



- (51) **International Patent Classification:**
G06F 5/00 (2006.01)
- (21) **International Application Number:**
PCT/AU2015/050010
- (22) **International Filing Date:**
13 January 2015 (13.01.2015)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (30) **Priority Data:**
2014900093 13 January 2014 (13.01.2014) AU
- (72) **Inventor; and**
- (71) **Applicant : GUAN, Jeffery** [AU/AU]; 90 Church Lane, Cranebrook, New South Wales 2749 (AU).
- (74) **Agent: DAVIES COLLISON CAVE;** Level 14, 255 Elizabeth Street, Sydney, New South Wales 2000 (AU).
- (81) **Designated States** (*unless otherwise indicated, for every kind of national protection available*): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY,

BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

- (84) **Designated States** (*unless otherwise indicated, for every kind of regional protection available*): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— with international search report (Art. 21(3))

(54) **Title:** AN INTERFACE DEVICE, SERVER PROCESSING SYSTEM, METHOD, PROCESSING SYSTEM, MOBILE COMMUNICATION DEVICE AND COMPUTER PROGRAM

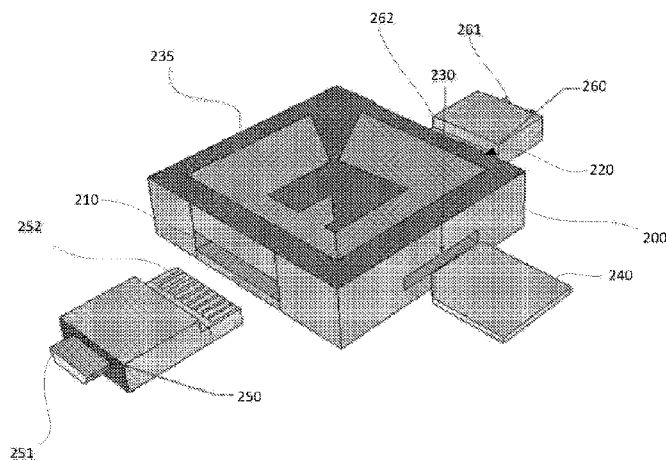


FIGURE 2

(57) **Abstract:** An interface device (200) to interface a first device (400) to a second device (500). The interface device includes a first part (200) and a second part (220). The interface device (200) further includes a third part (230) to which a memory or processor is connected. In use, the processor is configured to receive first device type data, second device type data, to thereby determine a configuration file for configuring the interface device (400) such that the first device (400) can be interfaced to the second device (500).



- 1 -

**AN INTERFACE DEVICE, SERVER PROCESSING SYSTEM, METHOD,
PROCESSING SYSTEM, MOBILE COMMUNICATION DEVICE AND
COMPUTER PROGRAM**

5 Field of Invention

The present invention relates to an interface device, server processing system, method, processing system, mobile communication device, and computer program for enabling a first device to interface with a second device.

10 Background of the Invention

The reference in this specification to any prior publication (or information derived from it), or to any matter which is known, is not, and should not be taken as, an acknowledgement or admission or any form of suggestion that prior publication (or information derived from it) or known matter forms part of the common general knowledge in the field of endeavour

15 to which this specification relates.

More and more devices are becoming computerised. However, it is a common problem that one device may not be able to interface with another component due to different protocols or other technological barriers. For example, two device may transfer or receive
20 data using different protocols. In order to progress toward a pervasive computing environment, there is a need to enable devices that cannot typically communicate, to be able to communicate together.

The present invention seeks to overcome the deficiencies of the prior art in providing an
25 interface device to enable a first device to interface with a second device.

The present invention also seeks to provide a processing system to enable an interface device to be configured such that a first device can be thereby interfaced with a second device.

30

- 2 -

Summary of Invention

In a first aspect there is provided a server processing system for generating a configuration file for configuring an interface device to enable a first device to interface with a second device, wherein the server processing system is configured to:

- 5 receive first device type data indicative of a first device type;
 receive second device type data indicative of a second device type;
 receive data manipulation data indicative of data manipulation to be performed to
first data received from the first device to generate second data for transfer to the second
device; and
- 10 determine, using the first device type data, the second device type data, and the data
manipulation data, a configuration file for configuring the interface device.

In certain embodiments, the server processing system generates the configuration file using the first device type data, the second device type data, and the data manipulation data.

- 15 In certain embodiments, the first device type data, the second device type data and the data
manipulation data are received via a processing system, and wherein the server processing
system transfers the configuration file to the processing system which facilitates in the
loading of the configuration file to the interface device.

- 20 In certain embodiments, the server processing system has access to a repository including
a plurality of device drivers/IP cores, wherein the server processing system is configured to
retrieve one or more of the device drivers/IP cores for generating the configuration file.

- 25 In certain embodiments, the server processing system utilises a digital rights management
system for controlling use of one or more of the device drivers/IP cores in the repository
for generating the configuration file.

- 30 In certain embodiments, the first device type data is indicative of a selected first device
type from a plurality of first device types available in the repository, wherein the server

- 3 -

processing system retrieves, from the plurality of device drivers/IP cores in the repository, an input interface core corresponding to the first device type for generating the configuration file.

- 5 In certain embodiments, the second device type data is indicative of a selected second device type from a plurality of second device types available in the repository, wherein the server processing system retrieves, from the plurality of device drivers/IP cores in the repository, an output interface core corresponding to the second device type for generating the configuration file.

10

In certain embodiments, the data manipulation data is indicative of a selected data manipulation from a plurality of data manipulations available in the repository, wherein the server processing system retrieves, from the plurality of device drivers/IP cores in the repository, a data manipulation core corresponding to the data manipulation required for
15 generating the configuration file.

In certain embodiments, the data manipulation data is indicative of code for performing the data manipulation, wherein the server processing system uses the code for generating the configuration file.

20

In certain embodiments, the server processing system uses the code for generating an executable file, wherein the configuration file and executable file are loaded to the interface device to enable the first device to interface with the second device.

- 25 In a second aspect there is provided a method for generating a configuration file for configuring an interface device to enable a first device to interface with a second device, wherein the method includes, in a server processing system, steps of:

receiving first device type data indicative of a first device type;
receiving second device type data indicative of a second device type;

- 4 -

receiving data manipulation data indicative of data manipulation to be performed to first data received from the first device to generate second data for transfer to the second device; and

generating, using the first device type data, the second device type data, and the
5 data manipulation data, a configuration file for configuring the interface device.

In certain embodiments, the method includes the server processing system generating the configuration file using the first device type data, the second device type data, and the data manipulation data.

10

In certain embodiments, method includes the server processing system receiving the first device type data, the second device type data and the data manipulation data from a processing system, and wherein the method includes the server processing system transferring the configuration file to the processing system which facilitates in the loading
15 of the configuration file to the interface device.

In certain embodiments, the server processing system has access to a repository including a plurality of device drivers/IP cores, wherein the method includes the server processing system retrieving one or more of the device drivers/IP cores for generating the
20 configuration file.

In certain embodiments, the method includes the server processing system utilising a digital rights management system for controlling use of one or more of the device drivers/IP cores in the repository for generating the configuration file.

25

In certain embodiments, the first device type data is indicative of a selected first device type from a plurality of first device types available in the repository, wherein the method includes the server processing system retrieving, from the plurality of device drivers/IP cores in the repository, an input interface core corresponding to the first device type for
30 generating the configuration file.

- 5 -

In certain embodiments, the second device type data is indicative of a selected second device type from a plurality of second device types available in the repository, wherein the method includes the server processing system retrieving, from the plurality of device
5 drivers/IP cores in the repository, an output interface core corresponding to the second device type for generating the configuration file.

In certain embodiments, the data manipulation data is indicative of a selected data manipulation from a plurality of data manipulations available in the core repository,
10 wherein the method includes the server processing system retrieving, from the plurality of IP cores in the repository, a data manipulation core corresponding to the data manipulation required for generating the configuration file.

In certain embodiments, the data manipulation data is indicative of code for performing the
15 data manipulation, wherein the method includes the server processing system using the code for generating the configuration file.

In certain embodiments, the method includes the server processing system using the code for generating an executable file, wherein the configuration file and executable file are
20 loaded to the interface device to enable the first device to interface with the second device.

In a third aspect there is provided a computer program stored on a computer readable medium, the computer program including executable instructions which configure a server processing system to perform the method of the second aspect.
25

In a fourth aspect there is provided a processing system for interacting with a server processing system for requesting a configuration file for configuring an interface device, wherein the processing system is configured to:

transfer, to the server processing system, first device type data indicative of a first
30 device type;

- 6 -

transfer, to the server processing system, second device type data indicative of a second device type;

transfer, to the server processing system, data manipulation data indicative of data manipulation to be performed to first data received from the first device to generate second
5 data for transfer to the second device; and

receive, from the server processing system, a configuration file for configuring the interface device, wherein the configuration file is generated based upon the first device type data, the second device type data, and the data manipulation data;

transfer the configuration file to memory of the interface device, wherein loading of
10 the configuration file configures the interface device to enable a first device connected to the interface device to interface with a second device connected to the interface device.

