EXTRACTING DATA SECTIONS FROM A TRANSMITTED DATA STREAM

The present invention relates to an apparatus for and a method of storing one or more sections of a transmitted data stream such as an MPEG bitstream containing video and audio television data in addition to the sections of data.

The advent of digital transmission systems intended primarily for broadcasting television signals, in particular but not exclusively satellite television systems, has opened up the possibility of using such systems for other purposes. One of these is to provide interactivity with the end user.

One way of doing this is to run an application on the receiver/decoder through which the television signal is received. The code for the application could be permanently stored in the receiver/decoder. However, this would be rather limiting. Preferably, the receiver/decoder should be able to download the code for a required application. In this way, more variety may be provided, and applications can be updated as required without any action on the part of the user.

In computer systems, particularly systems where the computer system is incorporated in some larger system such as a receiver/decoder for digital television, the amount of memory is often limited. This means that the memory must be organized so that the use of memory space by the various functions required by the system is minimized. In addition, it may also be necessary to minimize the time required to access at least some parts of the memory.

Accordingly, the present invention is particularly, but not exclusively, concerned with the downloading into the memory of the receiver/decoder only that data specifically required by an application.

In a first aspect the present invention provides a method of storing at least one of a plurality of sections of a transmitted data stream, said one section having at least one section characteristic, said method comprising the steps of:

receiving the data stream;

filtering said one section from said data stream according to at least said one section characteristic; and

storing said one section.

This filtering step enables only the sections required by an application to be loaded into the memory of a receiver/decoder.

In a preferred embodiment, the filtering step comprises the steps of:

specifying at least one filtering characteristic, each characteristic comprising a filtering criterion and a value for that filtering criterion;

comparing the or each section characteristic with a respective filtering characteristic; and

filtering said one section from said data stream when the or each section characteristic corresponds to the respective filtering characteristic.

A second aspect of the present invention provides a method of storing a plurality of sections of a transmitted data stream using the above method, each section being stored in a respective memory location.

Preferably, this method further comprises the step of:

specifying at least one data filtering characteristic, each data filtering characteristic comprising a data filtering criterion and a value for that data filtering criterion;

comparing data stored in each stored section with a respective data filtering characteristic; and

enabling said stored section to be replaced in said memory when said data stored in that stored section does not correspond with the respective data filtering characteristic.

Preferably, the method further comprises the step of supplying a section identification signal to an application when said data stored in said stored section corresponds to

said respective data filtering characteristic.

The section identification signal preferably comprises the storage address of said one section.

In a third aspect, the present invention provides a method of storing a group of sections according to the above method, said sections being cyclically transmitted in said data stream, each group including a first section and a last section; said method comprising the steps of:

storing the first section;

subsequently downloading sections successively transmitted immediately after said first section, each section being successively stored in a respective memory location, until either said last section is downloaded or a predetermined number of sections have been stored.

In a fourth aspect, the present invention provides a method of storing a group of sections according to the above method, said sections being cyclically transmitted in said data stream, said group including a first section and a last section; said method comprising the steps of:

downloading the first section;

subsequently downloading sections successively transmitted immediately after said first section, each section being successively stored in a respective memory location, until either said last section is downloaded or a predetermined number of sections have been stored whereupon the sections stored in the memory locations are successively overwritten by subsequently downloaded sections until said last section is downloaded; and

storing said last section in said memory.

In a fifth aspect, the present invention provides a method of storing a group of sections cyclically transmitted in a data stream, said group including a first section and a last section; said method comprising the steps of:

receiving the data stream;

downloading the first section and storing it in a memory;

subsequently downloading sections successively transmitted immediately after said first section, each section being successively stored in a respective memory location, until either said last section is downloaded or a predetermined number of sections have been stored in said memory.

In a sixth aspect, the present invention provides a method of storing a group of sections cyclically transmitted in a data stream, said group including a first section and a last section; said method comprising the steps of:

receiving the data stream;

downloading the first section;

subsequently downloading sections successively transmitted immediately after said first section, each section being successively stored in a respective memory location, until either said last section is downloaded or a predetermined number of sections have been stored, whereupon the sections stored in the memory locations are successively overwritten by subsequently downloaded sections until said last section is downloaded; and

storing said last section in said memory.

By virtue of the above third to sixth aspects of the present invention, only those sections required from a group of sections are stored in the memory of the receiver/decoder. For example, when viewing a list of, say, three television programmes scheduled for transmission at a particular time, by means of the third and fifth aspects only the first four sections of the group are downloaded from the data stream, enabling an end user quickly to scroll down the list, starting from the first programme shown at that time. By means of the fourth and sixth aspects only the last four sections of the group are downloaded from the data stream, enabling the end user to scroll quickly up the list, starting from the last programme shown at that time and allowing the memory of the receiver/decoder to be minimised.

In a seventh aspect, the present invention provides apparatus for storing at least one of a plurality of sections of a transmitted data stream, said one section having at least

one section characteristic, said apparatus comprising:

means for receiving the data stream;

means for filtering said one section from said data stream according to at least said one section characteristic; and

means for storing said one section.

Preferably, said filtering means comprises:

means for specifying at least one filtering characteristic, each characteristic comprising a filtering criterion and a value for that filtering criterion;

means for comparing the or each one section characteristic with a respective filtering characteristic; and

means for filtering said one section from said data stream when the or each section characteristic corresponds to the respective filtering characteristic.

