39 -E/F/S** Page **229**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1503841-90)**

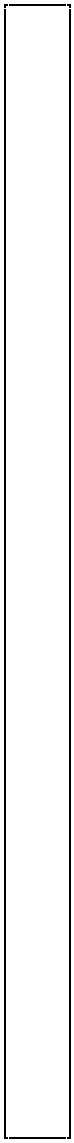
Manuscrit: **COM XV-R 39 -E/F/S** Page **230**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1503850-90)**

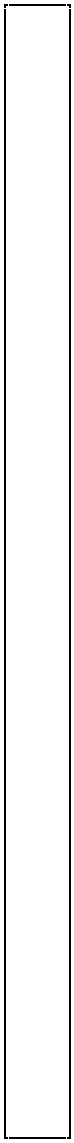
Manuscrit: **COM XV-R 39 -E/F/S** Page **231**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1503860-90)**

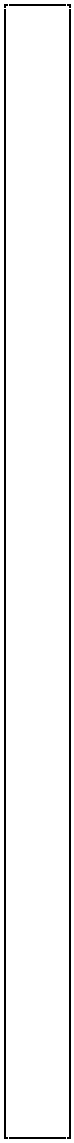
Manuscrit: **COM XV-R 39 -E/F/S** Page **232**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1503870-90)**

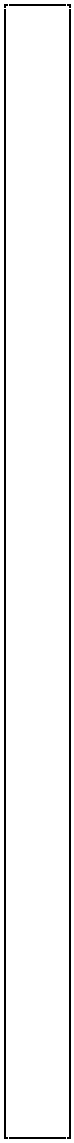
Manuscrit: **COM XV-R 39 -E/F/S** Page **233**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1503880-90)**

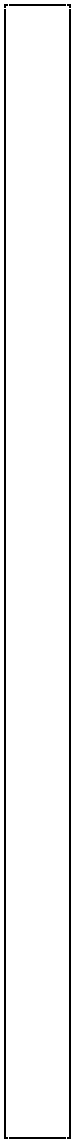
Manuscrit: **COM XV-R 39 -E/F/S** Page **234**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1503891-90)**

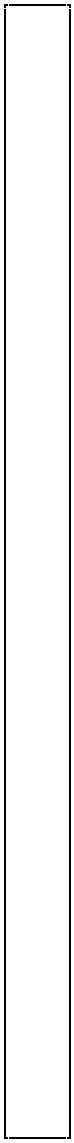
Manuscrit: **COM XV-R 39 -E/F/S** Page **235**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1503901-90)**

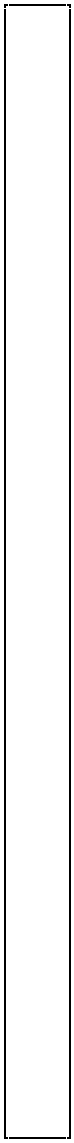
Manuscrit: **COM XV-R 39 -E/F/S** Page **236**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1503910-90)**

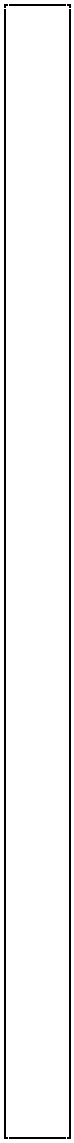
Manuscrit: **COM XV-R 39 -E/F/S** Page **237**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1503921-90)**

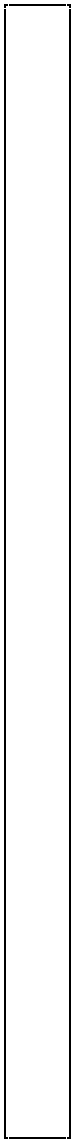
Manuscrit: **COM XV-R 39 -E/F/S** Page **238**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1503930-90)**

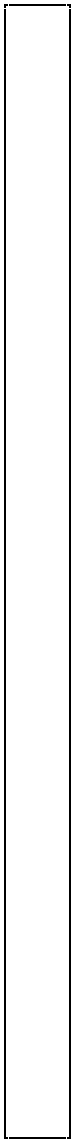
Manuscrit: **COM XV-R 39 -E/F/S** Page **239**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1503940-90)**

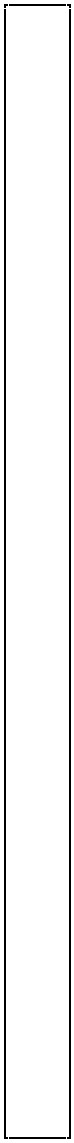
Manuscrit: **COM XV-R 39 -E/F/S** Page **240**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1503950-90)**

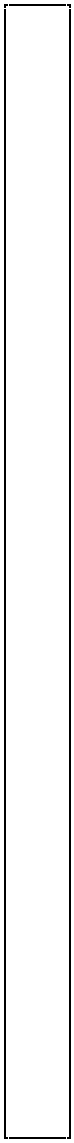
Manuscrit: **COM XV-R 39 -E/F/S** Page **241**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1503960-90)**

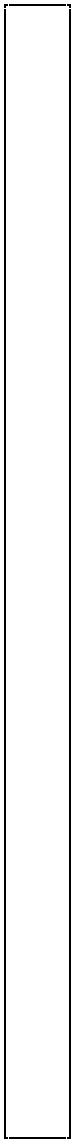
Manuscrit: **COM XV-R 39 -E/F/S** Page **242**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1503970-90)**

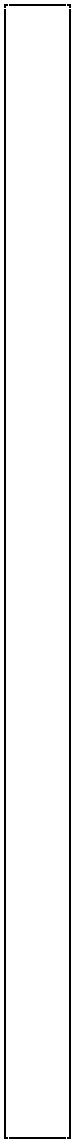
Manuscrit: **COM XV-R 39 -E/F/S** Page **243**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1503980-90)**

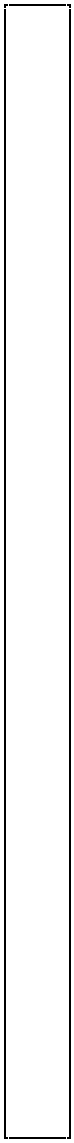
Manuscrit: **COM XV-R 39 -E/F/S** Page **244**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1503990-90)**

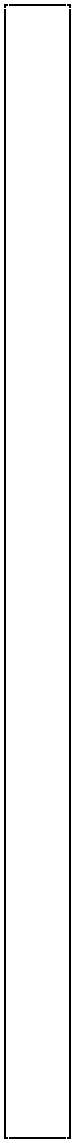
Manuscrit: **COM XV-R 39 -E/F/S** Page **245**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504000-90)**

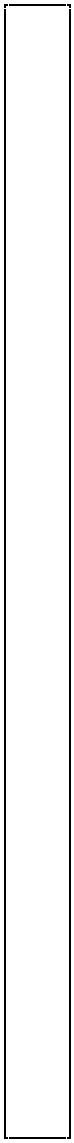
Manuscrit: **COM XV-R 39 -E/F/S** Page **246**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504010-90)**

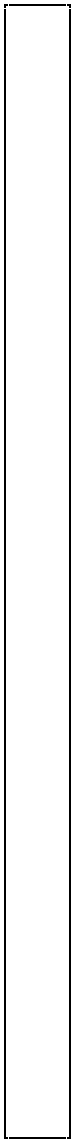
Manuscrit: **COM XV-R 39 -E/F/S** Page **247**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504020-90)**

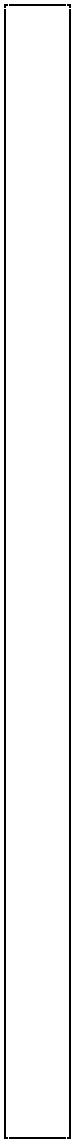
Manuscrit: **COM XV-R 39 -E/F/S** Page **248**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504050-90)**

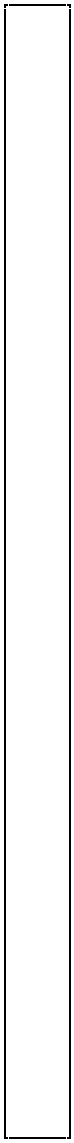
Manuscrit: **COM XV-R 39 -E/F/S** Page **249**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504040-90)**

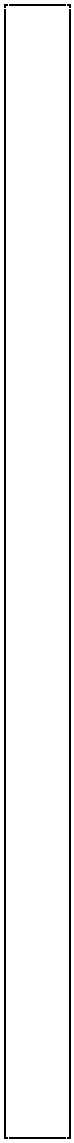
Manuscrit: **COM XV-R 39 -E/F/S** Page **250**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504050-90)**

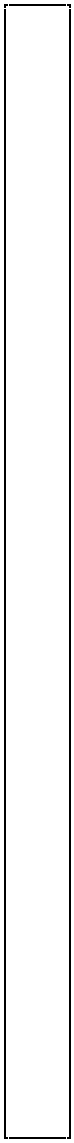
Manuscrit: **COM XV-R 39 -E/F/S** Page **251**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504060-90)**

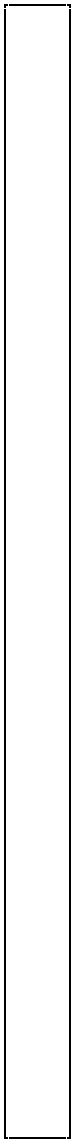
Manuscrit: **COM XV-R 39 -E/F/S** Page **252**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504070-90)**

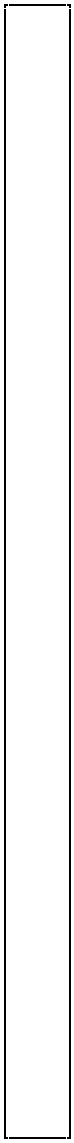
Manuscrit: **COM XV-R 39 -E/F/S** Page **253**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504080-90)**

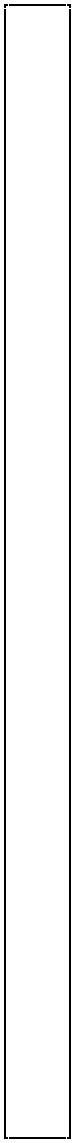
Manuscrit: **COM XV-R 39 -E/F/S** Page **254**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504090-90)**

Manuscrit: **COM XV-R 39 -E/F/S** Page **255**



MONTAGE: Monter les figures dans l´anglais uniquement.

