(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
7 October 2004 (07.10.2004)

PCT

(10) International Publication Number
WO 2004/086240 A1

(51) International Patent Classification[7]: G06F 13/28

(21) International Application Number:
PCT/IB2004/050326

(22) International Filing Date: 24 March 2004 (24.03.2004)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
03100820.4 28 March 2003 (28.03.2003) EP

(71) Applicant *(for all designated States except US)*: **KONINKLIJKE PHILIPS ELECTRONICS N.V.** [NL/NL]; Groenewoudseweg 1, NL-5621 BA Eindhoven (NL).

(72) Inventor; and

(75) Inventor/Applicant *(for US only)*: **SREEDHARAN, Sanjeev, M.** [IN/IN]; c/o Prof. Holstlaan 6, NL-5656 AA Eindhoven (NL).

(74) Agent: **DE, JONG, Durk, J.**; Prof. Holstlaan 6, NL-5656 AA Eindhoven (NL).

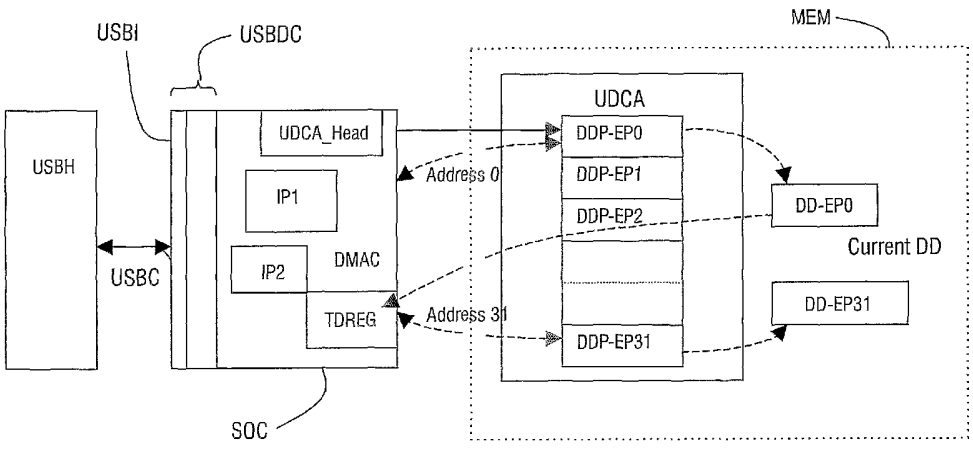
(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*
— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: DATA PROCESSING SYSTEM WITH A DMA CONTROLLER FOR STORING THE DESCRIPTOR OF THE ACTIVE CHANNEL



(57) Abstract: A data processing system according to the invention comprises a USB device function (USBDC) for communicating with a USB Host controller (USBH), a DMA controller (DMAC) for enabling direct access to a memory (MEM), and a storage facility (MEM) for storing data and control information (DD-EPO DD-EP3 1) relating to the DMA transmission. The control information includes information indicative for the location of a buffer space (dma buffer start_addr) to be used by one or more DMA transmissions.According to the invention the control information is stored in a memory (MEM), the memory further comprising reference information (DDP-EPO,.... DDP/EPP3 1) indicating the location of the control information.

WO 2004/086240 A1

DATA PROCESSING SYSTEM WITH A DMA CONTROLLER FOR STORING THE DESCRIPTOR OF THE ACTIVE CHANNEL

The invention relates to a data processing system.

The invention further relates to a method for transmitting data.

In the few years since its introduction Universal Serial Bus (USB) has become a de facto Industry standard for connecting peripherals to PCs and laptops for data exchange. Today an increasing number of mobile consumer products - portable digital assistants (PDAs), mobile phones, digital cameras, portable storage devices etc uses the USB interface to exchange data with the host PCs. The peripherals implement the device functionality and the PC the host functionality. Endpoints inside the device are the source or recipient of data. A USB device can have a maximum of 32 physical endpoints.

Figure 1 schematically shows an arrangement comprising a host controller USBH, a system on chip SOC coupled via a USB cable USBC to the host device and a system memory MEM. The system on chip comprises a USB device function USBDC, and a memory controller MC which are mutually coupled to an internal bus, for example a AHB bus as shown. Other devices may be coupled to the internal bus as well. The USB device function includes a DMA controller.

For the data transfer the host establishes a transfer pipe to the endpoints.