In an eighth aspect, the present invention provides apparatus as described above for downloading a plurality of such sections, said storing means comprising a plurality of memory locations for storing respective sections.

Preferably, the apparatus further comprises:

means for specifying at least one data filtering characteristic, each data filtering characteristic comprising a data filtering criterion and a value for that data filtering criterion;

means for comparing data stored in each stored section with a respective data filtering characteristic; and

means for enabling said stored section to be replaced in said memory when said data stored in that stored section does not correspond with the respective data filtering characteristic.

The apparatus may also comprise:

means for supplying a section identification signal to an application when said data stored in said stored section corresponds to said respective data filtering characteristic.

The above method or apparatus may be used with such a data stream in the form of an MPEG bitstream which also includes video and audio television data in addition to said sections.

Preferred features of the present invention will now be described, purely by way of example, with reference to the accompanying drawings, in which:-

Figure 1 shows the overall architecture of a digital television system according to the preferred embodiment of the present invention;

Figure 2 shows the architecture of an interactive system of the digital television system;

Figure 3 shows the arrangement of files within a module downloaded into the memory of an interactive receiver/decoder;

Figure 4 shows the arrangement of a section;

Figure 5 shows the arrangement of memory volumes of the memory of the interactive receiver/decoder;

Figure 6 is a schematic diagram of interfaces of the receiver/decoder;

Figure 7 shows the architecture of the software in the receiver/decoder;

Figure 8 shows the architecture of a receiver system for downloading sections from an MPEG data stream;

Figure 9 shows the arrangement of bytes in a section which are used by a hardware prefilter;

Figure 10 shows an example of a hardware filter mask array and a hardware filter

value array of a hardware filter;

Figure 11 shows the arrangement of bytes in a section which are used by a software prefilter;

Figure 12 shows an example of the loading of a table;

Figure 13 shows an example of an arrangement of sections in a group of sections;

Figure 14 shows an example of a following mode download of a group of sections;

Figure 15 shows an example of a preceding mode download of a group of sections;

Figure 16 shows the arrangement of fields in a descriptor of a group of sections: and

Figure 17 shows the arrangement of fields in a descriptor of a table of sections.

An overview of a digital television system 1000 is shown in Figure 1. The invention includes a mostly conventional digital television system 2000 which uses the known MPEG-2 compression system to transmit compressed digital signals. In more detail, MPEG-2 compressor 2002 in a broadcast centre receives a digital signal stream (typically a stream of video signals). The compressor 2002 is connected to a multiplexer and scrambler 2004 by linkage 2006. The multiplexer 2004 receives a plurality of further input signals, assembles one or more transport streams and transmits compressed digital signals to a transmitter 2008 of the broadcast centre via linkage 2010, which can of course take a wide variety of forms including telecom links. The transmitter 2008 transmits electromagnetic signals via uplink 2012 towards a satellite transponder 2014, where they are electronically processed and broadcast via notional downlink 2016 to earth receiver 2018, conventionally in the form of a dish owned or rented by the end user. The signals received by receiver 2018 are transmitted to an integrated receiver/decoder 2020 owned or rented by the end user and connected to the end user's television set 2022. The receiver/decoder 2020

decodes the compressed MPEG-2 signal into a television signal for the television set 2022.

A conditional access system 3000 is connected to the multiplexer 2004 and the receiver/decoder 2020, and is located partly in the broadcast centre and partly in the decoder. It enables the end user to access digital television broadcasts from one or more broadcast suppliers. A smartcard, capable of deciphering messages relating to commercial offers (that is, one or several television programmes sold by the broadcast supplier), can be inserted into the receiver/decoder 2020. Using the decoder 2020 and smartcard, the end user may purchase commercial offers in either a subscription mode or a pay-per-view mode.

An interactive system 4000, also connected to the multiplexer 2004 and the receiver/decoder 2020 and again located partly in the broadcast centre and partly in the decoder, enables the end user to interact with various applications via a modemmed back channel 4002.

Figure 2 shows the general architecture of the interactive television system 4000 of the digital television system 1000 of the present invention.

For example, the interactive system 4000 allows an end user to buy items from on-screen catalogues, consult local news and weather maps on demand and play games through his television set.

The interactive system 4000 comprises in overview four main elements:

an authoring tool 4004 at the broadcast centre (or elsewhere) for enabling a broadcast supplier to create, develop, debug and test applications;

an application and data server 4006 the broadcast centre, connected to the authoring tool 4004 for enabling a broadcast supplier to prepare, authenticate and format applications and data for delivery to the multiplexer and scrambler 2004 for insertion into the MPEG-2 transport stream (typically the private section thereof) to be broadcast to the end user;

a virtual machine including a run time engine (RTE) 4008, which is an executable code installed in the receiver/decoder 2020 owned or rented by the end user for enabling an end user to receive, authenticate, decompress, and load applications into the working memory 2024 of the receiver/decoder 2020 for execution. The engine 4008 also runs resident, general-purpose applications. The engine 4008 is independent of the hardware and operating system; and

a modemmed back channel 4002 between the receiver/decoder 2020 and the application and data server 4006 to enable signals instructing the server 4006 to insert data and applications into the MPEG-2 transport stream at the request of the end user.