A.3.2 *Logic diagrams for DCME receive side*

The logic diagrams in this section of Annex A.3 are supplemental to the description of the DCME receive side structure given in Annex A.2. The receive side handling procedures are contained within one block:

RCP: Receive channel processing block.

A.3.2.1 *The RCP block*

The RCP block contains three different processes and the following signals:

L4: Rxdata (Integer);

L51: Assign (Integer, Integer, Call-type);

L52: Rxtranspreq (Integer), Rxtransprel (Integer);

L53: SC-bitmap (Bit-mode-matrix);

L54: Seize (Integer, Integer), Release,
Seizev (Integer), Mode-map (Integer);

L55: Trigger;

L56: Setcod (Integer, Integer, Boolean);

L57: Addressmap-for-SCs (Bit-mode-matrix);

L58: Process-reset;

L59: Process-Reset;

L60: Trigger;

L61: Process-reset.

The states used by the processes are the following:

a) *(RUD) Rx status update and overload channel decoding process (0,4)*

Wait

b) *(BMI) Bit map implementation process (0,4)*

Wait

c) *(DEC) Decoder control process (0,)*

Wait-for-signal.

The signals have the following meaning:

L4 – Rxdata (Integer) – This signal is sent to the transmit side assignment processes upon reception of an assignment message destined for the terminal, which marks a transition from a different state to a data call.

L51 – Assign (Integer, Integer, Call-type) – This signal contains the information included within the assignment message delivered to the process. The variables contain the following information:

SC number, IC number and Call-type.

The last variable can contain three different possibilities:

"Voice", "Data", "Transp".

Within the resource maps of the RUD process, it is possible to have the following different Call-type values:

"Disc", "Voice", "Data", "Transp", "Bank".

L52 – Rxtranspreq (Integer) – A signal given to the 64 kbit/s transparent circuit handler (TCH) upon reception of an assignment message, destined for the terminal, marking a transition from a different state to a transparent call. The Integer value refers to the local IC.

L52 – Rxtransprel (Integer) – The reverse of the above. It is sent when a transition occurs from a transparent call to a different state.

L53 – SC bitmap (Bit-mode-matrix) – A complex signal which defines the mapping between the bits of the incoming bearer and the decoder inputs.

L54 – Seize (Integer, Integer) – This signal contains the information required to connect the output of a decoder to the correct IC number in order to complete the circuit. It contains the local IC number associated with the received channel and the mode in which the decoder should be set (4/5/8 bit mode).

L54 – Seizev (Integer) – This signal is sent in order to associate the output at a decoder used for a voice connection with the correct local IC. The signal contains the local IC number to be used.

L54 – Release – This signal is used to release a designated decoder back into the decoder pool.

L54 – Mode-map (Integer) – This signals contains the mode that is to be used for a decoder that is connected to a voice channel. The parameter contains the actual mode to be used.

L55, L60 – Trigger – This signal is used to indicate that a Setcod signal, as well as an Addressmap-for-SCs signal, should be delivered to the hardware.

L56 – Setcod (Integer, Integer, Boolean) – This signal is issued to generate a specific hardware connection for a certain decoder. The signal contains the local IC number, the mode to be used and a Reset command.

L57 – Addressmap-for-SCs (Bit-mode-matrix) – This complex signal contains the bit associations required to connect the bits of an SC to a decoder.

L58, L59, L61 – Process-reset – This signal is issued by the map-change-handler in association with a map change. The reception of this signal causes the termination of the process that receives it.

A.3.2.2 *The RUD process*

The RUD process is assumed to be generated by the map-change-handler (MCH). The MCH will generate as many instances of the RUD process as is required. There will be one instance of the RUD process for each pool containing traffic destined for this unit (up to four). The RUD process will process the assignment message contained within the pool that it has been assigned to and generate the required actions based on the contents of this message. When the RUD process is created by the map-change-handler, a number of variables are passed along to the RUD process. These are:

– *dcdlist* – The list contains a list of the decoder numbers that the process may use. The list excludes those decoders used to handle pre-assigned channels.

– *bt* – The total number of 4-bit channels that the remote correspondents pool contains.

– *iclist* – This list includes all IC numbers that may be contained in the received assignment message of the received pool to which the RUD instance is assigned. Pre-assigned IC numbers are excluded.

– *sclist* – This list contains the SC numbers that may be contained in the received assignment message of the received pool to which the RUD instance is assigned. Pre-assigned SCs are excluded.

– *pre* – This array contains the IC numbers of any pre-assigned channels that the remote DCME may have in the pool to which the RUD instance is assigned.

– *presc* – This array contains the SC numbers associated with the pre-assigned channels that the remote DCME may use in the pool to which the RUD instance is assigned. Only the even numbered SCs for transparent channels are given.

– *premode* – This array contains the mode that is to be associated with each pre-assigned SC given in "presc".

– *sel* – This array contains the decoders that are to be used in association with pre-assigned channels described above.

– *ptot* – This integer value contains the total number of pre-assigned channels that are to be dealt with at start-up.

– *ad* – This array uses the remote IC number to index the local IC numbers that make up the circuit. If the remote IC is not addressed to the unit, the number contained will be zero.

– *bit bank* – This array, with a maximum of 12 entries, contains in ascending SC number order, the SC numbers that are to be used for bit bank handling. At start-up, the array will contain those SCs that are to be used for bit banks in order to handle those SCs that are to be pre-assigned 40 kbit/s channels.

– *btot* – This Integer value contains the total number of bit banks that are in use at any given time. At start-up, this value equals the number of bit banks that are required to handle the pre-assigned 40 kbit/s channels.

– *DECPID(i)* – This process identifier array gives the correct addresses to any required decoder process with a given number. It is used when addressing signals towards the DEC processes.

– *BMIPID* – This process identifier variable is used to address signals to the correct BMI process.

– *s* – This integer variable defines the lowest number of bits/samples permitted. Its value is either three, for 3-bit encoding, or two, for 2-bit encoding.

Arrays are used as a resource map for the receive process. These are:

– *sat(nr)=bc* – This array uses the remote IC number to index the SC number that the IC is connected to. The array is initialized as all zeros.

– *ic(bc)=nr* – This array uses the SC numbers to index the remote IC number to which it is connected. The array is initialized as all zeros.

– *typ(bc)="Call-type"* – This array uses the received SC numbers to index the type of connection being received on that SC. This value can be either:

"Transp", "Data", "Voice", "Disc", "Bank".

The array is initialized as all "Disc".

– *dec(lnr)=dcd* – This array uses the local (transmit) IC number to index the decoder to which a received channel, destined for this unit, is connected. It is initialized as all zeros.

The process also uses two lists in order to be able to generate the overload bit positions. These are:

Voicelist, Overloadlist

They are handled in the same way as on the transmit side. The following variables and procedure calls are used within the RUD process.

- *i* – Counter.

- *prep* – This integer is used to delay deletion of bit banks just after a bit bank has been assigned. Normally, an attempt will be made to delete bit banks which are not required after each assignment message has been processed. This is done by the procedure Bit-bank-handling. There may, however, be a delay of one or two DCME frames after the assignment of a bit bank has occurred until the data assignment is made which required the generation of a new bit bank. This is due to the possible need to reassign a voice call before the data call can be assigned. When a bit bank is assigned, prep is set to 1. This will cause the procedure to bypass deletion of possible bit banks when it is invoked. After a maximum of two DCME frames, the deletion will once more be started, i.e., when a bit bank is created in frame i the deletion will start in frame i + 2.