In certain embodiments, the processing system is a mobile communication device operable to execute a mobile application for transferring the first device type data, the second device
15 type data and the data manipulation data to the server processing system.

In certain embodiments, the first device type data is indicative of a selected first device type from a plurality of first device types for use by the server processing system to determine an input interface core for generating the configuration file.
20

In certain embodiments, the second device type data is indicative of a selected second device type from a plurality of second device types for use by the server processing system to determine an output interface core for generating the configuration file.

25 In certain embodiments, the data manipulation data is indicative of a selected data manipulation from a plurality of data manipulations available from the server processing system for generating the configuration file.

In certain embodiments, the data manipulation data is indicative of code for performing the
30 data manipulation, wherein the code is input via the mobile application, wherein the

- 7 -

programming code is transferred to the server processing system for generating the configuration file.

5 In certain embodiments, the processing system provides user identification data to the server processing system for use in a digital rights management system for generating the configuration file.

In a fifth aspect there is provided a method for interacting with a server processing system for requesting a configuration file for configuring an interface device, wherein method
10 includes, in a processing system, steps of:

transferring, to the server processing system, first device type data indicative of a first device type;

transferring, to the server processing system, second device type data indicative of a second device type;

15 transferring, to the server processing system, data manipulation data indicative of data manipulation to be performed to first data received from the first device to generate second data for transfer to the second device; and

receiving, from the server processing system, a configuration file for configuring the interface device, wherein the configuration file is generated based upon the first device
20 type data, the second device type data, and the data manipulation data;

transferring the configuration file to memory of the interface device, wherein loading of the configuration file configures the interface device to enable a first device connected to the interface device to interface with a second device connected to the interface device.

25

In certain embodiments, the processing system is a mobile communication device, wherein the method includes the mobile communication device executing a mobile application for transferring the first device type data, the second device type data and the data manipulation data to the server processing system.

30

- 8 -

In certain embodiments, the mobile communicating device is adapted to wirelessly transfer at least some of said data, such as, but not limited to wirelessly transferring the configuration file to the interface device.

- 5 In certain embodiments, the first device type data is indicative of a selected first device type from a plurality of first device types for use by the server processing system to determine an input interface core for generating the configuration file.

- 10 In certain embodiments, the second device type data is indicative of a selected second device type from a plurality of second device types for use by the server processing system to determine an output interface core for generating the configuration file.

- 15 In certain embodiments, the data manipulation data is indicative of a selected data manipulation from a plurality of data manipulations available from the server processing system for generating the configuration file.

- 20 In certain embodiments, the data manipulation data is indicative of code for performing the data manipulation, wherein the method includes inputting the code via the mobile application, wherein the programming code is transferred to the server processing system for generating the configuration file.

- In certain embodiments, the method includes the processing system providing user identification data to the server processing system for use in a digital rights management system for generating the configuration file.

25

In a sixth aspect there is provided a computer program stored on a computer readable medium, the computer program including executable instructions which configure a processing system to perform the method of the fifth aspect.

- 9 -

In a seventh aspect there is provided an interface device for enabling a first device to interface with a second device, wherein the device includes:

- a first port for receiving an electrical connection for the first device;
- a second port for receiving an electrical connection for the second device;
- 5 a third port for connecting the interface device to a memory device or processing system;
- a programmable device in electrical connection with the first port and the second port;
- a memory for storing a configuration file received from the memory device or
- 10 processing system via the third port, wherein execution of the configuration file configures the programmable device to enable the first device to interface with the second device.

In an eighth aspect there is provided an interface device for enabling a first device to interface with a second device, wherein the device includes:

- 15 a first port:
 - for receiving an electrical connection for the first device; and
 - for connecting the interface device to a memory device or processing system;
- a second port for receiving an electrical connection for the second device;
- 20 a switch for selectively operating the interface device in a configuration mode and an interface mode; and
- a programmable device in electrical connection with the first port and the second port;
- a memory for storing a configuration file received from the memory device or
- 25 processing system via the first port when the interface device is operating in the configuration mode, wherein execution of the configuration file configures the programmable device to enable the first device to interface with the second device when the interface device operates in the interface mode.

- 10 -

In certain embodiments, the programmable device is configurable to perform data manipulation upon first data received from the first device to generate second data which is transferred to the second device.

- 5 In certain embodiments, the interface device includes a processor in electrical connection with the programmable device, wherein the memory stores an executable file for execution by the processor to enable the first device to interface with the second device.

- In certain embodiments, the processor is configured, upon execution of the executable file,
10 to perform data manipulation upon the first data received from the first device to generate second data which is transferred to the second device.

- In certain embodiments, the first data is in accordance with a first protocol and the second data is in accordance with a second protocol, wherein the first and second protocols are
15 different.

- In certain embodiments, the programmable device may include a Field Programmable Gate Array, a System on Chip (SoC) device and/or any other like device which may preferably be programmed by a user (preferably in the field after manufacture of the device) and
20 which may preferably be provided in an integrated circuit (IC) or like form, integrating the various components of a computer system into a single chip/embedded form.

- In certain embodiments, the interface device further includes at least one further port for receiving an electrical connection for a further device which is to be interfaced with said
25 first and second devices, and wherein, said programmable device is provided in electrical connection with each of said further ports, and wherein execution of the configuration file configures the programmable device to enable each of said devices to be interfaced together.

- 30 In certain embodiments, each said device port is connected to its respective port either by

- 11 -

wired or wireless connection.

Brief Description of the Figures

Example embodiments should become apparent from the following description, which is
5 given by way of example only, of at least one preferred but non-limiting embodiment,
described in connection with the accompanying figures.

Figure 1 is a flow chart representing a method of configuring an interface device to enable
a first device to interface with a second device;

10

Figure 2 is an isometric perspective of an example of the interface device;

Figure 3 is a plan view of the interface device of Figure 2 connected to a first device in the
form of a mobile communication device and a second device in the form of an ethernet
15 network cable;

Figure 4 is a flowchart illustrating a method of configuring the interface device to enable a
first device to interface with a second device;

20 Figure 5 is a block diagram of an example of the interface device;

Figure 6 is an example of a mobile communication device presenting an interface of a
mobile device application for generating a configuration file for configuring the interface
device;

25

Figure 7 is an example of first architecture framework of an FPGA of the interface device;
and

Figure 8 is an example of a more detailed architecture framework of an FPGA interfacing
30 with a configuration system; and,

- 12 -

Figure 9 is an example of the interface device for coupling to the mobile communication device for configuration.

5 Detailed Description of Embodiments

The following modes, given by way of example only, are described in order to provide a more precise understanding of the subject matter of a preferred embodiment or embodiments.

- 10 In the figures, incorporated to illustrate features of an example embodiment, like reference numerals are used to identify like parts throughout the figures.

Referring to Figure 2 there is shown a schematic of an example of an interface device 200 to enable a first device 400 (see Figure 4) to interface with a second device 500 (see Figure 15 4). Figure 6 also provides a functional block diagram of the interface device 200. In particular, the interface device 200 includes a first port 210, a second port 220, at least one third port 230, 235, and a programmable device 620 in electrical connection with the first and second port 210, 220 and a memory 630.

- 20 The programmable device 620 may, for example, include a Field Programmable Gate Array, as described in the following embodiment, a System on Chip (SoC) device or any other like device which may typically be programmed by a user (preferably in the field after manufacture of the device) and which may preferably be provided in an integrated circuit (IC) or like form, integrating the various components of a computer system into a 25 single chip form.

More specifically, the first port 210 is configured for receiving an electrical connection for the first device 400. The electrical connection may be an indirect electrical connection via a first daughter board 250 which includes a first electrical connection 251 which couples to 30 the first device 400 and a second electrical connection 252 which couples to the first port

- 13 -

- 210 of the interface device 200. The second port 220 is configured for receiving an electrical connection for the second device 500. Similarly to the first port 210, the electrical connection may be an indirect electrical connection via a second daughter board 260 which includes a first electrical connection 261 which couples to the second device
- 5 500 and a second electrical connection 262 which couples to the second port 220 of the interface device 200. In one form, the first and second ports 210, 220 utilise a 19 pin connection to couple with the daughter boards 250, 260 which allow SerDes compatible pins from the FPGA/SoC 620 to interface with the ports 210, 220.
- 10 Each third port 230, 235 is configured for enabling data communication of the interface device with a memory device or processing system 240, 300. More specifically, the interface device 200 includes a removable memory port 230 for receiving a memory device 240 such as a SD card as shown in Figure 2. As shown in Figure 3, the interface device 200 also includes a coupling port 235 for coupling the interface device 200 to a processing
- 15 system such as a desktop computer, laptop, or a mobile communication device 300, etc. As shown in Figure 3, an electrical cable can be coupled between the coupling port 235 of the interface device 200 and the processing system in the form of the mobile communication device 300 to enable an electrical connection between the two devices 200, 300.
- 20 Referring more specifically to Figure 6, the FPGA 620 is in electrical connection with the first port 210 and the second port 220 of the interface device 200. The interface device 200 can also include a processor 610 which is in electrical connection with the FPGA 620. In one form, the connection between the FPGA 620 and the processor 610 is via an 8 bit data line with a clock to send data packets from the FPGA 620 to the processor 610. As shown
- 25 in Figure 6, the processor 610 may be provided as part of a microcontroller such as a PIC24, however other types of microcontrollers can be utilised. The memory 630 of the interface device 200 is configured to store a configuration file received from the memory device or processing system 240, 300 via the at least one third port 230, 235, wherein loading of the configuration file configures the FPGA 620 to enable the first device 400 to
- 30 interface with the second device 500.