The interactive television system operates using "applications" which control the functions of the receiver/decoder and various devices contained therein. Applications are represented in the engine 4008 as "resource files". A "module" is a set of resource files and data. Several modules may be required to make up an application. A "memory volume" of the receiver/decoder is a storage space for modules. An "interface" is used to download modules. Modules may be downloaded into the receiver/decoder 2020 from the MPEG-2 transport stream.

The elements mentioned in the previous paragraph are now described in more detail.

For the purposes of this specification, an application is a piece of computer code for controlling high level functions of preferably the receiver/decoder 2020. For example, when the end user positions the focus of a remote controller on a button object seen on the screen of the television set 2022 and presses a validation key, the instruction sequence associated with the button is run.

An interactive application proposes menus and executes commands at the request of the end user and provides data related to the purpose of the application. Applications may be either resident applications, that is, stored in the ROM (or FLASH or other non-volatile memory) of the receiver/decoder 2020, or broadcast and downloaded into the RAM or FLASH memory of the receiver/decoder 2020.

Examples of applications are:-

- An Initiating Application. The receiver/decoder 2020 is equipped with a resident initiating application which is an adaptable collection of modules (this term being defined in more detail hereunder) enabling the receiver/decoder 2020 to be immediately operative in the MPEG-2 environment. The application provides core features which can be modified by the broadcast supplier if required. It also provides an interface between the resident application and downloaded applications.
- A Startup Application. The startup application allows any application, either downloaded or resident, to run on the receiver/decoder 2020. This application acts as a bootstrap executed on arrival of a service in order to start the application. Startup is downloaded into RAM and therefore can be updated easily. It can be configured so that the interactive applications available on each channel can be selected and run, either immediately after downloading or after preloading. In the case of preloading, the application is loaded into the memory 2024 and is activated by the startup when required.
- A Program Guide. The Program Guide is an interactive application which gives full information about programming. For example, it may give information about, say, one week's television programmes provided on each channel of a digital television bouquet. By depressing a key on the remote controller 2026, the end user accesses an add-on screen, overlaid on the event shown on the screen of the television set 2022. This add-on screen is a browser giving information on the current and next events of each channel of the digital TV bouquet. By depressing another key on the remote controller 2026, the end user accesses an application which displays a list of information on events over one week. The end user can also search and sort events with simple and customised criteria. The end user can also access directly a selected channel.
- A Pay Per View application. The Pay Per View Application is an interactive service available on each PPV channel of the digital TV bouquet in conjunction with the conditional access system 3000. The end user can access the application using a TV guide or channel browser. Additionally, the

application starts automatically as soon as a PPV event is detected on the PPV channel. The end user is then able to buy the current event either through his daughter smartcard 3020 or via the communication server 3022 (using a modem, a telephone and DTMF codes, MINITEL or the like). The application may be either resident in the ROM of the receiver/decoder 2020 or downloadable into the RAM of the decoder 2020.

- A PC Download application. On request, an end user can download computer software using the PC download application.
- A Magazine Browser application. The magazine browser application comprises a cyclic video broadcast of images with end user navigation via on-screen buttons.
- A Quiz application. The quiz application is preferably synchronised with a broadcast quiz programme. As an example, multiple choice questions are displayed on the screen of the television 2022, and the user can select an answer using the remote controller 2026. The quiz application can inform the user whether the answer is correct or not, and can keep count of the user's score.
- A Teleshopping application. In one example of the teleshopping application, offers of goods for sale are transmitted to the receiver/decoder 2020 and displayed on the television 2022. Using the remote controller, the user can select a particular item to buy. The order for the item is sent via the modemmed back channel 4002 to the application and data server 4006 or to a separate sales system the telephone number of which has been downloaded to the receiver/decoder, possibly with an order to debit the account for a credit card which has been inserted into one of the card readers 4036 of the receiver/decoder 2020.
- A Telebanking application. In one example of the telebanking application, the user inserts a bank card into one of the card readers 4036 of the receiver/decoder 2020. The receiver/decoder 2020 dials up the user's bank, using a telephone number stored in the bank card or stored in the receiver/decoder, and then the application provides a number of facilities which can be selected using the remote controller 2026, for example for downloading

via the telephone line a statement of account, transferring funds between accounts, requesting a cheque book, etc.

- An Internet Browser application. In one example of the Internet browser application, instructions from the user, such as a request to view a web page having a particular URL, are entered using the remote controller 2026, and these are sent by the modemmed back channel 4002 to the application and data server 4006. The appropriate web page is then included in the transmissions from the broadcast centre, received by the receiver/decoder 2020 via the uplink 2012, transponder 2014 and downlink 2016, and displayed on the television 2022.

Applications are stored in memory locations in the receiver/decoder 2020 and represented as resource files. The resource files comprise graphic object description unit files, variables block unit files, instruction sequence files, application files and data files.

The graphic object description unit files describe the screens, the man-machine interface of the application. The variables block unit files describe the data structures handled by the application. The instruction sequence files describe the processing operations of the applications. The application files provide the entry points for the applications.

The applications constituted in this way can use data files, such as icon library files, image files, character font files, colour table files and ASCII text files. An interactive application can also obtain on-line data by effecting inputs and/or outputs.