The host will send data to or receive data from endpoints implemented inside the device through the USB cable USBC. In Figure 1 endpoints EP1 and EP3 will receive data from the Host. They are referred as OUT endpoint. EP2 and EP4 are capable of sending the data towards the host and are referred as IN endpoints. The data transfer is done in packets whose format is defined in the USB protocol. Each endpoint will have a buffer, which is capable of holding at least one packet of data corresponding to the endpoint.

Each endpoint will have an assigned space in the system memory MEM, which will act as the destination or source for the data packet. The logical view of the

data transfer from the endpoint to the memory is shown by the dotted lines. Physical data transfer happens through the AHB bus and memory controller MC.

An OUT endpoint should transfer the packet data it received from the host USBH to the system memory. MEM before the next packet arrives. From there it may be transferred to another destination if desired. Similarly an IN endpoint should retrieve one packet of data from the system memory MEM before the request for the data from the host arrives.

Data can be transferred from the endpoint buffer to the system memory MEM through the processor of the SOC or Direct Memory Access (DMA). DMA is preferred for the data transfer as it is faster. In DMA mode the Endpoints directly transfer the data to or from the memory through the DMA Controller implemented inside the USB Device core.

Figure 2 illustrates a conventional implementation of a DMA Controller. The DMA Controller comprises a DMA control logic DMAC, and a set of registers DMAREGS comprising information used by the DMA controller. The sets of registers are indicated as EP1_DMAREGS for Endpoint 1; EP2_DMAREGS for endpoint 2 and so on. The following information is comprised in the registers:

The start address of the DMA buffer in the system memory MEM, (EP1_B, EP2_B etc)

The length of the DMA Buffer in the system memory MEM,

Control information, such as the DMA-mode used and the maximum packet length,

DMA count information (Number of bytes transferred)

DMA status information

A DMA transfer can be characterised by a structure describing these parameters.

To support continuous transfer of data on an endpoint the DMA controller should keep the above information for the current and next buffers. The set of DMA registers for each endpoint will comprise:

DMA Start Address register current (32 bit wide)

DMA Start Address register next (32 bit wide)

DMA length Register current (16 bit wide)

DMA length Register next (16 bit wide)

DMA Count Register current (16 bit wide)

DMA count Register next (16 bit wide)

DMA control register (8 bit wide)

A major drawback of this conventional implementation is that the number of registers required inside the DMA controller increases proportionally with the number of endpoints. This will cause a corresponding increase in gate count (silicon area). This is schematically illustrated in Figure 3. As shown therein the gate count is about 12K for a single endpoint and increases with about 0.8K for each succeeding end-point.

It is a purpose of the invention to provide a data processing system as well as a method which allows using a relatively low number of registers independent of the number of endpoints involved.

A data processing system according to the invention is described by claim 1.

A method according to the invention is described by claim 2.

The proposed architecture uses a single physical resource to serve multiple endpoints in a time division multiplexed basis. The DMA controller uses the system memory itself to keep the information for the DMA transfers. DMA_REGS will not be implemented inside the DMA controller.

In the architecture of the invention the required number of gates is relatively low and independent of the allowable number of end-points.

In the implementation according to claim 2 it has a set of place-holder registers for keeping the status of a running DMA operation. The information need to conduct a DMA transfer will be distributed inside the memory as DMA descriptors (DD). Software create the DD and distribute it in the memory. The place-holder registers will be filled with the contents of DD when the DMA transfer is required.

These and other aspects of the invention as described with reference to the drawings. Therein

Figure 1 shows a conventional data processing device,

Figure 2 shows a conventional DMA controller,

Figure 3 shows the relation between the required number of gates and the number of end-points in the conventional DMA controller,

Figure 4 shows a data processing device according to the invention,

Figure 5 shows a method for transmitting data according to the invention,

Figure 6 shows the relation between the required number of gates and the number of end-points in the DMA controller of the data processing device according to the invention.

Figure 4 schematically shows a data processing system according to the invention. The data processing system comprises a USB interface USBDC for communicating with a USB host device USBH. The USB host is for example a PC, while the data processing system is for example a mobile consumer product, such as a portable digital assistant PDAs, mobile phone, a digital camera, or a portable storage device. The data processing system further comprises a DMA controller DMAC for enabling direct access to a memory MEM and a storage facility MEM for storing control information DD-EP0, ...DD-EP31 relating to the DMA transmission. The control information, which will be described in the sequel in more detail, includes at least information indicative for the location of a buffer space **dma_buffer_start_addr** to be used by one or more DMA transmissions.