- 14 -

As shown in Figure 6, the interface device 200 also includes or is in electrical connection with a power source 640 which is in electrical connection with the processor 610 and the FPGA 620. Furthermore, the interface device 200 includes a clock 650 for the FPGA 620 and the processor 610. In one form, a 100MHz clock is provided for the FPGA 620 and a 20MHz clock for the processor 610.

At least one of the processor 610 and the FPGA 620 performs data manipulation upon first data received from the first device 400 to generate second data which is transferred to the second device 500. This data manipulation may be required due to different protocols being utilised by the first device 400 and the second device 500. Thus, the data manipulation performed by the interface device 200 allows the data arriving in the first protocol to be decoded and then encoded according to the second protocol via the interface device 200. This arrangement advantageously allows for a first device 400, which generally incompatible to interface with a second device, to interface with the second device 500 via the interface device 200 despite the different protocols that are being utilised by each device 400, 500.

A system can also be provided for obtaining a configuration file for configuring the interface device 200 to enable the first device 400 to interface with the second device 500 via the interface device 200. In particular, the system includes a processing system 300 operated by user in data communication with a server processing system 525.

Referring to Figure 1 there is shown a flowchart representing a method of configuring the interface device 200 to enable the first device 400 to interface with a second device 500. In particular, at step 110, the method 100 includes the server processing system 525 receiving from the processing system 300 first device type data indicative of a first device type. At step 120, the method 100 includes the server processing system 525 receiving from the processing system 300 second device type data indicative of a second device type. At step 130, the method 100 includes the server processing system 525 receiving from the

- 15 -

processing system 300 data manipulation data indicative of data manipulation to be performed to first data received from the first device 400 to generate second data for transfer to the second device 500. At step 140, the method 100 includes the server processing system 525 determining, using the first device type data, the second device type data, and the data manipulation data, a configuration file for configuring the interface device 200.

Referring to Figure 5 there is shown a flowchart representing a method of configuring the interface device to enable the first device 400 to interface with the second device 500 via the interface device 200. In particular, a user can interact with the server processing system 525 via the processing system 300 to obtain a configuration file for configuring the interface device 200 to enable the first device 400 to interface with the second device 500 via the interface device 200. The processing system 300 operated by the user may be a desktop processing system or a mobile communication device such as a smart phone, tablet processing system or similar. The server processing system 525 may host or be associated with a web server and/or an application server for enabling user interaction with the server processing system 525 for obtaining the configuration file for configuring the interface device 200. The mobile communication device 300 may execute a mobile device computer application which communicates with the application server of the server processing system to obtain the configuration file for configuring the interface device. The general processing system may be presented a user interface 505 via an web browser application.

User interaction with the server processing system 525 may include the user providing input to utilise an existing project 501 which the user has previously developed. Alternatively, the user may provide input to create a new project 520.

In the event that the user creates a new project, the user can interact with the processing system 300 to select an input interface 525 indicative of the type of device which the first device 400 is which is intended to be coupled to the first port 210 of the interface device 200. Additionally, the user can interact with the processing system 300 to select an output

- 16 -

interface 555 indicative of the type of device which the second device 500 is which is intended to be coupled to the second port 220 of the interface device 200. First and second data indicative of the first device type and the second device type are transferred from the processing system 300 to the server processing system 525.

5

The server processing system 525 selects an input device drivers/IP core 810 and an output device drivers/IP core 850 from a core/driver repository based upon the first and second data indicative of the first device type and the second device type. In particular, the core repository includes a plurality of available IP cores or device drivers which the user can select to utilise for defining the configuration file used for configuring the interface device 200. In particular, the core repository can include a plurality of input IP cores, a plurality of output IP cores and a plurality of data manipulation IP cores. The selected input and output IP cores define the decoding and encoding processes required to enable some form of data manipulation to be performed upon the received data.

15

The user is then required to define the data manipulation to be performed by the interface device 200 to enable the first device 400 to interface with the second device 500. This can be done in a number of manners. In one form, the user is able to select a pre-built data manipulation core 540 that is stored in the core repository of the server processing system 525. The data manipulation core is also referred to as a 'payload'. The server processing system 525 utilises a digital rights management system for controlling use of prebuilt IP cores stored in the core repository. The user generally is required to provide user identification data (e.g. username and password access or the like) to utilise the services of the server processing system 525, wherein user data stored in a user database 530 is accessible by the server processing system to enable the digital rights management system to determine what access to IP cores the user has. The server processing system 525 can provide access to one of the pre-built IP cores or device drivers based on a previous purchase by the user or the server processing system 525 enable the user to purchase rights to one or more of the pre-built IP cores available. It will be appreciated that one or more of the IP cores stored in the core repository may be freely accessible and thus the digital

30

rights management system can enable free use of the particular IP cores for users. The digital rights management system may also control user access to input and output IP cores.

5 In the event that the user selects one of the existing data manipulation cores 540, a project builder 560 of the server processing system 525 uses the input IP core 810, the output IP core 850 and the data manipulation core 540 to generate the configuration file 910. The project builder 560 generates the configuration file 910 in response to user interaction with the interface of presented at the processing system 300, wherein a build command is
10 transferred to the server processing system 525 to initiate the generation of the configuration file 910. Generally, the project builder performs a compilation process to generate the configuration file 910. As shown in Figure 7, build indication data 710 can be transferred back to the processing system 300 indicating the progress of the generation of the configuration file 910.

15

In an alternative scenario, the user may wish to define a custom data manipulation core 545. In this instance, the user can input code via the processing system which is transferred to the server processing system for defining a custom data manipulation core 545. In particular, the user can provide code to define the creation of a custom core VHDL using
20 RTL code or using a soft or hard processor and writing sequential code using a programming language such as a C-variant programming language, wherein the programming code is compiled and the executable file executed by the processor 610 of the interface device 200. Whilst other programming languages can be used, for the purposes of clarity, the remaining examples are discussed in relation to C-variant source
25 code being provided via the processing system 300. Manipulation of the FPGA 620 can be implemented via VHDL which can therefore produce very dedicated and fast payload. However, soft/hard processors can be defined easily by most software developers. The project builder 560 of the server processing system 525 utilises the custom data manipulation core 545, 550, and the selected input and output IP cores 525, 555 to
30 generate the configuration file 910 as discussed above.

- 18 -

The dynamic compilation performed by project builder 560 of the server processing system 525 produces the configuration file and optionally an ELF file if a C-payload is selected for the user's desired setting.

5

If the payload is designed in VHDL, the project builder 560 facilitates the following process:

1. Select the correct input and output cores 535, 555 to be combined in the project (editing relevant compilation files) along with the appropriate FIFO
10 buffer cores 820, 840 attached to them.
2. Connect the relevant RTL files for the VHDL payload together and group them as a project along with the correct cores. In one form the project may be a Xilinx project, however other types of projects can be utilised.
3. Compile the configuration file 910.

15

If the payload is designed in C, there are two possible cases: soft processor or hard processor.

For hard processor, the project builder 560 facilitates the following dynamic compilation
20 process:

1. Compile the C code to an ELF file with hard processor tools.
2. Load a pass through core into where the VHDL payload should be located.
3. Follow steps 1 to 3 of the VHDL payload compilation.

25 For a soft processor, such as a Microblaze soft processor 920 as shown in Figure 9, the server processing system is configured to facilitate the following dynamic compilation process:

1. Compile the C code to an ELF file with a soft processor tool (e.g. gcc).
2. Load the soft core into where the VHDL payload should be located.
- 30 3. Follow steps 1 to 3 of the VHDL compilation

- 19 -

4. Combine a generic boot load ELF file with the configuration file

In a further alternative scenario, the user may select from one of a plurality of predefined projects which are stored in a project repository which is accessible by the server processing system 525. In particular, the user is able to use the processing system 300 to
5 browse through the existing projects to select one for use in configuring their respective interface device 200. The user can provide search criteria such as input and output device types as input where in response the server processing system 525 conducts a search and presents search results indicative of projects which at least partially or fully satisfy the
10 search criteria. At least some or each existing project can be pre-built by the project builder module 560 of the server processing system 525, thus the server processing system 525 can obtain access to a pre-built configuration file 910 for provision to the user for configuring their respective interface device 200. Access to an predefined project can be controlled via the digital rights management system as described above. In particular, a user may be
15 required to purchase rights to a predefined project from a virtual payload shop 515, wherein the purchase of rights are recorded by the server processing system 525 for the user.