The engine 4008 only loads into its memory those resource files it needs at a given time. These resource files are read from the graphic object description unit files, instruction sequence files and application files; variables block unit files are stored in memory following a call to a procedure for loading modules and remain locked there until a specific call to a procedure for unloading modules is made.

With reference to Figure 3, a module 4010, such as a tele-shopping module, is a set of resource files and data comprising the following:

a single application file 4012;

an undetermined number of graphic object description unit files 4014;

an undetermined number of variables block unit files 4016;

an undetermined number of instruction sequence files 4018; and

where appropriate, data files 4020 such as icon library files, image files, character font files, colour table files and ASCII text files.

In the MPEG data stream, each module comprises a group of MPEG tables. Each MPEG table may be formatted as a number of sections. In the MPEG data stream, each section has a "size" of up to 4 kbytes. For data transfer via the serial and parallel port, for example, modules similarly are split into tables and sections, the size of the section varying with the transport medium.

Modules are transported in the MPEG data stream in the form of data packets of typically 188 bytes within respective types of data stream, for example, video data streams, audio data streams and teletext data streams. Each packet is preceded by a Packet Identifier (PID) of 13 bits, one PID for every packet transported in the MPEG data stream. A programme map table (PMT table) contains a list of the different data streams and defines the contents of each data stream according to the respective PID. A PID may alert a device to the presence of applications in the data stream, the PID being identified using the PMT table.

With reference to Figure 4, each section 4300 typically comprises the following fields:

table ID (TID) 4302, typically having a size of 1 byte, at the beginning of the section 4300;

section length identifier 4304, typically having a size of 2 bytes;

section number (SN) 4306 (eg. 1) of that section in the table, the SN typically having a size of 1 byte;

the total number of sections (LSN) 4308 (eg. 3) in that table, the LSN typically having a size of 1 byte;

TID extension 4310, typically having a size of 2 bytes;

private data 4312; and

CRC 4314 of the section 4300. The purpose of the CRC 4314 is to check all of the bytes in the section 4300; if the CRC 4314 correlates with all of the preceding data, the section is accepted by the receiver/decoder 2020. Similarly, the private data field 4312 may comprise an MD5 signature at the end of the field and calculated from all of the preceding contents of that field 4312.

In respect of data received via the serial or parallel interfaces, the fields of a section may vary; typically a section received via either of the aforementioned interfaces comprises a reduced amount of data in fields 4302 to 4310 and no CRC 4314.

For a particular module/table, all of the sections making up that table have the same TID 4302 and the same TID extension 4310. For a particular application, all of the tables making up that application have the same TID but different respective TID extensions.

To access a module 4010 from, for example, the MPEG bitstream, both the PID for the module and a module directory are required. This directory simply lists the modules 4010 which can be downloaded from the carrier signal. Once this directory has been downloaded, it is possible for the application to download one or more modules 4010.

The concept of modules 4010 together with the concept of downloading small pieces of code allows the easy evolution of applications. They can be downloaded into permanent FLASH memory of the receiver/decoder 2020 as resident software or broadcast in order to be downloaded into the RAM of the decoder 2020 only when needed by the end user.

A memory volume is a storage space for modules 4010. Such storage spaces are located in the memory 2024 of the receiver/decoder 2020. With reference to Figure 5, the memory 2024 is divided into typically a RAM volume 4022, FLASH volume

4024, and ROM volume 4026. The memory may further be divided into memory volumes associated with the various interfaces through which modules are downloaded into the receiver/decoder 2020, for example an MPEG volume for storing modules downloaded from the MPEG bitstream and a serial volume for storing modules received via a serial interface.

The RAM volume 4022 in turn is divided into a zone dedicated to firmware, a working space for the engine 4008 and the buffers. The FLASH and other non-volatile memory can be accessed either by an application or the engine itself through a device manager.

Each volume contains a list of modules 4010, each module 4010 containing a list of files 4012, 4014, 4016, 4018, 4020. It is possible to have two files bearing the same name and which may be located in distinct modules. For example, a version of the application is typically stored in the ROM volume 4026, with later versions being downloadable into the FLASH volume 4024 to substitute the version stored in the ROM volume with that volume stored in the FLASH volume 4024. The contents of files may be compressed in LZW format, however as decompression of files takes a certain period of time they may be received in decompressed format.

Physical interfaces of the receiver/decoder 2020 are used for downloading data. With reference to Figure 6, the receiver/decoder 2020 contains, for example, six downloading media; MPEG flow tuner 4028, serial interface 4030, parallel interface 4032, modem 4034 and two card readers 4036.

With multiple sources of applications and multiple manufacturing sources of receiver/decoder 2020, it is important that one application behaves in the same way on every receiver/decoder, and each receiver/decoder should execute every application in the same, correct manner. With reference to Figure 7, the receiver/decoder 2020 comprises a run time engine 4008 running under the control of a microprocessor and a common application programming interface 4054. They are installed in every receiver/decoder 2020 so that all receiver/decoders 2020 are identical from the

application point of view.