According to the invention the data processing system is characterized in that the control information is stored in a memory (MEM). The memory further comprises reference information (DDP-EP0, ..., DDP-EP31) indicating the location of the control information. The DMA controller has a single set of registers (TDREG) for storing a copy of the control information related to the active DMA stream.

The control information information DD-EP0, ...DD-EP31 is stored in a memory area called USB Device Communication Area (UDCA) of the memory MEM, in which the DMA controller and the USB controller further store the device driver software. Each endpoint is assigned a reserved location in this area which holds a pointer (DDP) to the location where the data descriptor DD for that endpoint is kept. When a DMA transfer is requested for an endpoint the DMA controller fetches the DDP from the corresponding location for the endpoint. The start address of the UDCA area is defined in the location UDCA_Head defined in the DMA Controller. The location for the DDP are derived from the endpoint number and the UDCA head register value. In practice the location is derived by adding the endpoint number

multiplied by 4 to the UDCA head register value, when using word aligned addresses. The location of the data descriptor pointer DDP31 for the 31th endpoint is for example calculated as addr31 = UDCA_Head + 31*4.

A DMA descriptor DD represents one DMA transfer unit for the endpoint. When the buffer indicated by the DD is full the DD will be moved to the retired status. DMA descriptors for an endpoint form a linked list, i.e. one DD points to the next DD to be serviced. Thus when the current DD is retired the DMA controller is capable of fetching the next DD. Consequently the DMA transfer can go on for an indefinite period of time.

A DMA transfer can be characterised by a structure describing the parameters controlling the DMA . This structure is called the DMA Descriptor.

The DD is a structure consisting of 4 words (16 bytes). The fields are defined as below.

Word position	Bit position	Description
0	31:0	Next_dd_pointer(system memory address)
1	1:0	Dma_mode (00 -Normal)
	2	Next_DD_valid (1 - valid; 0 - invalid)
	3	Reserved
	4	Isochronous_endpoint (1 - isochronous; 0 - non-isochronous)
	15:5	Max_packet_size
	31: 16	Dma_buffer_length
2	31:0	Dma_buffer_start_addr
3	0	DD_retired (To be initialised to 0)
4	4:1	DD_status (To be initialised to 0) 0000 - not serviced 0001 - being serviced 0010 - normal completion 0011 - data under run (short packet) 1000 - data over run 1001 - system error
	5	Packet_valid (To be initialised to 0)
	15:6	Reserved
	31:16	Present_dma_count (To be initialised to 0)

next_DD_pointer

Pointer to the memory location from where the next DMA descriptor has to be fetched.

dma_mode

Defines which mode DMA has to operate.

next_DD_valid

This bit indicates whether the software has prepared the next DMA descriptor. If it is valid the DMA controller once finished with the current descriptor will load the new descriptor.

isochronous_endpoint

The descriptor belongs to an isochronous endpoint.

max_packet_size

This specifies the maximum packet size of the endpoint. This parameter has to be used while transferring the data for IN endpoints from the memory.

dma_buffer_length

This indicates the depth of the DMA buffer allocated for transferring the data. The DMA controller will stop using this descriptor when this limit is reached and will look for the next descriptor

dma_buffer_start_addr

The address from where the data has to be picked up or to be stored. This field is updated packet-wise by DMA controller.

DD_retired

This bit is set when the DMA controller finishes the current Descriptor. This will happen when the end of the buffer is reached or a short packet is transferred (no isochronous endpoints) or an error condition is detected.

DD_status

The status of the DMA transfer is encoded in this field.

packet_valid

This bit indicates that the last packet transferred to the memory is received with errors or not

present_dma_count

The number of bytes transferred by the DMA controller at any point of time. This is updated packet-wise by the DMA controller when it updates the descriptor.

A method for transmitting data according to the invention is illustrated with reference to Figure 5. According to the method shown the DMA controller fetches a DMA descriptor pointer DD corresponding to an end-point in step a.

In step b the DMA descriptor is fetched from the location pointed to by the DMA descriptor pointer DDP.