- *Count data (difference)* – This procedure checks to see if there is a possibility to delete a bit bank. This is checked by comparing the amount of data calls and pre-assigned 40 kbit/s calls with the amount of bit banks in use. If there are too many bit banks, the variable "Difference" is set to TRUE, otherwise FALSE. If the procedure finds that there are too few bit banks in comparison to the number of data calls and pre-assigned 40 kbit/s calls being handled, the deletion of bit banks is stopped. This is done by setting the variable "Difference" to FALSE.

- *Difference* – A Boolean variable described above.

- *scr* – This variable contains the SC number contained within the received assignment message.

- *icr* – This variable contains the remote IC number that "scr" is to be connected to according to the received assignment message.

- *flag* – This variable contains the type of connection that is specified by the assignment message (voice, data, transp).

- *Check content (scr, icr, flag, correct)* – This procedure checks if the assignment message content is valid. If this is the case, the value TRUE is delivered in the variable "Correct", the value FALSE if this is not the case. If the message is incorrect, it shall be disregarded. The checks that are assumed to be made are the following:

  - "icr" is in the range of numbers that the remote destination may use. This implies that it is part of the "iclist".

  - "scr" is in the range of the DSI pool (including overload channels) and is not used for a pre-assigned connection, i.e., it is included within the "sclist".

  - The connection proposed does not violate any strict rules, such as connection of a transparent call to an odd numbered SC or the connection of something other than "voice" to an overload channel, i.e., a channel with a number higher than bt.

- *Correct* – A Boolean variable described above.

- *Again* – This Boolean variable is used to disconnect both SCs when a transparent disconnect message is received and the SC is declared as something else than transparent.

– *Delete overload(sc)* – This procedure removes the SC number "sc" from the overloadlist.

– *k* – The SC number that "icr" was connected to previously.

– *nr* – The IC number that was connected to "scr" previously.

– *nr1* – The IC number that was connected to "scr+1" previously.

– *Insert in voicelist (sc)* – This procedure inserts the SC number "sc" into the voicelist in its appropriate place.

– *Remove from bit bankarray (sc, btot)* – This procedure removes the SC number "sc" from the bit bank array and pushes the indexed values above "SC" down one index position. It also updates the value of btot.

– *Insert2 in voicelist (s1, s2)* – This procedure inserts the SC numbers "s1" and "s2" into their appropriate place in the voicelist.

– *Insert in overloadlist (sc)* – This procedure inserts the SC number "sc" into its appropriate place in the overloadlist.

– *Make room in bit bankarray (nw, sc, bit bank)* – This procedure delivers the index of the bit bankarray where the SC number "sc" is supposed to fit in accordance with the rule for handling this array in variable nw. Starting at the greatest index used, "k" indexing a non-zero value, all indexed values down to "nw + 1" are moved up one index, leaving a space in the array at index "nw". The entry which should have the value "SC" associated with it is thus given the variable "nw".

– *nw* – An integer variable described above.

– *Remove from voicelist (sc)* – This procedure removes the SC number "sc" from the voicelist.

– *Select decoders (dcd)* – This procedure selects an unused decoder out of the "pool" and delivers the result in the variable "dcd". An unused decoder is a decoder which is part of the dcdlist but one which is not indexed by the "dec" array at a given moment. It should be noted that this pool could consist of one decoder per local IC number should this be the way in which a manufacturer elects to implement this.

– *dcd* – An Integer variable described above.

– *Remove2 from voicelist (s1, s2)* – This procedure removes SC numbers "s1" and "s2" from the voicelist.

– *Generate addresses (bitmap, mode array)* – This procedure uses the value of icr as a pointer and generates the modes to be used by the decoders handling voice connections in accordance with the current situation regarding overload channels. It also generates the addresses required in order for bits on the incoming bearer to be mapped to the correct bit positions at the input of the decoders. It puts the contents of this information into the signals Mode-map and SC bitmap. When there are not enough bits in the bit banks, the bits in the bit banks are distributed from the lowest SC numbered data channel to the highest SC numbered data channel. The data channel(s) which cannot be accommodated by the existing bit bank channels receives a dummy fifth bit set to "0". When there is not enough bits to create all existing overload channels, the available bits are distributed from the lowest SC numbered overload channel to the highest numbered overload channel. The overload channel(s) which cannot be accommodated receives dummy bits set to "0".

– *bitmap* – A complex variable which contains the bit map generated by the procedure Generate-addresses every DCME frame.

– *mode array* – An Integer array which contains the number of bits each decoder will receive each DCME frame.

A.3.2.3 *Transitions allowed within the RUD process*

It should be noted that the logic diagrams contained for the receive side handling allow for transitions that should not occur unless assignment messages are missed. These transitions have been included in order to achieve the quickest recovery of the bearer frame after losses of assignment messages have occurred. A list is given of these impossibilities below.

1) Explicit disconnection of a channel declared as "bank".
2) Implicit disconnection of an overload channel.
3) Implicit disconnection of channels declared as "transp".
4) Connection of "icr" to "scr" where "scr" is declared as "bank".
5) Connection of "icr" to "scr" where "scr" is not connected to "nr" but is declared as "transp".
6) Connection of "icr" to "scr" where "scr+1" is declared as "bank" and "flag" is "transp".
7) Implicit changes from "transp" to something else.

The complete description of the above has resulted in a significant number of diagrams needed to describe the receive protocols.

A.3.2.4 *The DEC process*

The DEC process is created by the map-change-handler at system start-up. It contains the following variables and procedure calls:

– *ic* – The local IC number to which the decoder is currently connected.
– *sc* – The SC number to which the decoder is currently connected.
– *mode* – The current mode of the decoder.
– *dec reset* – This variable stores the possibility of resetting the decoder. It is TRUE if a reset is to be made, otherwise FALSE.
– *change* – This variable is TRUE if a change in future values has occurred since the last Trigger signal, otherwise FALSE.
– *nr* – The transmit IC number that the decoder output should be given to.
– *fic* – The future IC number.
– *fmode* – The future mode.
– *cic* – The current local IC number.
– *cmode* – The current mode.
– *amd* – An Integer variable which stores the mode received within the mode-map signal.
– *md* – An Integer variable which stores the mode received within the Seize signal.
– *Store (fic, fmode)* – This procedure stores the parameters at the bottom of a queue.
– *Retrieve (cic, cmode)* – This procedure retrieves the appropriately delayed values stored in the queue, delivers the contents in the variables "cic", and "cmode". At initialization, the queue is to contain 0 in all of its positions.

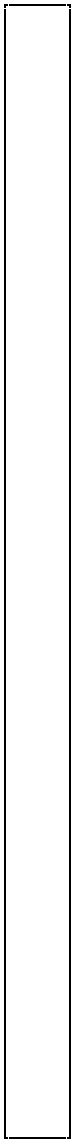
  It should be noted that a Setcod signal containing the values (0, 0, FALSE) should not cause the hardware to generate any type of connection.

A.3.2.5 *The BMI process*

This process will only delay the incoming signal SC-bitmap by an appropriate amount of DCME frames before sending the delayed contents in the signal Addressmap-for-SCs. The process contains the following procedure calls:

– *Store (bitmap)* – This procedure stores the contents of the SC-bitmap signal at the bottom of the queue.
– *Retrieve (bitmap)* – This procedure removes the appropriately delayed information stored in the queue and loads this information into the Address-map-for-SCs signal. At initialization, the queue contains information such that no connection will be generated when removing the contents and generating a signal to the hardware.

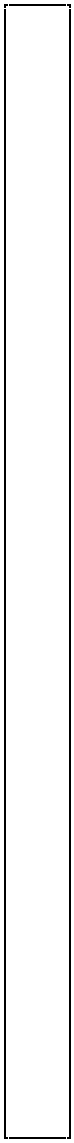
FIGURE **(sans numero)**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(sans numero)**

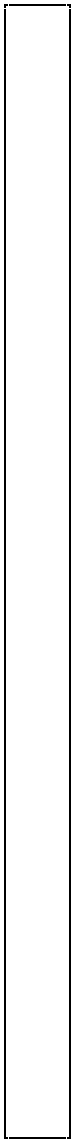
Manuscrit: **COM XV-R 39 -E/F/S** Page **265**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(sans numero)**

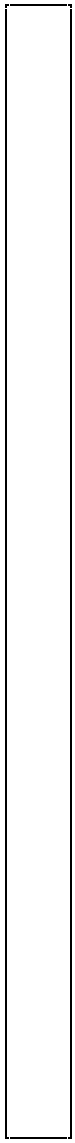
Manuscrit: **COM XV-R 39 -E/F/S** Page **266**



MONTAGE: Monter les figures dans l´anglais uniquement.

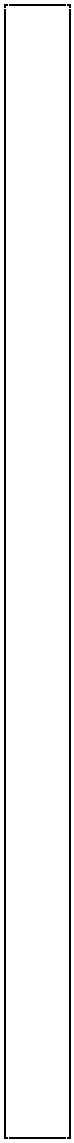
FIGURE **(sans numero)**

Manuscrit: **COM XV-R 39 -E/F/S** Page **267**



MONTAGE: Monter les figures dans l´anglais uniquement.