The configuration file 910, optionally the ELF file (if required) and a second configuration
20 file can then be transferred from the server processing system to the processing system 300 via the one or more networks. The transfer can be performed using a USB transfer class which can be either MTP (Media Transfer Protocol) or UMS (Universal Mass Storage).

In the event that the user is utilising a general processing system, the user may transfer the
25 configuration file 910 received from the server processing system 525 to a removable storage device 240 such as an SD card, USB storage device, a portable hard drive or the like. The removable storage device 240 can then be placed in data communication with one of the third ports 230 of the interface device 200 for boot loading to configure the interface device 200. Alternatively, the general processing system can be directly coupled to the

- 20 -

interface device 200 such that the configuration file is transferred directly from the processing system to the interface device 200.

In the event that the user is utilising a mobile processing system 300, the user can couple
5 the mobile processing system 300 with the interface device 200 via one of the third ports
235, wherein as shown in Figure 7 the user can interact with the interface of the application
740 to transfer the configuration file 910 from the mobile processing system 300 to the
memory 630 of the interface device 200. In particular, the user can select the interface
element 720 of the application interface 740 in order to initiate the transfer of the
10 configuration file 910. The interface device 200 can include a light emitting diode (LED)
which can flash when the transfer has been performed successfully.

The boot loading process by the microcontroller 610 of the interface device 200 initially
enumerates the connected device 240, 300 with the configuration file 910 to a UMS class.
15 Then the configuration file 910 is extracted and the data is transferred through a JTAG
connection to the FPGA 620 program lines. The microcontroller can then check a
configuration file stored on the connected device 240, 300 for several key parameters such
as whether a C-payload is to be loaded or not. If a C-payload is required, then the
microcontroller 610 loads a ELF file, stored in the connected device 240, 300 to a text
20 portion of the usable memory of the interface device 200. The microcontroller 610 finally
jumps to the text portion of the usable memory and starts the C-payload.

The microcontroller 610 can allow for USB Host connection to a UMS class device and
facilitate JTAG bit loading to the FPGA 620. Once it has completed the bit loading stage,
25 the interface device 200 can be used for processing and loading C-variant payloads.

Upon successfully configuring the interface device 200, the first device 400 can be coupled
to the first port 210 of the interface device 200 via the first daughter board 250 and the
second device 500 can be coupled to the second port 220 of the interface device 200 via
30 the second daughter board 260 such that data output by the first device 400 can undergo

- 21 -

decoding, manipulation and encoding by the interface device 200 to be output to the second device 500 such that the first device 400 interfaces with the second device 500 via the interface device 200.

5 The FPGA 620 internal architecture is shown in Figures 8 and 9. The compiler used by the server processing system produces this structure although the FIFO 820, 840 can be skipped to allow for faster processing system transfers which is referred to as "stream mode". The design of the FPGA 620 architecture for the interface device 200 includes three sections, namely an interface section 810, 850, buffer section 820, 840 and a payload
10 section 830. The interface section 810, 850 enables different interface protocols to decode the received data packets and/or to encode the sent data packets from the payload section. The buffer section 820, 840 allows for different interface cores to interact uniformly to the payload, as it can be dynamically compiled to be compatible with different bus standards (i.e. LocalBus, ACI, PLB, Wishbone, etc) which allows for external IP cores to be used.

15

As shown in Figure 7, a user can optionally select to share the created project via selecting the share interface element 730 of the interface 740. The server processing system 525 stores project data in the repository, wherein the digital rights management system can control access to the project data by other users.

20

The configured interface device 200 can be utilised for many applications. For example, in one form, the interface device 200 can be configured to enable a mobile communication device 400 to interface with an Ethernet device 500 as shown in Figure 4, thereby enabling fast network access which thereby allows for fast cloud access for mobile devices 400
25 (easy backups and allows for new concepts such as "mobile office").

In another example, a non-wireless device could be used with the interface device 200 to operate in a wireless manner. In particular, a wireless module could be coupled to the first port 210 of the interface device 200 and one or more speakers could be connected to the

- 22 -

second port 220 of the interface device 200. Thus, audio data could be wireless received by the speakers via the interface device 200.

The interface device 200 can also be utilised for co-processing or pre-processing. The
5 FPGA 620 can be used to address parallel processing problems. For example, it is known that gesture recognition can be performed poorly by certain mobile applications. Due to the parallel nature of image/video processing, filters can be applied easily on a parallel processing form. In particular, an external camera can be mounted on the mobile communication device. The external camera can provide images/video to the interface
10 device 200, wherein the interface device 200 can apply filters and send the processed data to the connected mobile communication device. This allows for greater frame rates and better quality data.

It will be appreciated that numerous variations and modifications may be made to this
15 invention. For example, more than two devices may be interfaced together by inclusion of additional connection ports and by appropriate inclusion of additional processing components/software in the interface device. Also, by way of example, it will be appreciated that the various electrical connections may be achieved either by hard-wired connections, or, wirelessly. Such variations and modifications will be apparent to a person
20 skilled in the art, and thereby should be considered to be covered by the invention as herein described and as hereinafter claimed.

The interface device 200 can optionally be configured through the first port 210. In this arrangement, the interface device 200 does not require an additional port for coupling the
25 interface device 210 to a memory device 300. The interface device 200 includes a switch which when actuated operates the interface in a configuration mode, wherein data received via the first port 210 from the memory device 300 coupled to the first port 210 is used to configure the FPGA 620. The switch can be actuated again to return the interface device to operating in an interface mode as discussed above.

- 23 -

Optional embodiments of the present invention may also be said to broadly consist in the parts, elements and features referred to or indicated herein, individually or collectively, in any or all combinations of two or more of the parts, elements or features, and wherein specific integers are mentioned herein which have known equivalents in the art to which
5 the invention relates, such known equivalents are deemed to be incorporated herein as if individually set forth.

Although a preferred embodiment has been described in detail, it should be understood that many modifications, changes, substitutions or alterations will be apparent to those skilled
10 in the art without departing from the scope of the present invention.

Claims

1. A server processing system for generating a configuration file for configuring an interface device to enable a first device to interface with a second device, wherein the
5 server processing system is configured to:
 - receive first device type data indicative of a first device type;
 - receive second device type data indicative of a second device type;
 - receive data manipulation data indicative of data manipulation to be performed to
first data received from the first device to generate second data for transfer to the second
10 device; and
 - determine, using the first device type data, the second device type data, and the data
manipulation data, a configuration file for configuring the interface device.
2. The server processing system according to claim 1, wherein the server processing
15 system generates the configuration file using the first device type data, the second device
type data, and the data manipulation data.
3. The server processing system according to claim 1 or 2, wherein the first device
type data, the second device type data and the data manipulation data are received via a
20 processing system, and wherein the server processing system transfers the configuration
file to the processing system which facilitates in the loading of the configuration file to the
interface device.
4. The server processing system according to claim 3, wherein the server processing
25 system has access to a repository including a plurality of device drivers/IP cores, wherein
the server processing system is configured to retrieve one or more of the device drivers/IP
cores for generating the configuration file.

- 25 -

5. The server processing system according to claim 4, wherein the server processing system utilises a digital rights management system for controlling use of one or more of the device drivers/IP cores in the repository for generating the configuration file.

5 6. The server processing system according to claim 4 or 5, wherein the first device type data is indicative of a selected first device type from a plurality of first device types available in the repository, wherein the server processing system retrieves, from the plurality of device drivers/IP cores in the repository, an input interface core corresponding to the first device type for generating the configuration file.

10

7. The server processing system according to any one of claims 4 to 6, wherein the second device type data is indicative of a selected second device type from a plurality of second device types available in the repository, wherein the server processing system retrieves, from the plurality of device drivers/IP cores in the repository, an output interface
15 core corresponding to the second device type for generating the configuration file.

8. The server processing system according to any one of claims 4 to 7, wherein the data manipulation data is indicative of a selected data manipulation from a plurality of data manipulations available in the repository, wherein the server processing system retrieves,
20 from the plurality of device drivers/IP cores in the repository, a data manipulation core corresponding to the data manipulation required for generating the configuration file.

9. The server processing system according to any one of claims 4 to 8, wherein the data manipulation data is indicative of code for performing the data manipulation, wherein
25 the server processing system uses the code for generating the configuration file.

10. The server processing system according to any one of claims 9, wherein the server processing system uses the code for generating an executable file, wherein the configuration file and executable file are loaded to the interface device to enable the first
30 device to interface with the second device.

- 26 -

11. A method for generating a configuration file for configuring an interface device to enable a first device to interface with a second device, wherein the method includes, in a server processing system, steps of:

- 5 receiving first device type data indicative of a first device type;
 receiving second device type data indicative of a second device type;
 receiving data manipulation data indicative of data manipulation to be performed to
first data received from the first device to generate second data for transfer to the second
device; and
10 generating, using the first device type data, the second device type data, and the
data manipulation data, a configuration file for configuring the interface device.

12. The method according to claim 11, wherein the method includes the server
processing system generating the configuration file using the first device type data, the
15 second device type data, and the data manipulation data.