In step c it is checked whether the DMA descriptor is in a retired state. Subsequently in step d the DMA descriptor is copied in a set of registers TDREG, provided that the DMA descriptor is not in Retired state. If DD is in a retired state DDP is

updated in step e by the 'next_dd_pointer' in the currently fetched DD. Then steps b and c are repeated.

In step f a packet transfer is executed using the data copied into the set of registers TDREG,

In step g the copied DMA descriptor is updated to take into account the changes in the status of the transfer for the selected endpoint,

In step h the updated DMA descriptor is written back to the memory location from where it was read.

Typically the host will send the packets in a multiplexed fashion to the device. For example the first packet may belong to endpoint EP2 the second one from endpoint EP4 etc. Hence, the serviced endpoint changes from packet to packet. However, a case may occur, where a plurality of packets originate from the same endpoint, e.g. EP2. In this case it is not necessary to fetch the DMA descriptor DD again and again for the packets as its value does not change. Unncessary fetching of the DMA descriptor DD and its pointer DDP can be avoided by maintaining a flag. When set, it will prevent a new fetch of DDP and DD. The flag is reset when the endpoint is different or DD is retired.

Figure 6 schematically illustrates that the data processing system and the method for transferring data according to the invention allow an arbitrary number of end points using an architecture having a relatively low number of gates, the number being independent of the number of end-points.

CLAIMS:

1. Data processing system comprising
- a USB device function (USBDC) for communicating with a USB Host controller (USBH),
- a DMA controller (DMAC) for enabling direct access to a memory (MEM),
- a storage facility (MEM) for storing data and control information (DD-EP0, DD-EP31) relating to the DMA transmission, including information indicative for the location of a buffer space (dma_buffer_start_addr) to be used by one or more DMA transmissions,

characterized in that,

the control information is stored in a memory (MEM), the memory further comprising reference information (DDP-EP0, ..., DDP-EP31) indicating the location of the control information,

2. Data processing system according to claim 1, characterized in that the DMA controller has a single set of registers (TDREG) for storing a copy of the control information related to the active DMA stream.

3. Data processing system according to claim 1 or 2, characterized in that the reference information (DDP-EP0, ..., DDP-EP31) and the driver software for the USB device function are stored in a common area (UDCA) of the memory (MEM).

4. Data processing system according to claim 3, characterized in that the start address of the common area is defined in a register (UDCA-Head) in the DMA Controller (DMAC).

5. Method for transmitting data comprising the steps of

c. checking the status (c) of the DMA descriptor,

c1. carrying out steps d, f, g and h if the status is not "RETIRED"

c2. carrying out step e if the status is retired and continuing with step b.

d. copying (d) the DMA descriptor in a set of registers,

f. executing (f) a packet transfer using the copied data,

g. updating (g) the copied DMA descriptor

h. writing back the updated DMA descriptor to the memory location from where it was read.

e. updating the DMA descriptor pointer and continuing with step b.

6. Method for transmitting data according to claim 5, further comprising the steps of

i. verifying a flag indicating whether the DMA descriptor relates to a new endpoint, and if this is true performing in addition steps a and b before step c.

a. fetching (a) a DMA descriptor pointer (DD) corresponding to an end-point,

b. fetching (b) the DMA descriptor from the location pointed to by the DMA descriptor pointer (DDP).