Figure 7 shows the architecture of the receiver/decoder 2020 for running applications 4056. The virtual machine 4007 executes applications 4056, which may comprise applications 4056' coupled directly to the virtual machine or applications 4056" downloaded to the receiver/decoder 2020 from, for example, the MPEG data stream. The run time engine 4008 also displays graphics and text, calls devices for services, receives "events" and uses functions of a library 4058 for specific computation.

With reference to Figure 7, with respect to an application a function of the decoder 2020 is "seen" as a device 4060. There may, therefore, be functions of the receiver/decoder 2020 which may not be seen by any application.

A device 4060 comprises a logical device unit which may correspond to a component 4062 or physical interface 4064 of the hardware 4066. Such devices are referred to as "low level devices" 4068. The output of such a device 4068 may be connected to at least one device driver 4070 for converting the logical signals output by the device 4068 into signals required to drive, for example, a hardware interface 4064. Alternatively, the device 4068 may itself drive a component or interface of the receiver/decoder 2020, that is, the output of the device may be connected directly to the hardware 4066.

Examples of low level devices 4068 are described below.

An LCARD device enables a program to communicate with the smartcard contained in one smartcard reader 4036, and an RCARD device enables a program to communicate with the smartcard contained in the other smartcard reader 4036. For example, these devices enable a program to read the state of the card, read the card history and send an input message to the card. The devices also inform a program of the insertion of a card in to the reader, removal of a card from a reader and card reset if not requested by the program. The LCARD and RCARD devices are specific to the protocol used for running the card. Typically an ISO7816 protocol is used.

An SCTV device enables a program to verify and configure of a scart outlet to the television set 2022. For example, this device enables a program to request information about the sound characteristic of the scart outlet, perform a "MUTE" on the sound and dynamically program RGB levels.

A TUNER device enables a program to use the tuner 4028. For example, the device enables a program to perform a scan from either a minimum frequency or a current frequency of the tuner, read the tuner parameters and program the tuner.

A SERIAL device enables a program to communicate with equipment via a serial link and a PARALLEL device enables a program to communicate with equipment via a parallel link. For example, these devices enable a program to send a message via the respective link and inform a program of the reception of a message via that link.

A MODEM device allows the receiver decoder to communicate with a data service via an internal half-duplex modem supporting V23. The MODEM device requests the dialling of a number, the sending of a message to the data server and disconnection of the modem, and signals reception of a message, the detection of errors and the loss or detection of a carrier.

Remote devices, executed in a remote location, can be any of the local devices, except that a port and protocol must be defined.

In addition to "low level devices" the receiver/decoder 2020 may also include "high level devices" 4072 which control operations of the receiver/decoder 2020.

With respect to sections loaded from an MPEG data stream, a device 4072 referred to as an "MLOAD device" allows an application to load an MPEG section, a complete MPEG table or a group of MPEG sections corresponding to hardware and software filtering criteria.

With reference to Figure 8, electromagnetic signals are received by receiver 2018 and transmitted to MPEG tuner 4028. The tuner typically scans a range of frequencies, stopping only when carrier frequency is detected within that range. The thus detected signals are transmitted to demodulator 4500, which demodulates the signals and transmits them to demultiplexer 4502. Demultiplexer 4502 is connected to MPEG chip 4504, which in turn is connected to television 2022. The demultiplexer 4502 is also connected to hardware filter 4506 which typically provides up to 32 outputs connected to RAM volume 4022 of the receiver/decoder 2020.

The electromagnetic signals received by the demultiplexer 4502 comprise a number of data streams of data packets, typically comprising a stream of video packets, a stream of audio packets, a stream of teletext packets and a stream of subtitle packets for a service provided by the broadcast supplier. These data packets are demultiplexed by the demultiplexer 4502 and transmitted to MPEG chip 4504 for decoding into signals for the television set 2022.

The data packets relating to applications are transmitted to the hardware filter 4506. With reference to Figure 9, the hardware filter, or prefilter, 4506 utilises one or more of typically 8 bytes of a section 4508 of the data stream, typically offset byte 0 (4510) and offset bytes 3 to 9 (4512) to filter sections from the data stream. These bytes include the TID 4302, section number 4306, last section number 4308 and TID extension 4310 of the MPEG section 4508.

To define the hardware prefilter characteristic, the application specifies which bits of the section 4508 to take into account and the expected value of each selected bit. Accordingly, the hardware prefilter 4506 comprises a filter mask array 4514 and a filter value array 4516 respectively, an example of each being shown in Figure 10. In Figure 10, bits 4518, 4520 and 4522 are chosen using filter mask 4514 and the value of each of these bits, 0, 1 and 0 respectively, is chosen using filter value 4516. With those filtering criteria, MPEG section 4524 would be output to the buffers of the RAM 4022 of the receiver/decoder 2020, whereas MPEG section 4526 would not be output to the buffers.

The hardware prefilter 4506 operates by means of a demultiplexer chip, avoiding wasting processor power within the main processor.

The microprocessor may utilise the RAM 4022 of the receiver/decoder 2020 in which the sections are stored as a software filter 4526. By means of the software filter 4526, it is possible to select which of the sections loaded in the buffers of the RAM volume 4022 will be brought to the application.

The principles of software filtering are similar to those of hardware filtering. The software filter 4526 uses typically 8 consecutive bytes 4528 of data in the MPEG section 4508. As shown in Figure 11, the position of the first byte 4530 of these 8 consecutive bytes is defined with an offset 4532 based on the first byte 4510 in the section 4508.