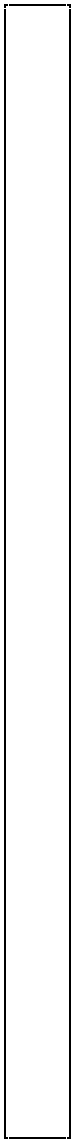
FIGURE **(sans numero)**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(sans numero)**

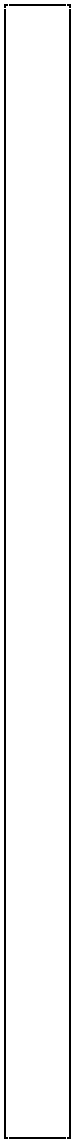
Manuscrit: **COM XV-R 39 -E/F/S** Page **269**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(sans numero)**

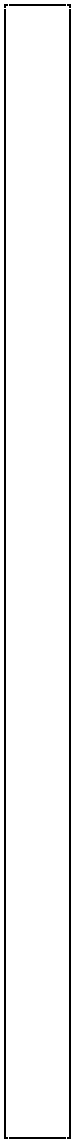
Manuscrit: **COM XV-R 39 -E/F/S** Page **270**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504100-90)**

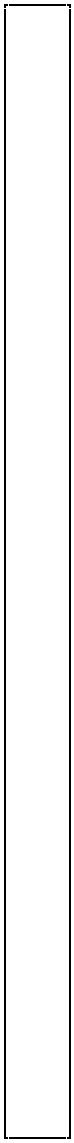
Manuscrit: **COM XV-R 39 -E/F/S** Page **271**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504110-90)**

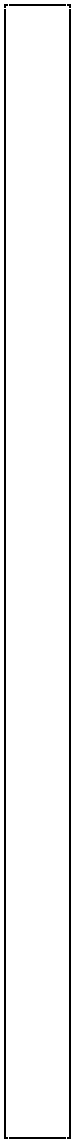
Manuscrit: **COM XV-R 39 -E/F/S** Page **272**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504120-90)**

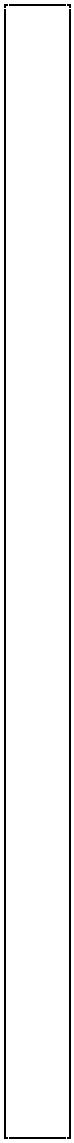
Manuscrit: **COM XV-R 39 -E/F/S** Page **273**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504130-90)**

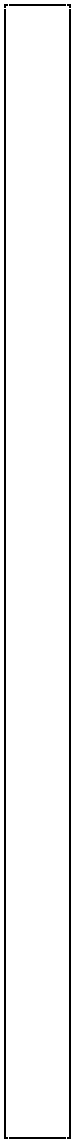
Manuscrit: **COM XV-R 39 -E/F/S** Page **274**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504140-90)**

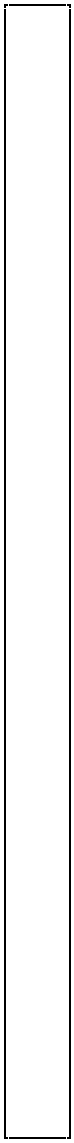
Manuscrit: **COM XV-R 39 -E/F/S** Page **275**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504150-90)**

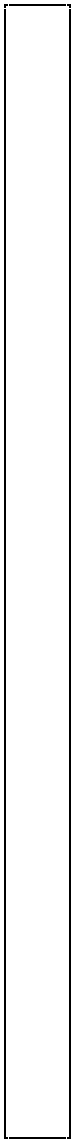
Manuscrit: **COM XV-R 39 -E/F/S** Page **276**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504160-90)**

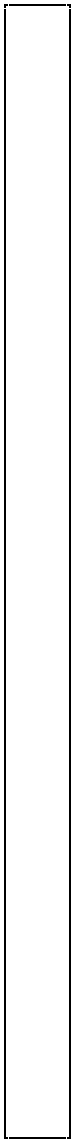
Manuscrit: **COM XV-R 39 -E/F/S** Page **277**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504170-90)**

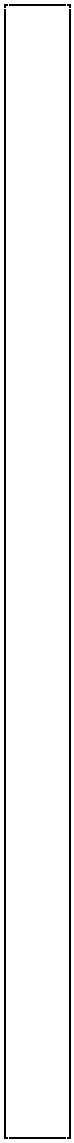
Manuscrit: **COM XV-R 39 -E/F/S** Page **278**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504180-90)**

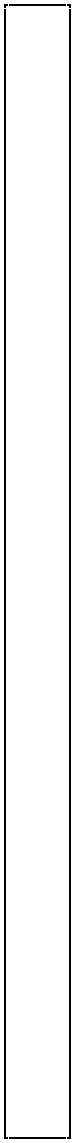
Manuscrit: **COM XV-R 39 -E/F/S** Page **279**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504190-90)**

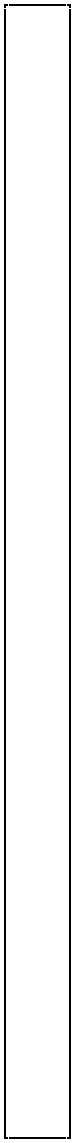
Manuscrit: **COM XV-R 39 -E/F/S** Page **280**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504200-90)**

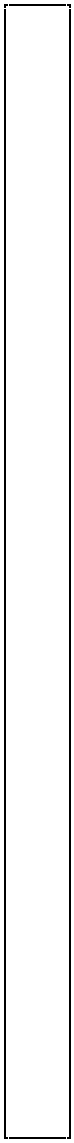
Manuscrit: **COM XV-R 39 -E/F/S** Page **281**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504210-90)**

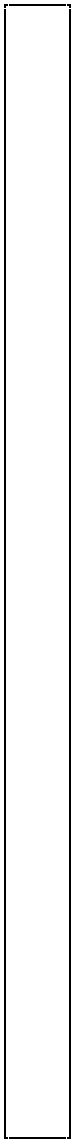
Manuscrit: **COM XV-R 39 -E/F/S** Page **282**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504220-90)**

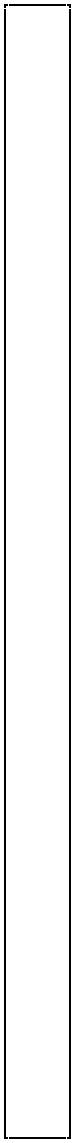
Manuscrit: **COM XV-R 39 -E/F/S** Page **283**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504230-90)**

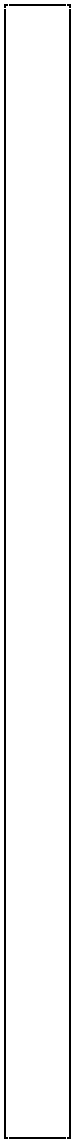
Manuscrit: **COM XV-R 39 -E/F/S** Page **284**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504240-90)**

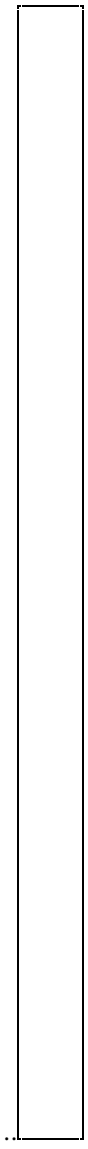
Manuscrit: **COM XV-R 39 -E/F/S** Page **285**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504250-90)**

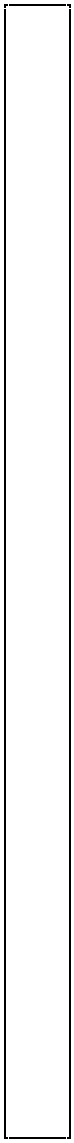
Manuscrit: **COM XV-R 39 -E/F/S** Page **286**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504260-90)**

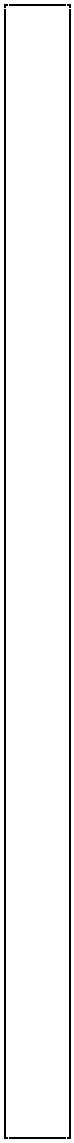
Manuscrit: **COM XV-R 39 -E/F/S** Page **287**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504270-90)**

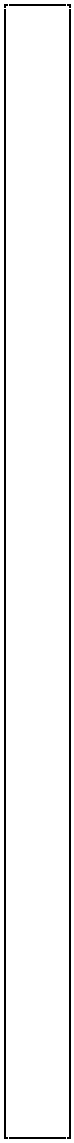
Manuscrit: **COM XV-R 39 -E/F/S** Page **288**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504280-90)**

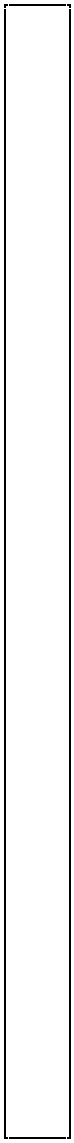
Manuscrit: **COM XV-R 39 -E/F/S** Page **289**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504290-90)**