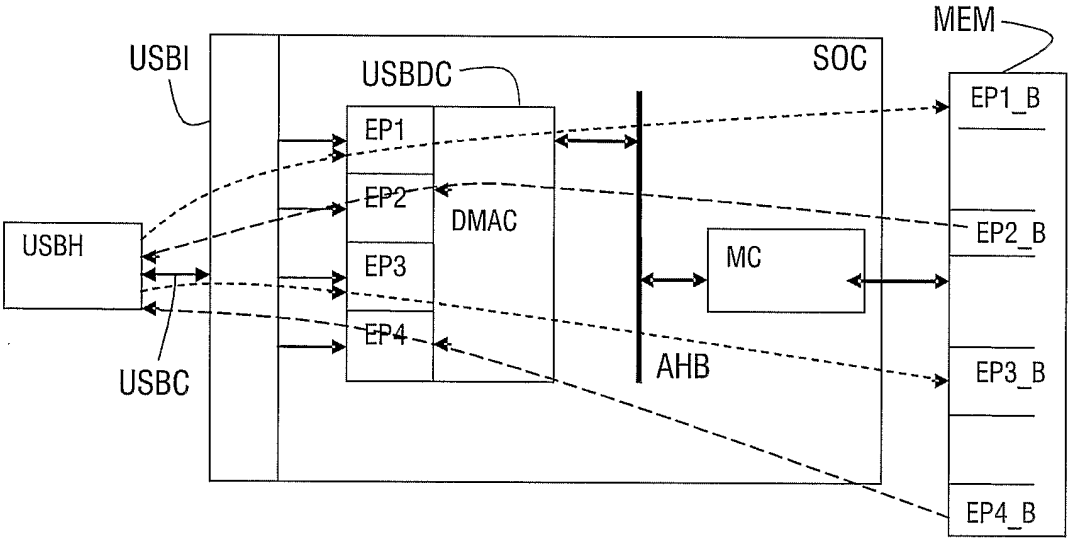

MEM
USBI
USBDC
SOC
EP1_B
EP1
EP2
DMAC
EP2_B
USBH
MC
EP3
EP4
AHB
EP3_B
USBC
EP4_B

FIG.1

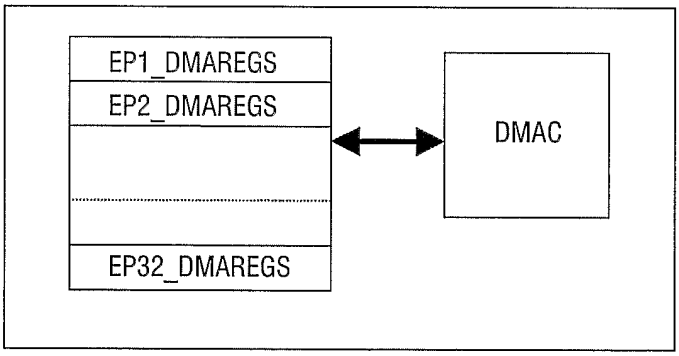

EP1_DMAREGS
EP2_DMAREGS
DMAC
EP32_DMAREGS

FIG.2

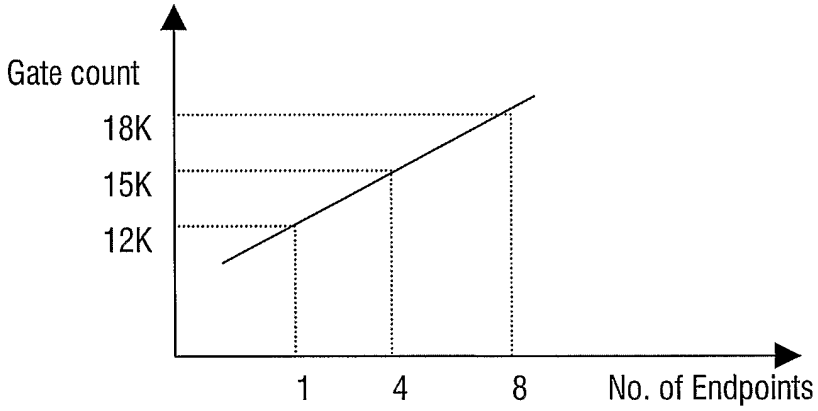

Gate count
18K
15K
12K
1
4
8