13. The method according to claim 11 or 12, wherein method includes the server
processing system receiving the first device type data, the second device type data and the
data manipulation data from a processing system, and wherein the method includes the
20 server processing system transferring the configuration file to the processing system which
facilitates in the loading of the configuration file to the interface device.

14. The method according to claim 13, wherein the server processing system has access
to a repository including a plurality of device drivers/IP cores, wherein the method
25 includes the server processing system retrieving one or more of the device drivers/IP cores
for generating the configuration file.

15. The method according to claim 14, wherein the method includes the server
processing system utilising a digital rights management system for controlling use of one

- 27 -

or more of the device drivers/IP cores in the repository for generating the configuration file.

16. The method according to claim 14 or 15, wherein the first device type data is
5 indicative of a selected first device type from a plurality of first device types available in the repository, wherein the method includes the server processing system retrieving, from the plurality of device drivers/IP cores in the repository, an input interface core corresponding to the first device type for generating the configuration file.

10 17. The method according to any one of claims 14 to 16, wherein the second device type data is indicative of a selected second device type from a plurality of second device types available in the repository, wherein the method includes the server processing system retrieving, from the plurality of device drivers/IP cores in the repository, an output interface core corresponding to the second device type for generating the configuration
15 file.

18. The method according to any one of claims 14 to 17, wherein the data manipulation data is indicative of a selected data manipulation from a plurality of data manipulations available in the repository, wherein the method includes the server processing system
20 retrieving, from the plurality of device drivers/IP cores in the repository, a data manipulation core corresponding to the data manipulation required for generating the configuration file.

19. The method according to any one of claims 14 to 18, wherein the data manipulation
25 data is indicative of code for performing the data manipulation, wherein the method includes the server processing system using the code for generating the configuration file.

20. The method according to any one of claims 19, wherein the method includes the server processing system using the code for generating an executable file, wherein the

- 28 -

configuration file and executable file are loaded to the interface device to enable the first device to interface with the second device.

21. A computer program stored on a computer readable medium, the computer
5 program including executable instructions which configure a server processing system to perform the method of any one of claims 11 to 20.

22. A processing system for interacting with a server processing system for requesting
10 a configuration file for configuring an interface device, wherein the processing system is configured to:

transfer, to the server processing system, first device type data indicative of a first device type;

transfer, to the server processing system, second device type data indicative of a second device type;

15 transfer, to the server processing system, data manipulation data indicative of data manipulation to be performed to first data received from the first device to generate second data for transfer to the second device; and

receive, from the server processing system, a configuration file for configuring the interface device, wherein the configuration file is generated based upon the first device
20 type data, the second device type data, and the data manipulation data;

transfer the configuration file to memory of the interface device, wherein loading of the configuration file configures the interface device to enable a first device connected to the interface device to interface with a second device connected to the interface device.

25 23. The processing system according to claim 22, wherein the processing system is a mobile communication device operable to execute a mobile application for transferring the first device type data, the second device type data and the data manipulation data to the server processing system.

- 29 -

24. The processing system according to claim 23, wherein the first device type data is indicative of a selected first device type from a plurality of first device types for use by the server processing system to determine an input interface core for generating the configuration file.

5

25. The processing system according to claim 23 or 24, wherein the second device type data is indicative of a selected second device type from a plurality of second device types for use by the server processing system to determine an output interface core for generating the configuration file.

10

26. The processing system according to any one of claims 23 to 25, wherein the data manipulation data is indicative of a selected data manipulation from a plurality of data manipulations available from the server processing system for generating the configuration file.

15

27. The processing system according to any one of claims 23 to 26, wherein the data manipulation data is indicative of code for performing the data manipulation, wherein the code is input via the mobile application, wherein the programming code is transferred to the server processing system for generating the configuration file.

20

28. The processing system according to any one of claims 22 to 27, wherein the processing system provides user identification data to the server processing system for use in a digital rights management system for generating the configuration file.

25 29. A method for interacting with a server processing system for requesting a configuration file for configuring an interface device, wherein method includes, in a processing system, steps of:

transferring, to the server processing system, first device type data indicative of a first device type;

- 30 -

transferring, to the server processing system, second device type data indicative of a second device type;

transferring, to the server processing system, data manipulation data indicative of data manipulation to be performed to first data received from the first device to generate
5 second data for transfer to the second device; and

receiving, from the server processing system, a configuration file for configuring the interface device, wherein the configuration file is generated based upon the first device type data, the second device type data, and the data manipulation data;

transferring the configuration file to memory of the interface device, wherein
10 loading of the configuration file configures the interface device to enable a first device connected to the interface device to interface with a second device connected to the interface device.

30. The method according to claim 29, wherein the processing system is a mobile
15 communication device, wherein the method includes the mobile communication device executing a mobile application for transferring the first device type data, the second device type data and the data manipulation data to the server processing system.

31. The method according to claims 29 or 30, wherein the mobile communication
20 device is adapted to transfer the configuration file, by either wired or wireless transmission.

32. The method according to claim 30 or 31, wherein the first device type data is indicative of a selected first device type from a plurality of first device types for use by the
25 server processing system to determine an input interface core for generating the configuration file.

33. The method according to any one of claims 30 to 32, wherein the second device type data is indicative of a selected second device type from a plurality of second device
30 types for use by the server processing system to determine an output interface core for generating the configuration file.

- 31 -

34. The method according to any one of claims 30 to 33, wherein the data manipulation data is indicative of a selected data manipulation from a plurality of data manipulations available from the server processing system for generating the configuration file.

5

35. The method according to any one of claims 30 to 34, wherein the data manipulation data is indicative of code for performing the data manipulation, wherein the method includes inputting the code via the mobile application, wherein the programming code is transferred to the server processing system for generating the configuration file.

10

36. The method according to any one of claims 29 to 35, wherein the method includes the processing system providing user identification data to the server processing system for use in a digital rights management system for generating the configuration file.

15

37. A computer program stored on a computer readable medium, the computer program including executable instructions which configure processing system to perform the method of any one of claims 29 to 36.

20

38. An interface device for enabling a first device to interface with a second device, wherein the device includes:

a first port for receiving an electrical connection for the first device;

a second port for receiving an electrical connection for the second device;

a third port for connecting the interface device to a memory device or processing system;

25

a programmable device in electrical connection with the first port and the second port;

a memory for storing a configuration file received from the memory device or processing system via the third port, wherein execution of the configuration file configures the programmable device to enable the first device to interface with the second device.

30

- 32 -

39. An interface device for enabling a first device to interface with a second device, wherein the device includes:

a first port:

for receiving an electrical connection for the first device; and

5 for connecting the interface device to a memory device or processing system;

a second port for receiving an electrical connection for the second device;

a switch for selectively operating the interface device in a configuration mode and an interface mode; and

10 a programmable device in electrical connection with the first port and the second port;

a memory for storing a configuration file received from the memory device or processing system via the first port when the interface device is operating in the configuration mode, wherein execution of the configuration file configures the
15 programmable device to enable the first device to interface with the second device when the interface device operates in the interface mode.

40. The interface device according to claim 38 or 39, wherein the programmable device is configurable to perform data manipulation upon first data received from the first device
20 to generate second data which is transferred to the second device.

41. The interface device according to claim 40, wherein the interface device includes a processor in electrical connection with the programmable device, wherein the memory stores an executable file for execution by the processor to enable the first device to
25 interface with the second device.

42. The interface device according to claim 41, wherein the processor is configured, upon execution of the executable file, to perform data manipulation upon the first data received from the first device to generate second data which is transferred to the second
30 device.

- 33 -

43. The interface device according to claim 38 to 42, wherein the first data is in accordance with a first protocol and the second data is in accordance with a second protocol, wherein the first and second protocols are different.

5

44. The interface device as claimed in any one of claims 38 to 43, wherein the programmable device includes a Field Programmable Gate Array (FPGA), a System on Chip (SoC) device and/or any other like device which may be programmed by a user (preferably in the field after manufacture of the device) and which is preferably provided
10 in an integrated circuit (IC) or like form, integrating the various components of a computer system into a single chip/embedded form.

45. An interface device as claimed in anyone of claims 38 to 44, further including:
at least one further port for receiving an electrical connection for a further device which is
15 to be interfaced with said first and second devices, and wherein, said programmable device is provided in electrical connection with each of said further ports, and wherein execution of the configuration file configures the programmable device to enable each of said devices to be interfaced together.

20 46. An interface device as claimed in any one of claims 38 to 45, wherein each said device port is connected to its respective port either by wired or wireless connection.