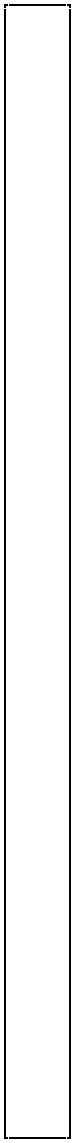
Manuscrit: **COM XV-R 39 -E/F/S** Page **290**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504300-90)**

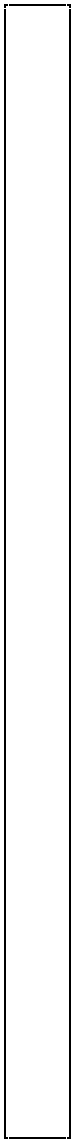
Manuscrit: **COM XV-R 39 -E/F/S** Page **291**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504310-90)**

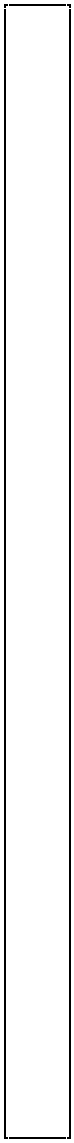
Manuscrit: **COM XV-R 39 -E/F/S** Page **292**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504320-90)**

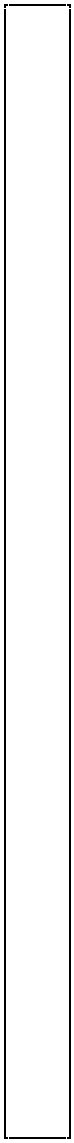
Manuscrit: **COM XV-R 39 -E/F/S** Page **293**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504330-90)**

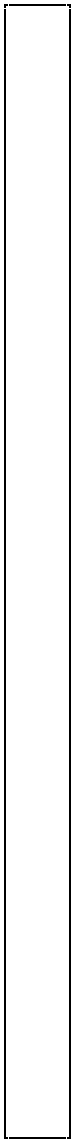
Manuscrit: **COM XV-R 39 -E/F/S** Page **294**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504340-90)**

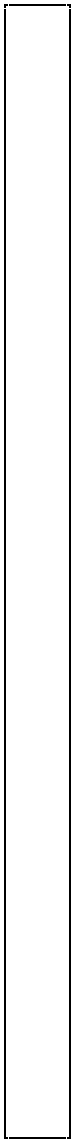
Manuscrit: **COM XV-R 39 -E/F/S** Page **295**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504350-90)**

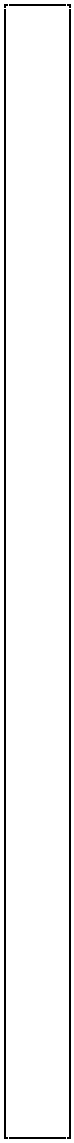
Manuscrit: **COM XV-R 39 -E/F/S** Page **296**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504360-90)**

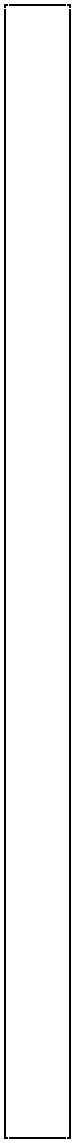
Manuscrit: **COM XV-R 39 -E/F/S** Page **297**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504370-90)**

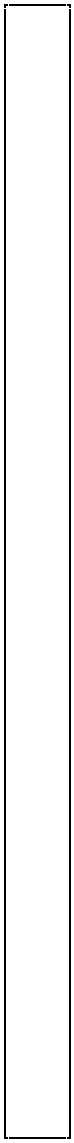
Manuscrit: **COM XV-R 39 -E/F/S** Page **298**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504380-90)**

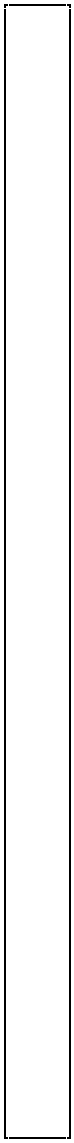
Manuscrit: **COM XV-R 39 -E/F/S** Page **299**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504390-90)**

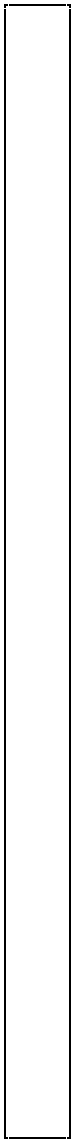
Manuscrit: **COM XV-R 39 -E/F/S** Page **300**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504400-90)**

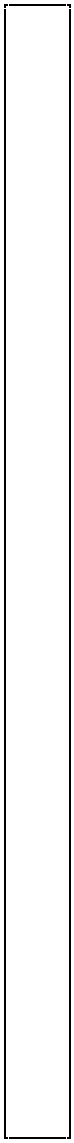
Manuscrit: **COM XV-R 39 -E/F/S** Page **301**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504410-90)**

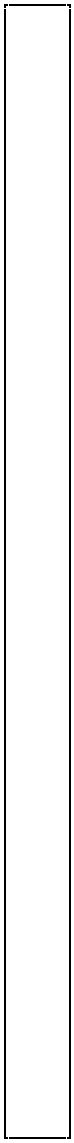
Manuscrit: **COM XV-R 39 -E/F/S** Page **302**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504420-90)**

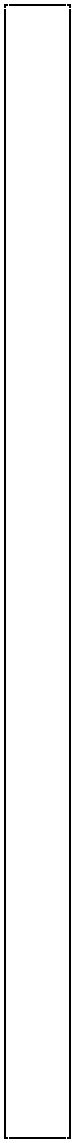
Manuscrit: **COM XV-R 39 -E/F/S** Page **303**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504430-90)**

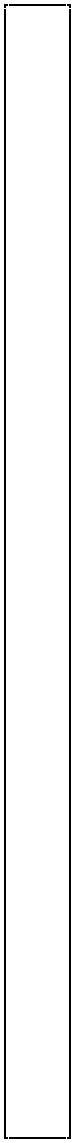
Manuscrit: **COM XV-R 39 -E/F/S** Page **304**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504440-90)**

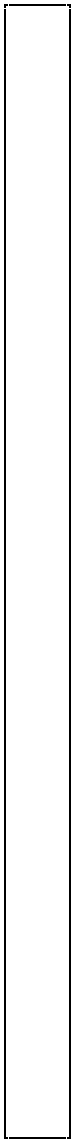
Manuscrit: **COM XV-R 39 -E/F/S** Page **305**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504450-90)**

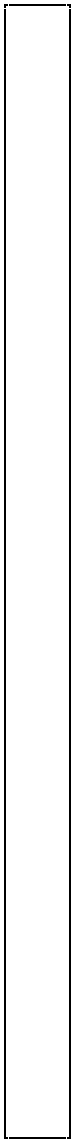
Manuscrit: **COM XV-R 39 -E/F/S** Page **306**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504460-90)**

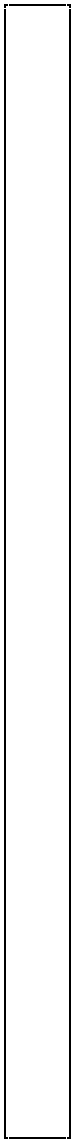
Manuscrit: **COM XV-R 39 -E/F/S** Page **307**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504470-90)**

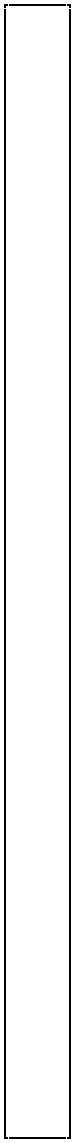
Manuscrit: **COM XV-R 39 -E/F/S** Page **308**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504480-90)**

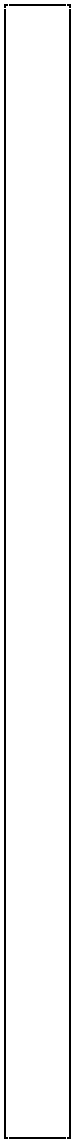
Manuscrit: **COM XV-R 39 -E/F/S** Page **309**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504490-90)**

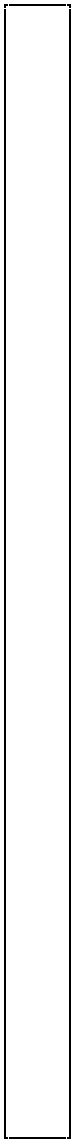
Manuscrit: **COM XV-R 39 -E/F/S** Page **310**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504500-90)**

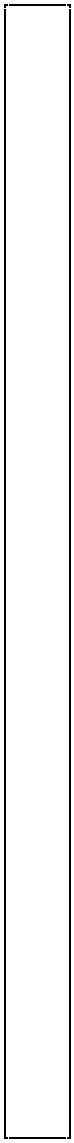
Manuscrit: **COM XV-R 39 -E/F/S** Page **311**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504510-90)**

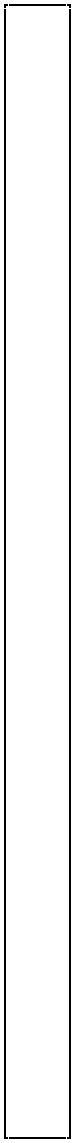
Manuscrit: **COM XV-R 39 -E/F/S** Page **312**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504520-90)**