No. of Endpoints

FIG.3

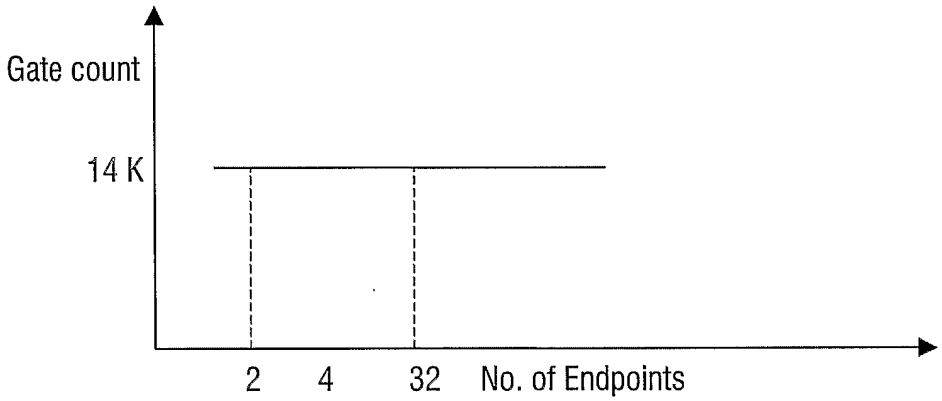

Gate count
14 K
2
4
32
No. of Endpoints

FIG.6

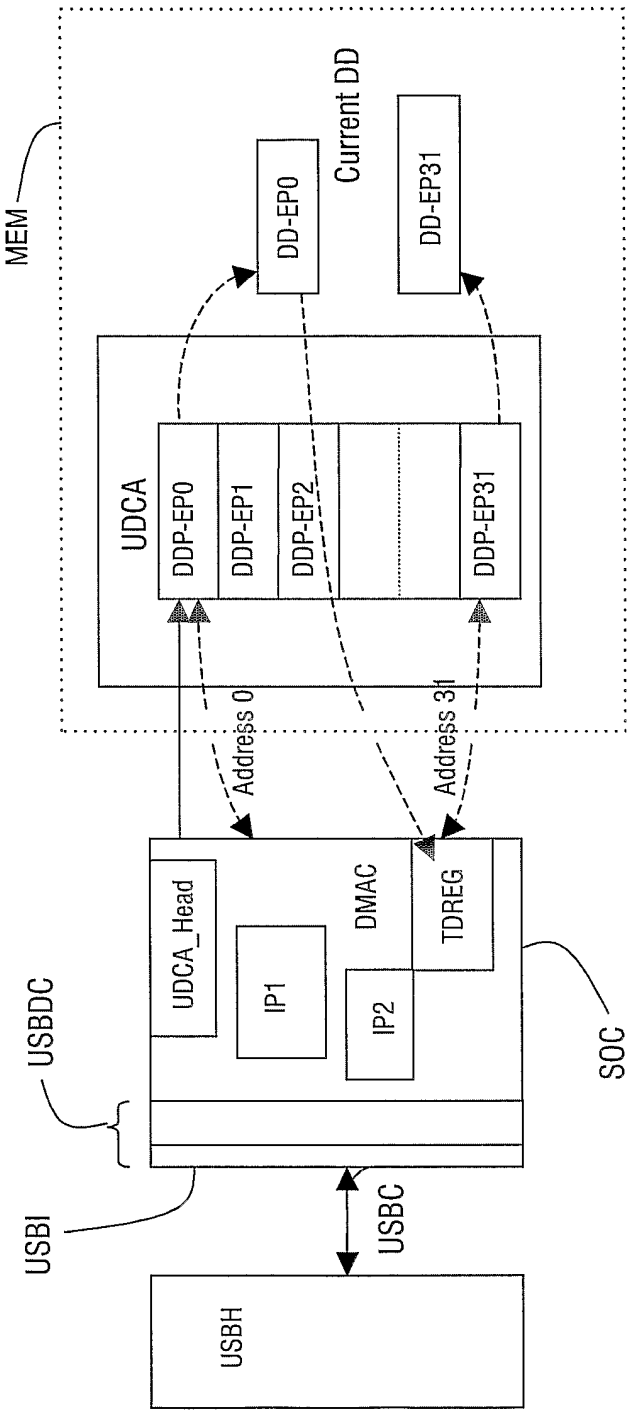

MEM
Current DD
DD-EP0
DD-EP31
UDCA
DDP-EP0
DDP-EP1
DDP-EP2
DDP-EP31
Address 0
Address 31
UDCA_Head
IP1
DMAC
IP2
TDREG
SOC
USBDC
USBI
USBC
USBH