- 1 / 6 -

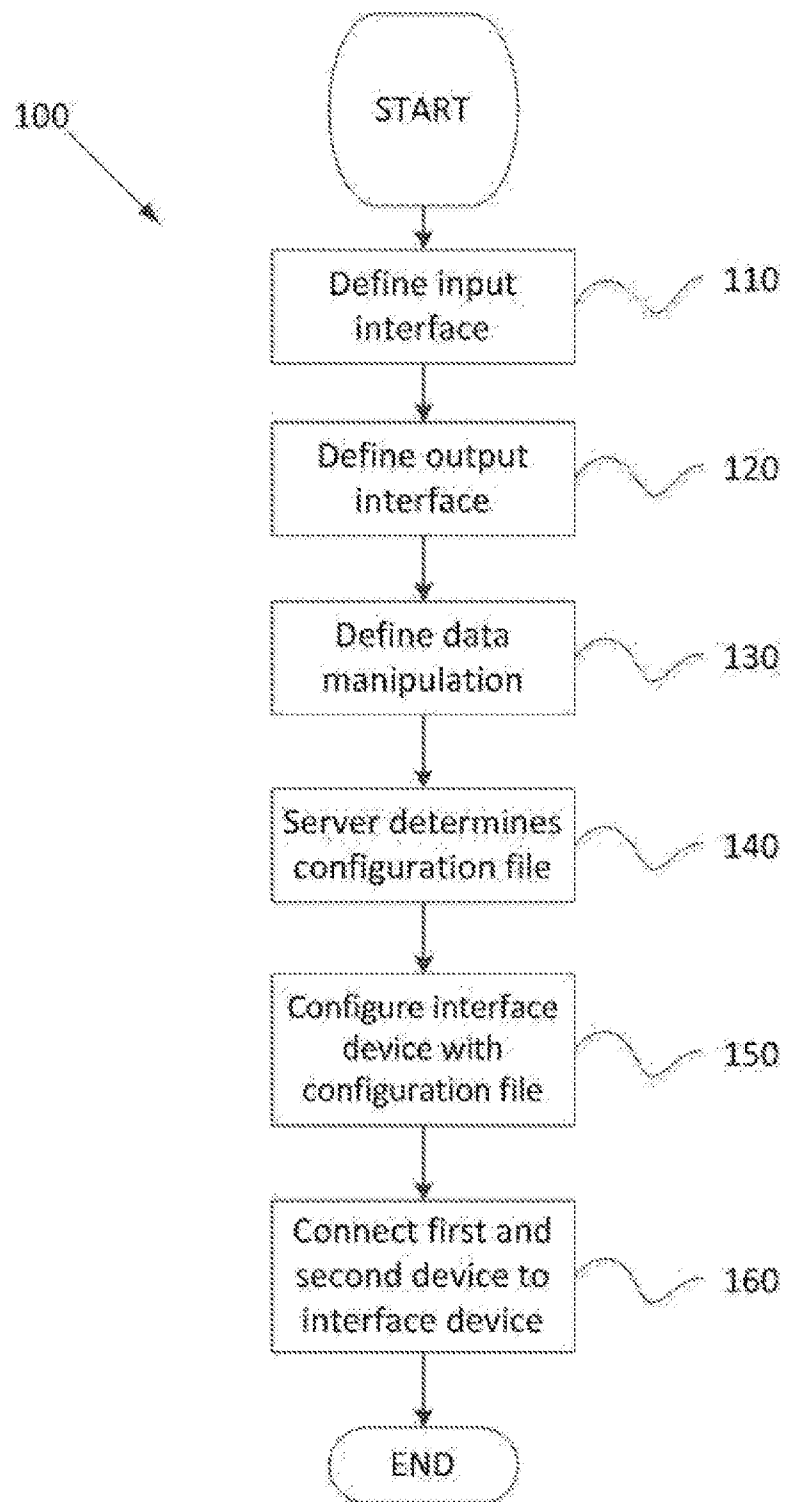


FIGURE 1

- 2 / 6 -

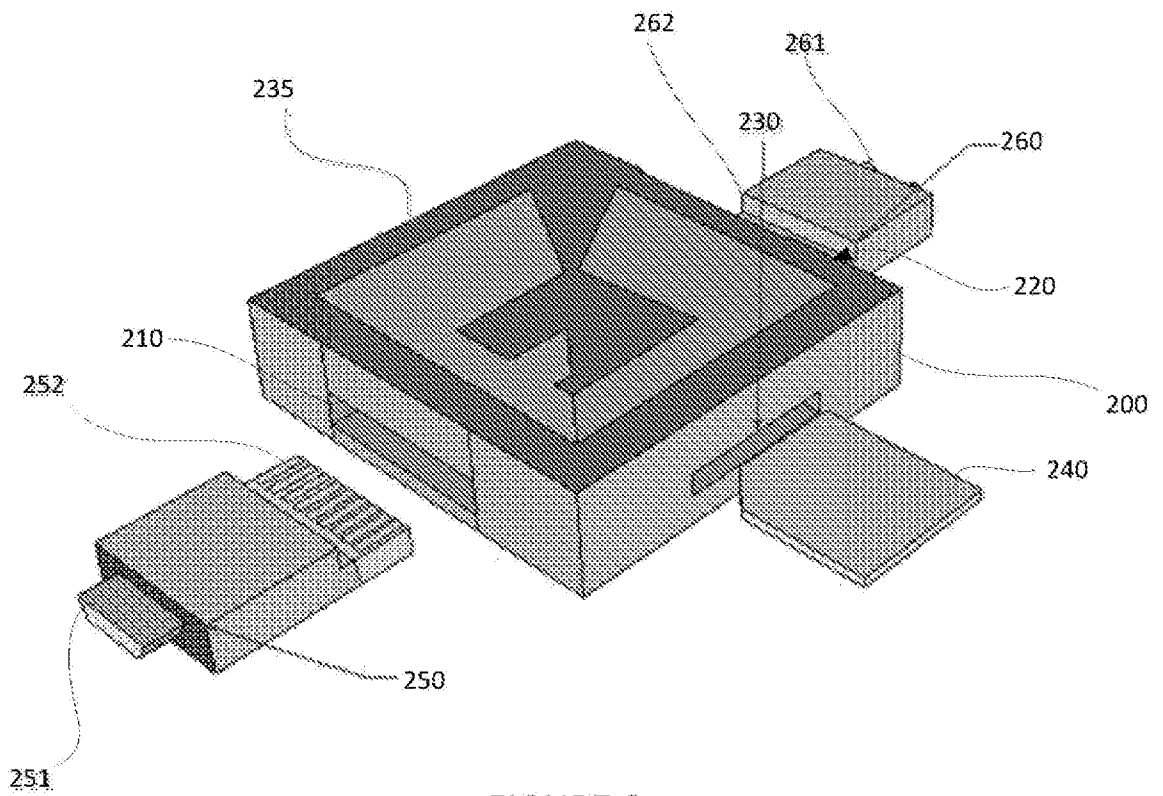


FIGURE 2

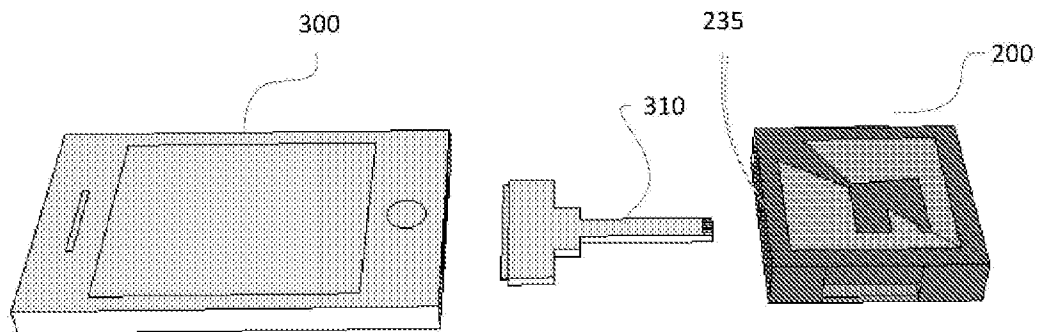
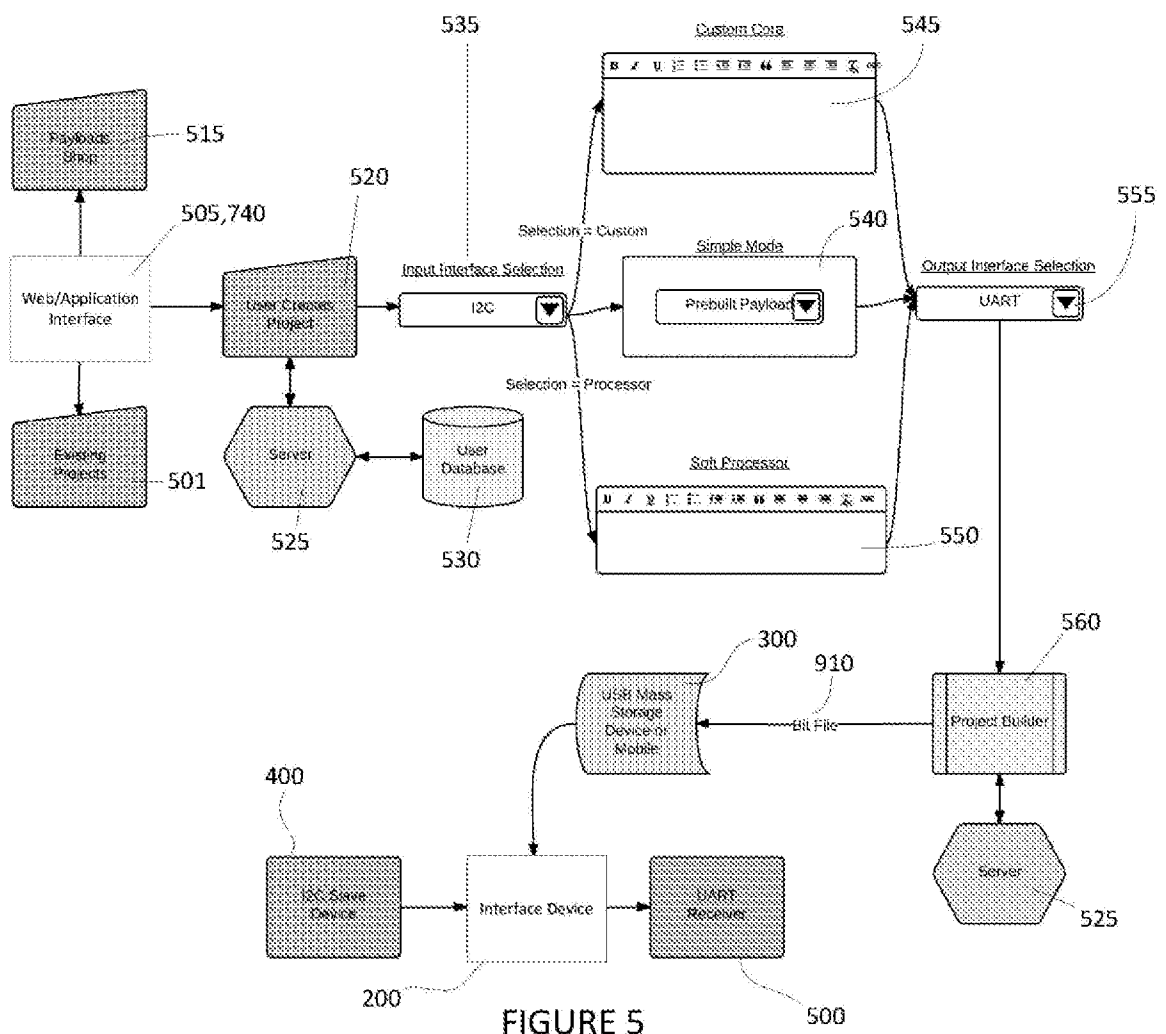
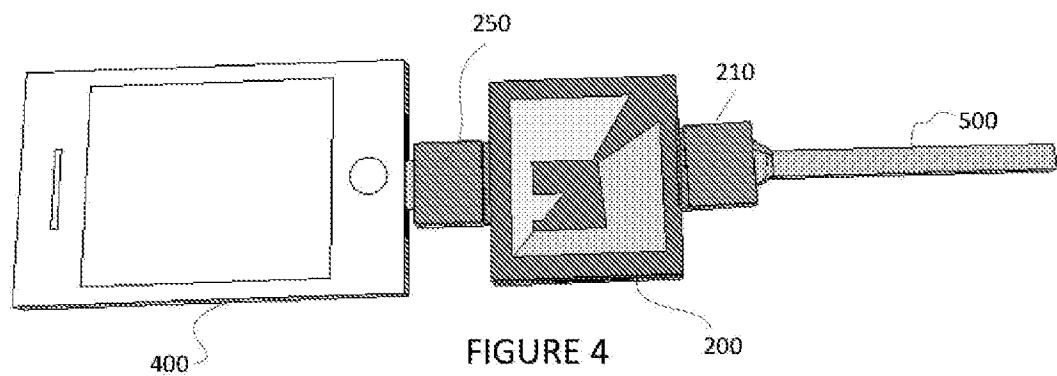