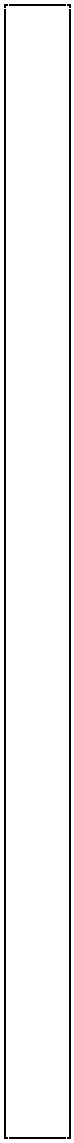
Manuscrit: **COM XV-R 39 -E/F/S** Page **313**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504530-90)**

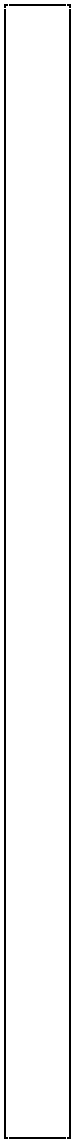
Manuscrit: **COM XV-R 39 -E/F/S** Page **314**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504540-90)**

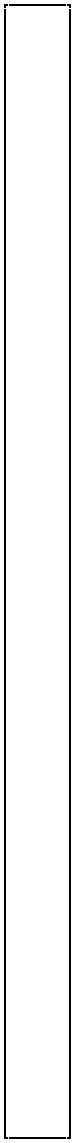
Manuscrit: **COM XV-R 39 -E/F/S** Page **315**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504550-90)**

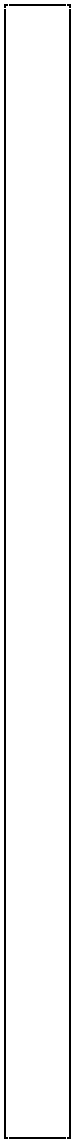
Manuscrit: **COM XV-R 39 -E/F/S** Page **316**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504560-90)**

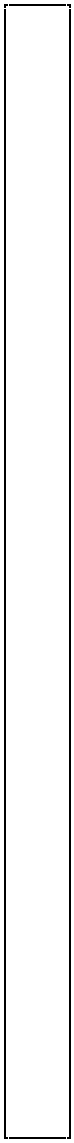
Manuscrit: **COM XV-R 39 -E/F/S** Page **317**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504570-90)**

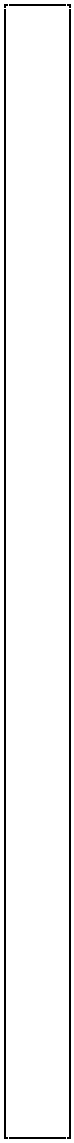
Manuscrit: **COM XV-R 39 -E/F/S** Page **318**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504580-90)**

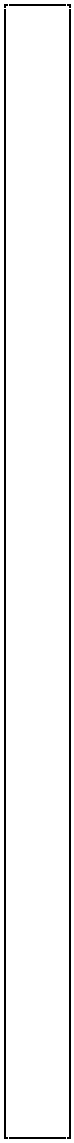
Manuscrit: **COM XV-R 39 -E/F/S** Page **319**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504590-90)**

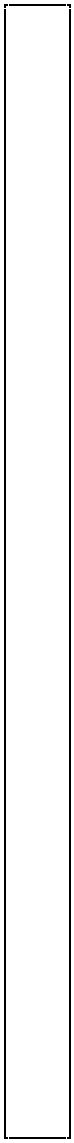
Manuscrit: **COM XV-R 39 -E/F/S** Page **320**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504600-90)**

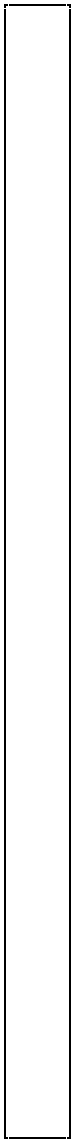
Manuscrit: **COM XV-R 39 -E/F/S** Page **321**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504610-90)**

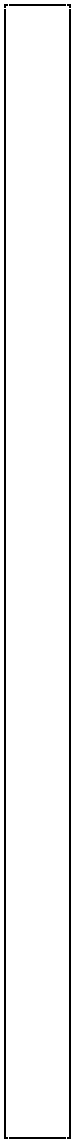
Manuscrit: **COM XV-R 39 -E/F/S** Page **322**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504620-90)**

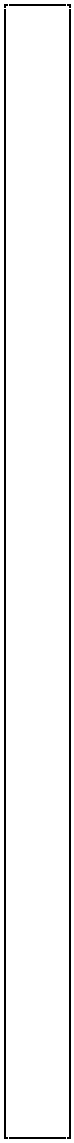
Manuscrit: **COM XV-R 39 -E/F/S** Page **323**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504630-90)**

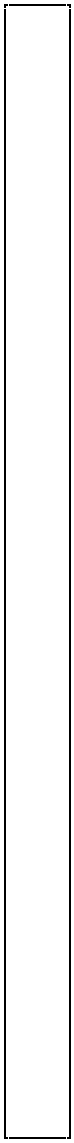
Manuscrit: **COM XV-R 39 -E/F/S** Page **324**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504640-90)**

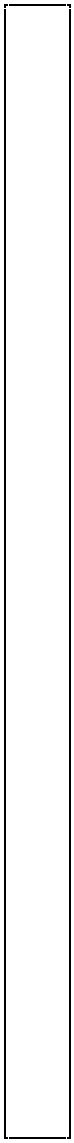
Manuscrit: **COM XV-R 39 -E/F/S** Page **325**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504650-90)**

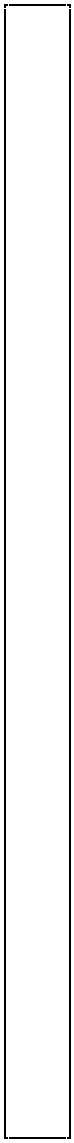
Manuscrit: **COM XV-R 39 -E/F/S** Page **326**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504660-90)**

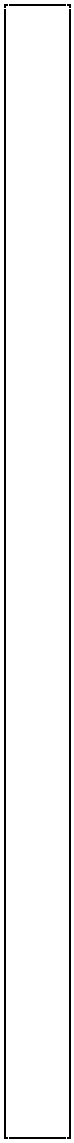
Manuscrit: **COM XV-R 39 -E/F/S** Page **327**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504670-90)**

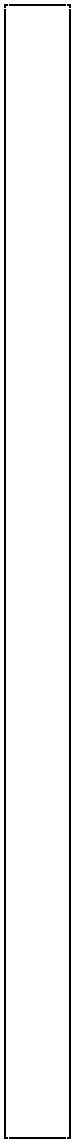
Manuscrit: **COM XV-R 39 -E/F/S** Page **328**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504680-90)**

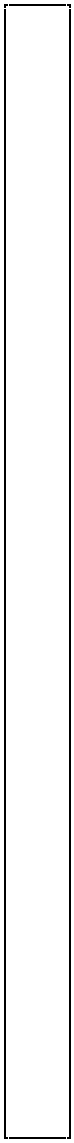
Manuscrit: **COM XV-R 39 -E/F/S** Page **329**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504690-90)**

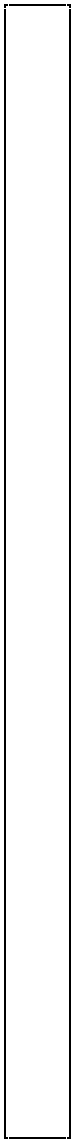
Manuscrit: **COM XV-R 39 -E/F/S** Page **330**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504700-90)**

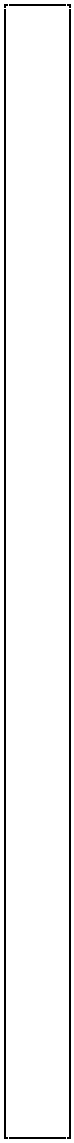
Manuscrit: **COM XV-R 39 -E/F/S** Page **331**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504710-90)**

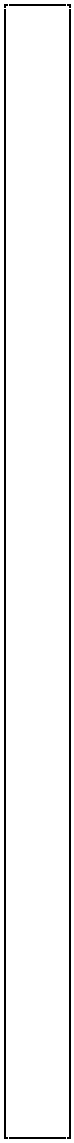
Manuscrit: **COM XV-R 39 -E/F/S** Page **332**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504720-90)**

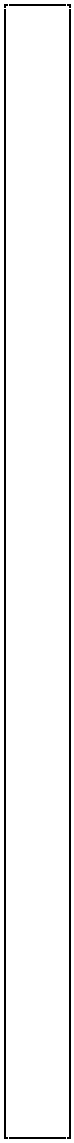
Manuscrit: **COM XV-R 39 -E/F/S** Page **333**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504730-90)**

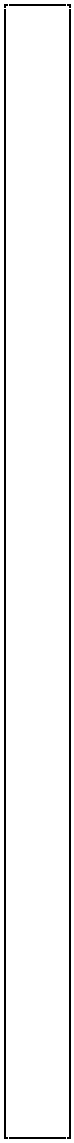
Manuscrit: **COM XV-R 39 -E/F/S** Page **334**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504740-90)**