If after loading via the hardware filter the data stored in an MPEG section does not correspond to the criteria for software filtering, the buffer in which the MPEG section is stored may be used to load another MPEG section.

To load an MPEG section from the MPEG data stream, the MLOAD device is supplied typically with the following information:

the PID of the data stream in which the MPEG section is to be found;

a descrambling data array comprising a number of PID ECMs for descrambling PIDs;

an operator reference, which recognises a particular ECM to be used to descramble the PID;

the hardware filter value array to be applied;

the hardware filter mask array to be applied;

the software filter offset value to be applied;

the software filter value array to be applied;

the software filter mask array to be applied;

the maximum size of the section (up to 4096 bytes); and

a time out function determining how long the device should wait to load a

section, a group of sections or a table.

When it is indicated that the maximum size of a section is 512 bytes, the size of the section cannot exceed 512 bytes.

The section is loaded all at once into the buffer of the RAM volume 4022 allocated by the device. When the section has been loaded, and if it corresponds to the software filtering criteria, the device should free the demultiplexing route and inform the application via an "event", which includes an execution report and the address of the buffer containing the section. After loading, a section is identified by the address of the buffer in which it is stored.

When a table is to be loaded, the device is also supplied with the TID 4302. Each section of the table is loaded into the respective buffer allocated therefor all at once. However, it is not essential that the tables are loaded by putting sections end to end and in section order. As each section of the table is loaded, the device may inform the application that that section of the table has been loaded. Alternatively or additionally the device may inform the application via an event that the entire table has been loaded successfully into the buffers allocated therefor.

With reference to Figure 12, a table is typically loaded in the following way (description of the software filtering is omitted for reasons of clarity only);

step 1: specify the hardware filter mask array 4534 and the hardware filter value array 4536;

step 2: load a first section 4538 corresponding to the hardware (and software) filter;

step 3: read the value of the LSN 4540 of the section 4538 to determine the number of sections to be loaded (= LSN + 1) and change the hardware filter mask array 4534 and the hardware filter value array 4536 according to the value of the LSN 4540;

steps 4 to 6: load the remaining sections 4542, 4544 and 4546 according to the values of the TID and the LSN and send an event to the application.

Table loading is only complete when an event indicating successful downloading is entered into the queue of the run time engine 4006. Before this event is placed in the engine queue, abortion of the loading of the table should be successful.

After a table has been loaded, it is identified with a name assigned to it by the application when requesting the downloading of a table.

Information regarding the way in which a table is stored in the memory of the receiver/decoder 2020 can be accessed via a call which returns a descriptor of the table. An example of such a descriptor is shown in Figure 17. The descriptor 4700 includes a field 4702 comprising the total number of sections (LSN + 1) in the table and a list 4704 of the buffers containing the sections 4538, 4542, 4544 and 4546 which make up the table commencing with the section 4544 having the SN of "0", then the section 4546 having the SN of "1" and so on up to the section 4542 having the SN corresponding to the LSN.

A group of sections comprises all of the sections which correspond to the hardware and software filtering criteria. The sections appear cyclically in the MPEG data stream.

With reference to Figure 13, a group 4548 is delimited by two sections; a first section (SDG) 4550 and a last section (SFG) 4552, each of which corresponds to respective filter criteria, with the other sections (SG) 4554 of the group therebetween.

The loading of sections in a group may occur using two different loading modes.

In the "following mode", a predetermined number of sections 4554 which immediately follow the SDG 4550 are loaded. For each type of section (SDG, SG, SFG), the application identifies the TID, hardware filter, software filter and the maximum section length.

With reference to Figure 14, the device preferably first applies hardware and software

filtering criteria to enable the downloading and storing of the SDG 4550 only This avoids the loading of redundant SGs 4554 in buffers of the RAM memory volume 4022 of the receiver/decoder 2020 prior to the loading of the SDG 4550. Upon detection, SDG 4550 is stored in a memory location or buffer 4556. The device then applies hardware and software filtering criteria to enable the downloading of the desired SGs and SFG from the group of MPEG data sections.

In the example shown in Figure 14, three SGs 4554 from a group of MPEG sections comprising one SDG, seven SGs and one SFG are to be downloaded from the MPEG data stream. Following storage of the SDG 4550 in buffer 4556, the first SG 4558 from the group of MPEG sections to be transmitted after the SDG 4550 is stored in buffer 4560. Likewise, the second SG 4562 from the group of MPEG sections to be transmitted after the SDG 4550 is stored in buffer 4564 and the third SG 4566 from the group of MPEG sections to be transmitted after the SDG 4550 is stored in buffer 4568.

Storage of the SGs continues until either the required number of sections 4554 have been stored or the SFG 4552 is downloaded. If downloaded, the SFG 4552 is not stored in a memory location.

As each section of the group is loaded, the device may inform the application via an event that that section of the group has been loaded. Alternatively, or additionally, the device may inform the application of only either the completion of the storage of the MPEG sections or the downloading of the SFG 4552 by sending an event to the application.

In the "preceding mode", a predetermined number of sections which immediately precede the SFG are stored in the memory locations. For each type of section (SDG, SG, SFG), the application identifies the PID, TID, hardware filter, software filter and the maximum section length.