FIGURE 3

- 3 / 6 -



- 4 / 6 -

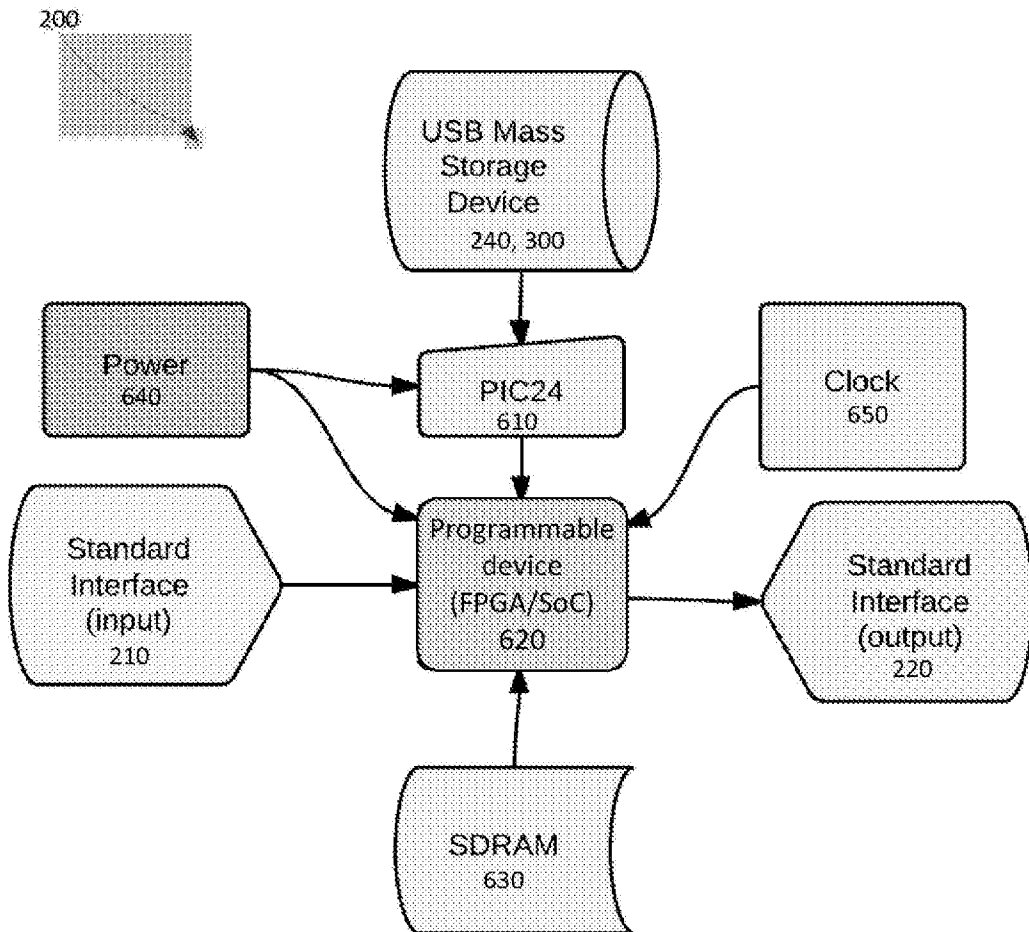


FIGURE 6

- 5/6 -

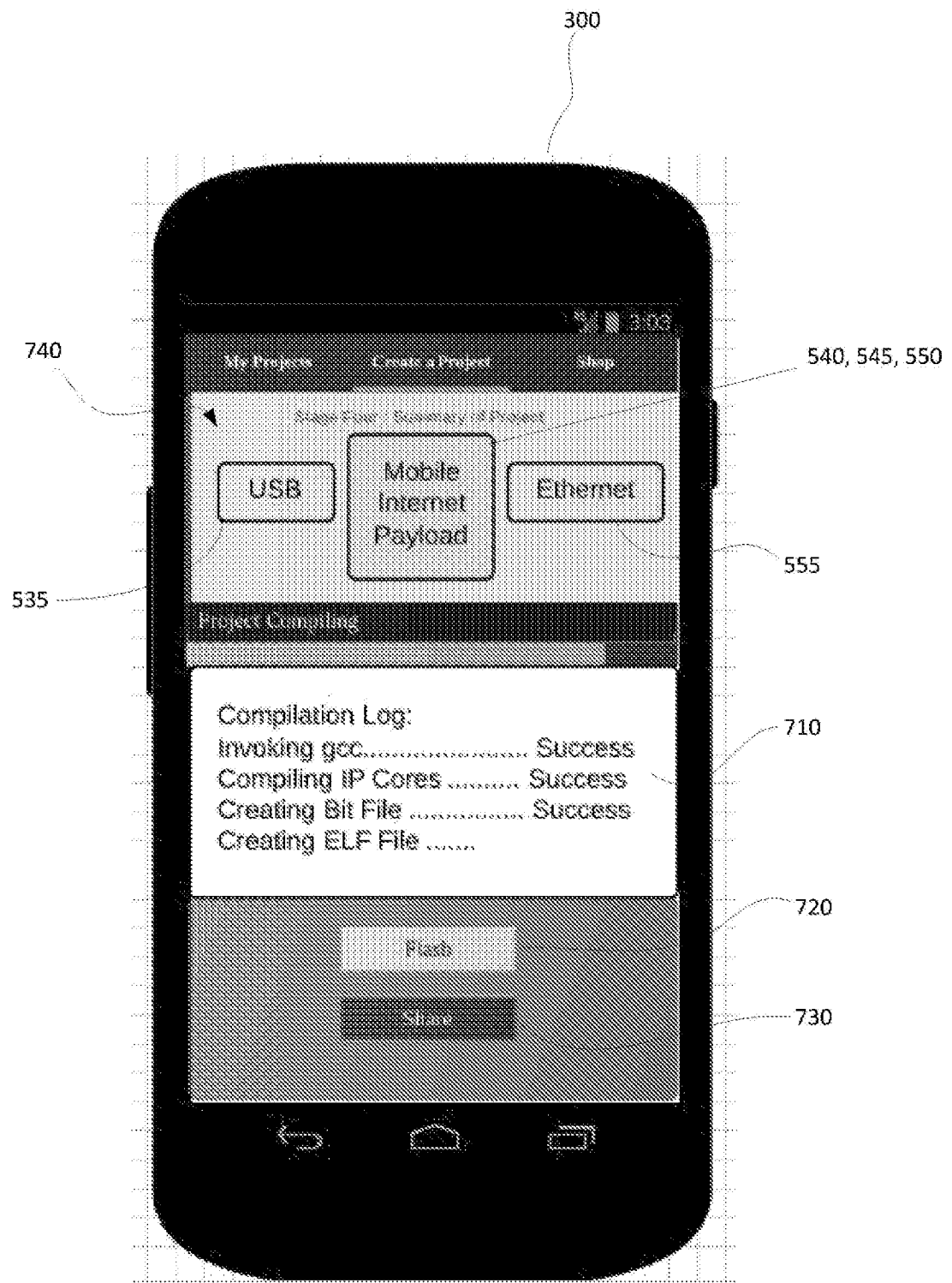


FIGURE 7

- 6 / 6 -

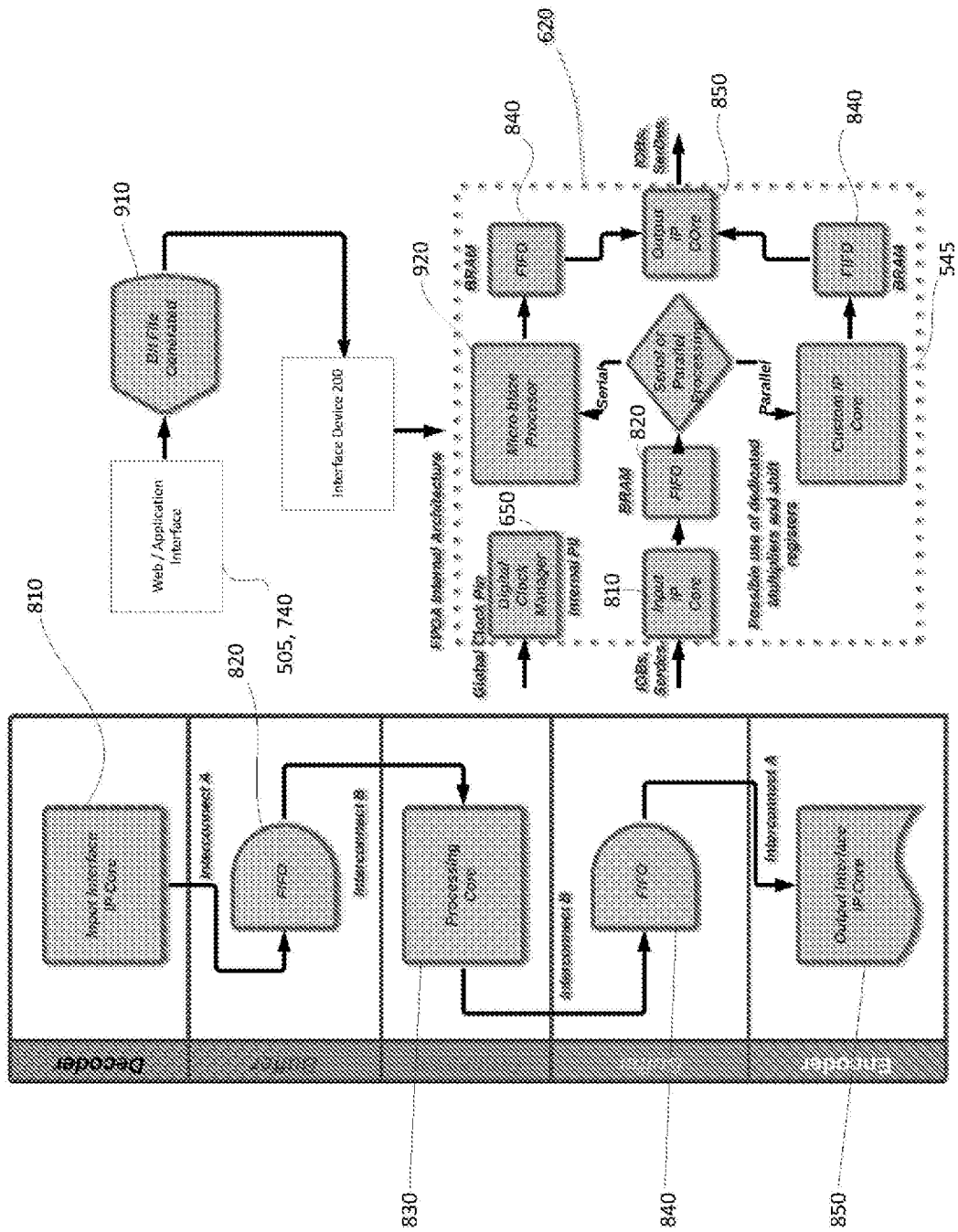


FIGURE 9

FIGURE 8

INTERNATIONAL SEARCH REPORT

International application No.
PCT/AU2015/050010

A. CLASSIFICATION OF SUBJECT MATTER

G06F 5/00 (2006.01)

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPODOC, WPI: Keywords (interface, device, server, configuration, setup, emulate, port, adapter, translate, convert) and like terms.

Google Patents, Google: Similar keywords as above.

Espace, Google Scholar, Google: Applicant/Inventor name search.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
	Documents are listed in the continuation of Box C	



Further documents are listed in the continuation of Box C



See patent family annex

* Special categories of cited documents: "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier application or patent but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed		"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family
Date of the actual completion of the international search 12 March 2015	Date of mailing of the international search report 12 March 2015	
Name and mailing address of the ISA/AU AUSTRALIAN PATENT OFFICE PO BOX 200, WODEN ACT 2606, AUSTRALIA Email address: pct@ipaustalia.gov.au		Authorised officer MD Reza-E Rabbi AUSTRALIAN PATENT OFFICE (ISO 9001 Quality Certified Service) Telephone No. 0262833141

INTERNATIONAL SEARCH REPORT		International application No.
C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		PCT/AU2015/050010
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 4972470 A (FARAGO) 20 November 1990 Abstract, lines 38 to 54 column 2, lines 15 to 27 column 3, fig 1.	1-46