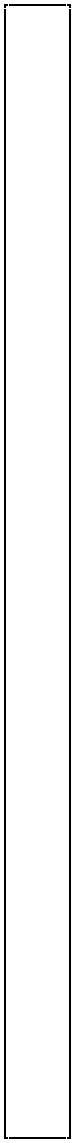
Manuscrit: **COM XV-R 39 -E/F/S** Page **335**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504750-90)**

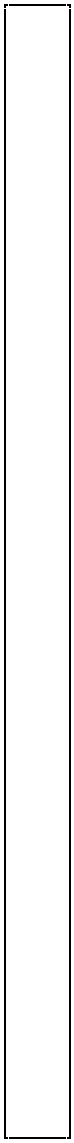
Manuscrit: **COM XV-R 39 -E/F/S** Page **336**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504760-90)**

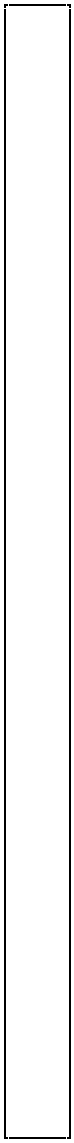
Manuscrit: **COM XV-R 39 -E/F/S** Page **337**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504770-90)**

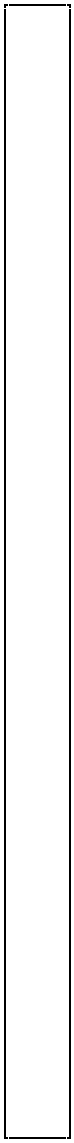
Manuscrit: **COM XV-R 39 -E/F/S** Page **338**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504780-90)**

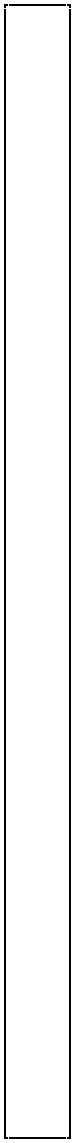
Manuscrit: **COM XV-R 39 -E/F/S** Page **339**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504790-90)**

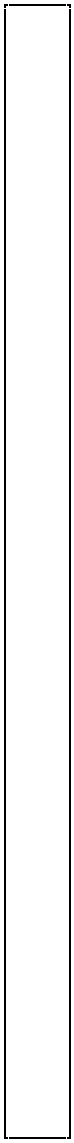
Manuscrit: **COM XV-R 39 -E/F/S** Page **340**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504800-90)**

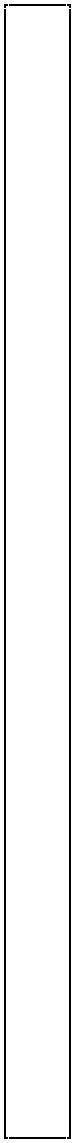
Manuscrit: **COM XV-R 39 -E/F/S** Page **341**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504810-90)**

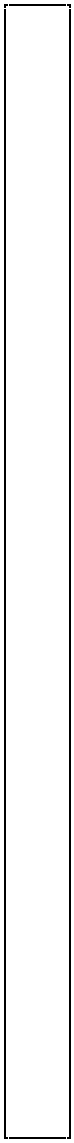
Manuscrit: **COM XV-R 39 -E/F/S** Page **342**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504820-90)**

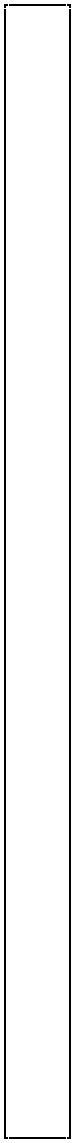
Manuscrit: **COM XV-R 39 -E/F/S** Page **343**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504830-90)**

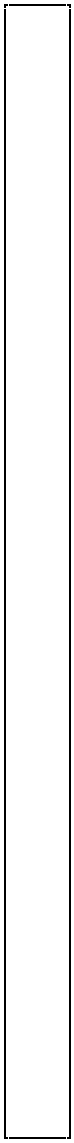
Manuscrit: **COM XV-R 39 -E/F/S** Page **344**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504840-90)**

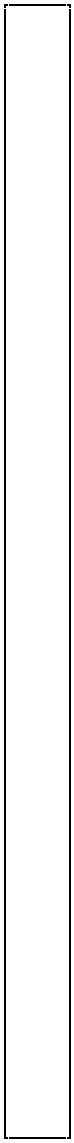
Manuscrit: **COM XV-R 39 -E/F/S** Page **345**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504850-90)**

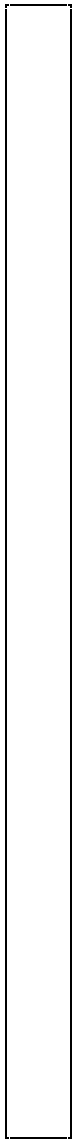
Manuscrit: **COM XV-R 39 -E/F/S** Page **346**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504860-90)**

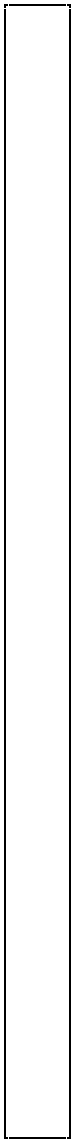
Manuscrit: **COM XV-R 39 -E/F/S** Page **347**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504870-90)**

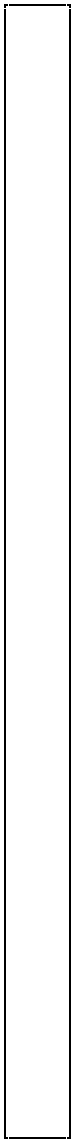
Manuscrit: **COM XV-R 39 -E/F/S** Page **348**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504880-90)**

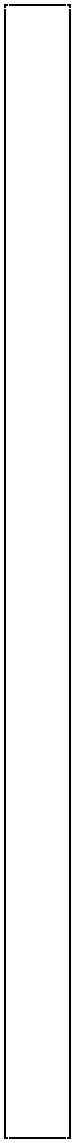
Manuscrit: **COM XV-R 39 -E/F/S** Page **349**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504890-90)**

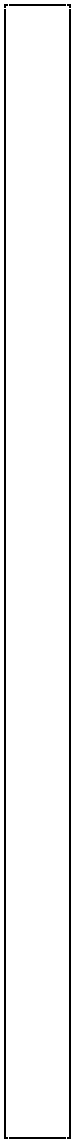
Manuscrit: **COM XV-R 39 -E/F/S** Page **350**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504900-90)**

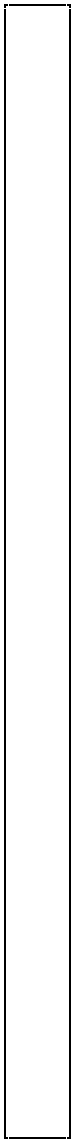
Manuscrit: **COM XV-R 39 -E/F/S** Page **351**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504910-90)**

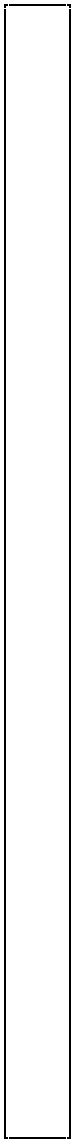
Manuscrit: **COM XV-R 39 -E/F/S** Page **352**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504920-90)**

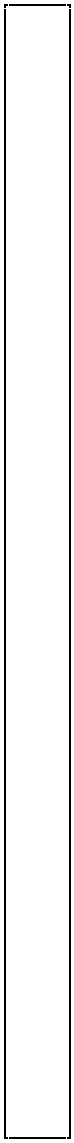
Manuscrit: **COM XV-R 39 -E/F/S** Page **353**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504930-90)**

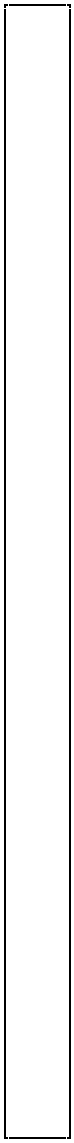
Manuscrit: **COM XV-R 39 -E/F/S** Page **354**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504940-90)**

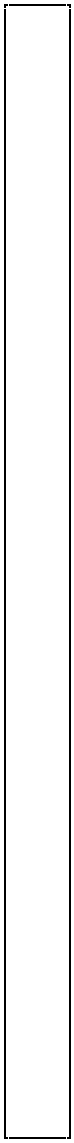
Manuscrit: **COM XV-R 39 -E/F/S** Page **355**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504950-90)**

Manuscrit: **COM XV-R 39 -E/F/S** Page **356**



MONTAGE: Monter les figures dans l´anglais uniquement.

A.3.3 *Logic diagrams for on-demand transparent circuit handling*

The logic diagrams in this section of Annex A.3 are supplemental to the description of the on-demand transparent circuit handling given in section 8 of the Recommendation and includes a user optional TCH/DLC interaction override facility. The on-demand transparent circuit handling procedure is contained within one block:

TCH: transparent-channel-handling-block.