Loading in this mode requires the use of a circular buffer to store MPEG sections.

With reference to Figure 15, the device preferably first applies hardware and software filtering criteria to enable the downloading and storing of the SDG 4570 only. This avoids the loading of redundant SGs 4544 in the RAM memory volume 4022 of the receiver/decoder 2020 prior to the loading of the SDG 4570. Upon detection, SDG 4550 is not stored. The device then applies hardware and software filtering criteria to enable the downloading of the desired SGs and SFG from the group of MPEG data sections.

In the example shown in Figure 15, three SGs 4544 from a group of MPEG sections comprising one SDG, seven SGs and one SFG are to be downloaded from the MPEG data stream. Following the downloading of the SDG 4570, the first SG 4572 from the group of MPEG sections to be transmitted after the SDG 4570 is stored in buffer 4574. Likewise, the second SG 4576 from the group of MPEG sections to be transmitted after the SDG 4550 is stored in buffer 4578 and the third SG 4580 from the group of MPEG sections to be transmitted after the SDG 4570 is stored in buffer 4582.

The fourth SG 4584 from the group of MPEG sections to be transmitted after the SDG 4550 is stored in buffer 4574, that is, the buffer in which the first SG 4572 had been stored. Similarly, each of the subsequently transmitted SGs 4586, 4588 and 4590 are stored in buffers 4578, 4582 and 4574 respectively.

Storage of SGs in this cyclic manner is continued until the SFG 4592 is detected by the device. This is the case even if the desired number of SGs have not been loaded before the SFG is detected. Upon detection, the SFG 4592 is stored in a buffer, and an event indicating successful downloading of the group is sent to the application.

Information regarding the way in which a group is stored in the RAM 4022 of the receiver/decoder 2020 can be accessed via a special call which returns a descriptor of the group. An example of such a descriptor is shown in Figure 16. The descriptor 4600 includes a field 4602 comprising the number of sections in the group and a list 4604 of the buffers containing either the SDG or the SFG (depending on the loading

mode) and the sections SG which make up the group in the order of loading.

After the group has been loaded, it is uniquely identified with a name assigned to it by the application when requesting the downloading of the group.

It is possible to add a section to a group or to delete a section from a group. A section may belong to more than one group.

When the application requests the addition of a section to a group, the address of the buffer containing the new section is put at the end of the descriptor 4604, and the field 4602 is updated. When the application requests the deletion of a section from a group, the address of the buffer containing this section is suppressed in the descriptor 4600, and field 4602 is updated. If the section was not the last section in the group, the descriptor is reorganised to remove any empty memory spaces. Thus, it is possible to have an "empty group" by deleting all of the sections.

Devices are identified with a unique identifier, for example, "LCARD_DEVICE_ID" identifies the LCARD device and "RCARD_DEVICE_ID" identifies the RCARD device.

When a new device 4062 is created, it can be installed in existing decoders 2020 by downloading the relevant application 4056 from the broadcast centre. Typically the maximum number of devices handled by one manager is 256.

This downloading is performed in the receiver decoder 2020 by an application 4056 which checks the hardware and software versions and, if correct, loads the software module representing the new device 4062 and asks a procedure of the toolbox 4058 to install the new device code within the firmware (in FLASH memory). This can provide a flexible and secure installation of new functions within the decoder 2020 without affecting the rest of the software.

It will be understood that the present invention has been described above purely by

way of example, and modifications of detail can be made within the scope of the invention.

Each feature disclosed in the description, and (where appropriate) the claims and drawings may be provided independently or in any appropriate combination.

In the aforementioned preferred embodiments, certain features of the present invention have been implemented using computer software. However, it will of course be clear to the skilled man that any of these features may be implemented using hardware. Furthermore, it will be readily understood that the functions performed by the hardware, the computer software, and such like are performed on or using electrical and like signals.

Cross reference is made to our co-pending applications, all bearing the same filing date, and entitled Signal Generation and Broadcasting (Attorney Reference no. PC/ASB/19707), Smartcard for use with a Receiver of Encrypted Broadcast Signals, and Receiver (Attorney Reference No. PC/ASB/19708), Broadcast and Reception System and Conditional Access System therefor (Attorney Reference No. PC/ASB/19710), Downloading a Computer File from a Transmitter via a Receiver/Decoder to a Computer (Attorney Reference No. PC/ASB/19711), Transmission and Reception of Television Programmes and Other Data (Attorney Reference No. PC/ASB/19712), Downloading Data (Attorney Reference No. PC/ASB/19713), Computer Memory Organisation (Attorney Reference No. PC/ASB/19714), Television or Radio Control System Development (Attorney Reference No. PC/ASB/19715), Extracting Data Sections from a Transmitted Data Stream (Attorney Reference No. PC/ASB/19716), Access Control System (Attorney Reference No. PC/ASB/19717), Data Processing System (Attorney Reference No. PC/ASB/19718), and Broadcast and Reception System, and Receiver/Decoder and Remote Controller therefor (Attorney Reference No. PC/ASB/19720). The disclosures of these documents are incorporated herein by reference. The list of applications includes the present application.