FIG.4

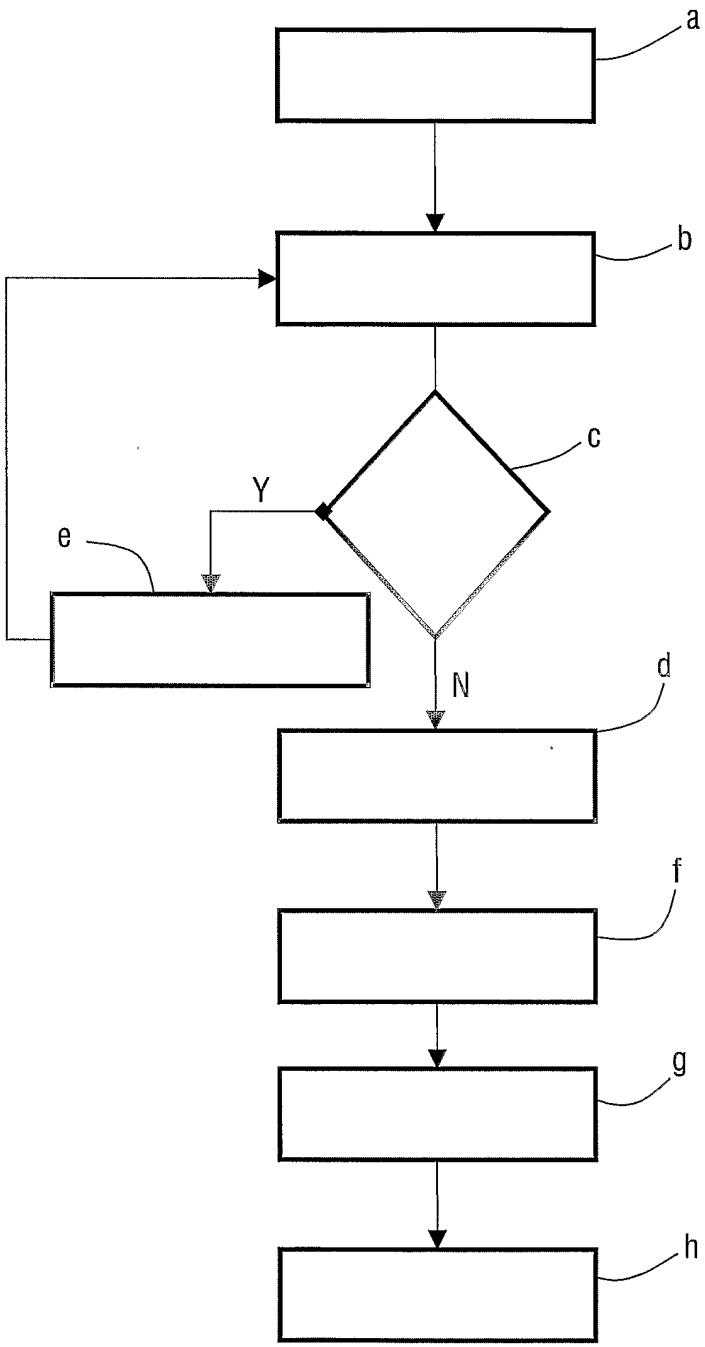


FIG.5

INTERNATIONAL SEARCH REPORT

International Application No
PCT/IB2004/050326

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F13/28

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)
IPC 7 G06F

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 practical, search terms used)

EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5 828 903 A (SADGER HAIM ET AL) 27 October 1998 (1998-10-27) column 1, line 18 - line 41 column 2, line 23 - line 40 column 4, line 26 - line 44 column 5, line 9 - line 39 column 5, line 59 - column 6, line 5 column 6, line 30 - line 50 figure 4 ----- -/--	1-6

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in 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 document 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	Date of mailing of the international search report
29 July 2004	05/08/2004

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31–70) 340–2040, Tx. 31 651 epo nl, Fax: (+31–70) 340–3016	Braccini, G

INTERNATIONAL SEARCH REPORT

International Application No
PCT/IB2004/050326

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	"USB UNIVERSAL SERIAL BUS SPECIFICATION VERSION 1.0" UNIVERSAL SERIAL BUS (USB), XX, XX, 15 January 1996 (1996-01-15), pages 1-268, XP002917782 page 181, paragraph 9.5 page 187, paragraph 9.6.4 page 197, line 21 - line 27 page 209, paragraph 10.5.2.3	3
A	US 6 266 715 B1 (LOYER BRUCE A ET AL) 24 July 2001 (2001-07-24) column 4, line 44 - line 60 column 6, line 49 - line 56 column 10, line 17 - line 39 figure 2	1-4
A	US 6 182 165 B1 (SPILO DAVID A) 30 January 2001 (2001-01-30) column 6, line 16 - line 39 figure 3	1-4

INTERNATIONAL SEARCH REPORT

nformation on patent family members

Int[redacted]tional Application No
PCT/IB2004/050326

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5828903 A	27-10-1998	NONE	
US 6266715 B1	24-07-2001	WO 9963448 A1	09-12-1999
US 6182165 B1	30-01-2001	NONE	