A.3.3.1 *The TCH block*

The TCH block contains one process and the following signals:

L5: Transpreq, Transprel

L30: S64, R64

L31: S64Ack, S64Nack, R64Ack Out-of-Service, Back-in-Service

L32: AD64, DD64

L33: Process-reset

L34: Override, No-Override

L35: Man-Reset

L52: Rxtranspreq, Rxtransprel

The states used by the process are the following:

(TCH) transparent circuit handling process (0,)

Not-64, Blocked, Circuit Out-of-Service, Connect-called-64, Connect-calling-64, Establish-forward-64, Disestablish-forward-64, Disestablish-backward-64, Autorecovery-64, Spurious-recovery.

The signals have the following meaning:

L5 – See previous explanation in Appendix I.

L30 – The S64 and R64 signals arrive from the SIU where they have been translated from the format used by the real originator, namely the ISC exchange. They imply that a transparent call shall be established or terminated.

L31 – The S64Ack, S64Nack and R64Ack signals are the responses that the TCH process generate upon reception of the L30 signals. The Out-of-Service and Back-in-Service signals are used to indicate to and from unavailability.

L32 – The AD64 and DD64 signals arrive from the DLC block in the DCME and indicate that the unit should stop accepting any new transparent requests from the ISC or that the DCME should start accepting requests for transparent connections from the ISC, respectively.

L33 – The Process-reset signal arrives from the MCH and causes a termination of the process instance that receives it.

L34 – The Override and No-Override signals are generated manually by the operator and indicate that the TCH/DLC interaction is disabled or enabled, respectively (see Note).

L35 – The Man-Reset signal is used by the operator to place an Out-of-Service circuit to Back-in-Service.

L52 – See previous explanation in Appendix II.

The TCH process is assumed to be created by the MCH at system start-up and after a map change. There is one instance of the process for every local IC handled by the DCME, according to its configuration data. The process uses the following variables:

disabled – This Boolean variable is TRUE when the TCH/DLC interaction is disabled, otherwise FALSE. It is set and reset by arrival of the signals Override and No-Override, respectively (see Note).

dlcon – This variable stores the current DLC condition for the IC handled by the process. If DLC is "ON" the variable is True, if "OFF" False.

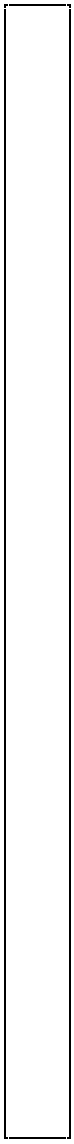
T1, T2, T3, T4 – Different timer values used by the TCH process instance.

ti – A timer variable.

Note – If the manual override is not implemented these signals do not exist. If the manual override is not implemented the variable is always FALSE.

FIGURE **(sans numero)**

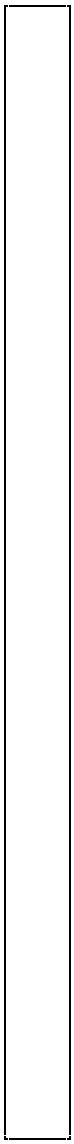
Manuscrit: **COM XV-R 39 -E/F/S** Page **359**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(sans numero)**

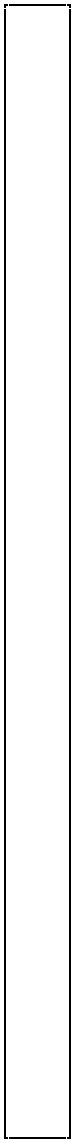
Manuscrit: **COM XV-R 39 -E/F/S** Page **360**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(sans numero)**

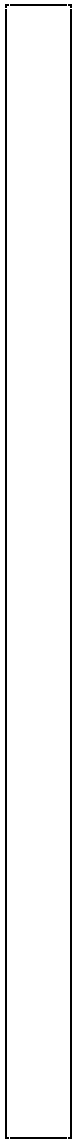
Manuscrit: **COM XV-R 39 -E/F/S** Page **361**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(sans numero)**

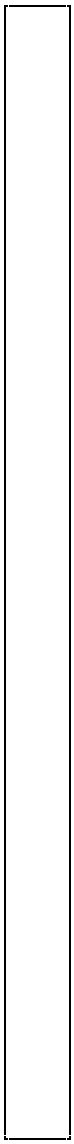
Manuscrit: **COM XV-R 39 -E/F/S** Page **362**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504960-90)**

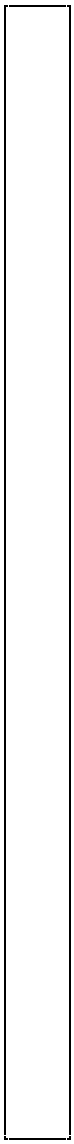
Manuscrit: **COM XV-R 39 -E/F/S** Page **363**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504970-90)**

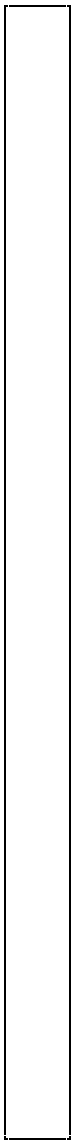
Manuscrit: **COM XV-R 39 -E/F/S** Page **364**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504980-90)**

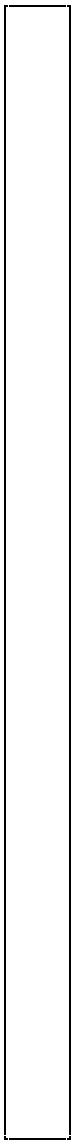
Manuscrit: **COM XV-R 39 -E/F/S** Page **365**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1504990-90)**

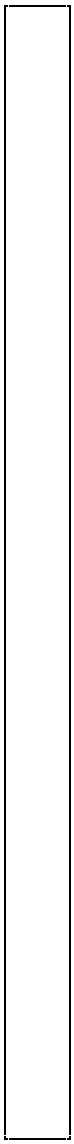
Manuscrit: **COM XV-R 39 -E/F/S** Page **366**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1505000-90)**

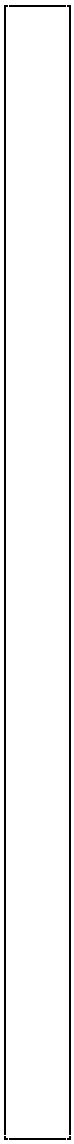
Manuscrit: **COM XV-R 39 -E/F/S** Page **367**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1505010-90)**

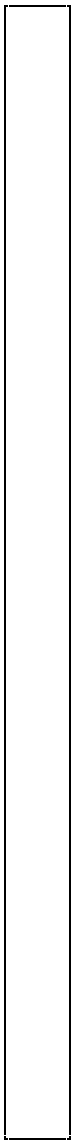
Manuscrit: **COM XV-R 39 -E/F/S** Page **368**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1505020-90)**

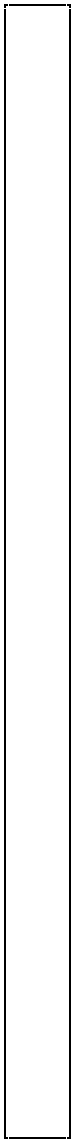
Manuscrit: **COM XV-R 39 -E/F/S** Page **369**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1505030-90)**

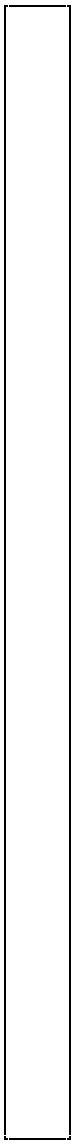
Manuscrit: **COM XV-R 39 -E/F/S** Page **370**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1505040-90)**

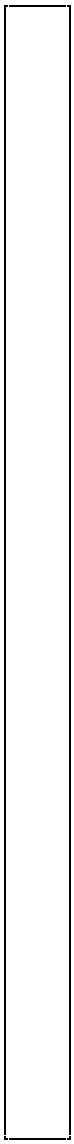
Manuscrit: **COM XV-R 39 -E/F/S** Page **371**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1505050-90)**

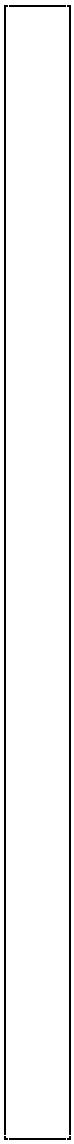
Manuscrit: **COM XV-R 39 -E/F/S** Page **372**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1505060-90)**

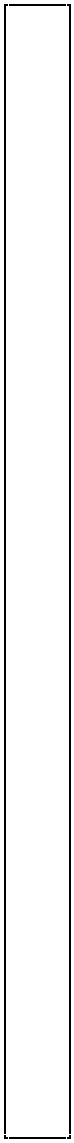
Manuscrit: **COM XV-R 39 -E/F/S** Page **373**



MONTAGE: Monter les figures dans l´anglais uniquement.

FIGURE **(T1505070-90)**

Manuscrit: **COM XV-R 39 -E/F/S** Page **374**



MONTAGE: Monter les figures dans l´anglais uniquement.