Y	US 8539048 B2 (LIN et al.) 17 September 2013 Abstract, line 59 column 1 to line 14 column 2, lines 30 to 55 column 3, fig 2, 3.	1-46
A	US 7380705 B2 (TERLIZZI et al.) 03 June 2008	
A	US 2007/0299996 A1 (GUY et al.) 27 December 2007	
A	US 4079452 A (LARSON et al.) 14 March 1978	

Form PCT/ISA/210 (fifth sheet) (July 2009)

INTERNATIONAL SEARCH REPORT Information on patent family members		International application No. PCT/AU2015/050010	
This Annex lists known patent family members relating to the patent documents cited in the above-mentioned international search report. The Australian Patent Office is in no way liable for these particulars which are merely given for the purpose of information.			
Patent Document/s Cited in Search Report		Patent Family Member/s	
Publication Number	Publication Date	Publication Number	Publication Date
US 4972470 A	20 November 1990		
US 8539048 B2	17 September 2013	US 8539048 B2	17 Sep 2013
		CN 102238140 A	09 Nov 2011
US 7380705 B2	03 June 2008	US 7380705 B2	03 Jun 2008
		CN 1842778 A	04 Oct 2006
		EP 1668526 A1	14 Jun 2006
		EP 1668526 B1	10 Dec 2008
		JP 2007503650 A	22 Feb 2007
		WO 2005024641 A1	17 Mar 2005
		WO 2005024641 B1	26 May 2005
US 2007/0299996 A1	27 December 2007		
US 4079452 A	14 March 1978	CA 1105585 A1	21 Jul 1981
		DE 2726811 A1	29 Dec 1977
		FR 2355332 A1	13 Jan 1978
		GB 1573046 A	13 Aug 1980
		GB 1573047 A	13 Aug 1980
		JP S5316542 A	15 Feb 1978
		JP S6058497 B2	20 Dec 1985
End of Annex			
Due to data integration issues this family listing may not include 10 digit Australian applications filed since May 2001. Form PCT/ISA/210 (Family Annex)(July 2009)			