THIS SHEET HAS BEEN REPLACED

This Sheet Has Been Replaced



CLAIMS

1. A method of storing at least one of a plurality of sections of a transmitted data stream, said one section having at least one section characteristic, said method comprising the steps of:

receiving the data stream;

filtering said one section from said data stream according to at least said one section characteristic; and

storing said one section.

2. A method according to Claim 1, wherein said filtering step comprises the steps of:

specifying at least one filtering characteristic, each characteristic comprising a filtering criterion and a value for that filtering criterion;

comparing the or each section characteristic with a respective filtering characteristic; and

filtering said one section from said data stream when the or each section characteristic corresponds to the respective filtering characteristic.

3. A method of storing a plurality of sections of a transmitted data stream using the method according to Claim 1 or Claim 2, each section being stored in a respective memory location.

4. A method according to Claim 3, said method further comprising the step of:

specifying at least one data filtering characteristic, each data filtering characteristic comprising a data filtering criterion and a value for that data filtering criterion;

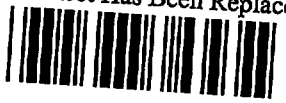
comparing data stored in each stored section with a respective data filtering characteristic; and

enabling said stored section to be replaced in said memory when said data stored in that stored section does not correspond with the respective data filtering



THIS SHEET HAS BEEN REPLACED

This Sheet Has Been Replaced



characteristic.

5. A method according to Claim 4, further comprising the step of:

supplying a section identification signal to an application when said data stored in said stored section corresponds to said respective data filtering characteristic.

6. A method according to Claim 5, wherein said section identification signal comprises the storage address of said one section.

7. A method of storing a group of sections according to the method of any of Claims 3 to 6, said sections being cyclically transmitted in said data stream, each group including a first section and a last section; said method comprising the steps of:

storing the first section;

subsequently downloading sections successively transmitted immediately after said first section, each section being successively stored in a respective memory location, until either said last section is downloaded or a predetermined number of sections have been stored.

8. A method of storing a group of sections according to the method of any of Claims 3 to 6, said sections being cyclically transmitted in said data stream, said group including a first section and a last section; said method comprising the steps of:

downloading the first section;

subsequently downloading sections successively transmitted immediately after said first section, each section being successively stored in a respective memory location, until either said last section is downloaded or a predetermined number of sections have been stored whereupon the sections stored in the memory locations are successively overwritten by subsequently downloaded sections until said last section is downloaded; and

storing said last section in said memory.

9. A method of storing a group of sections cyclically transmitted in a data stream, said group including a first section and a last section; said method comprising the



THIS SHEET HAS BEEN REPLACED

This Sheet Has Been Replaced



steps of:

receiving the data stream;

downloading the first section and storing it in a memory;

subsequently downloading sections successively transmitted immediately after said first section, each section being successively stored in a respective memory location, until either said last section is downloaded or a predetermined number of sections have been stored in said memory.

10. A method of storing a group of sections cyclically transmitted in a data stream, said group including a first section and a last section; said method comprising the steps of:

receiving a data stream comprising a plurality of sections;

initially downloading the first section;

subsequently downloading sections successively transmitted immediately after said first section, each section being successively stored in a respective memory location, until either said last section is downloaded or a predetermined number of sections have been stored, whereupon the sections stored in the memory locations are successively overwritten by subsequently downloaded sections until said last section is downloaded; and

storing said last section in said memory.

11. A method of storing at least one of a plurality of sections of a transmitted data stream substantially as herein described with reference to the accompanying drawings.

12. Apparatus for storing at least one of a plurality of sections of a transmitted data stream, said one section having at least one section characteristic, said apparatus comprising:

means for receiving the data stream;

means for filtering said one section from said data stream according to at least said one section characteristic; and

means for storing said one section.



THIS SHEET HAS BEEN REPLACED

13. Apparatus according to Claim 12, wherein said filtering means comprises:

means for specifying at least one filtering characteristic, each characteristic comprising a filtering criterion and a value for that filtering criterion;

means for comparing the or each one section characteristic with a respective filtering characteristic; and

means for filtering said one section from said data stream when the or each section characteristic corresponds to the respective filtering characteristic.

14. Apparatus according to Claim 12 or 13 for downloading a plurality of such sections, said storing means comprising a plurality of memory locations for storing respective sections.

15. Apparatus according to Claim 14, further comprising:

means for specifying at least one data filtering characteristic, each data filtering characteristic comprising a data filtering criterion and a value for that data filtering criterion;

means for comparing data stored in each stored section with a respective data filtering characteristic; and

means for enabling said stored section to be replaced in said memory when said data stored in that stored section does not correspond with the respective data filtering characteristic.

16. Apparatus according to Claim 15, further comprising:

means for supplying a section identification signal to an application when said data stored in said stored section corresponds to said respective data filtering characteristic.

17. Apparatus according to Claim 16, wherein said section identification signal comprises the storage address of said one section.

18. A method or apparatus according to any preceding claim, for use with such a

This Sheet Has Been Replaced

* C L M S *

THIS SHEET HAS BEEN REPLACED

data stream in the form of an MPEG bitstream containing video and audio television data in addition to said sections.

19. Apparatus for storing at least one of a plurality of sections of a transmitted data stream substantially as herein described with reference to and as shown in the accompanying drawings.

This Sheet Has Been Replaced



