

(19) World Intellectual Property Organization  
International Bureau



(43) International Publication Date  
25 June 2009 (25.06.2009)

PCT

(10) International Publication Number  
**WO 2009/077341 A1**

(51) International Patent Classification:  
**G06F 13/28** (2006.01)

(21) International Application Number:  
PCT/EP2008/066781

(22) International Filing Date:  
4 December 2008 (04.12.2008)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:  
0724439.5 14 December 2007 (14.12.2007) GB

(71) Applicant (for all designated States except US): **ICERA INC** [US/US]; Corporation Trust Center, 1209 Orange Street, Wilmington, New Castle, Delaware (US).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **MORRIS, Matthew** [GB/GB]; Flat 3, 4 Trelawney Road, Cotham, Bristol BS6 6EA (GB). **BOND, Andrew** [GB/GB]; Flat 27, Oswalds Court, Oswald Road, Redland, Bristol BS6 7HX (GB).

(74) Agent: **PAGE WHITE & FARRER**; Bedford House, John Street, London Greater London WC1N 2BF (GB).

(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, BR, BW, BY, BZ, CA, CH, 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, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

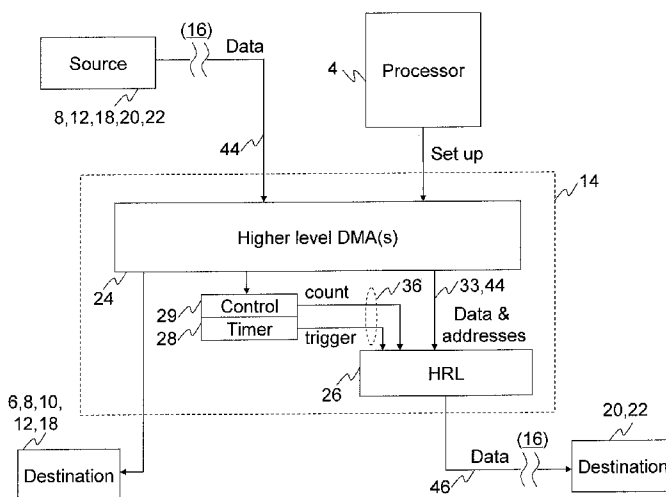
(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, NO, 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

(54) Title: DMA DATA TRANSFER

Figure 2



(57) Abstract: An integrated circuit comprising: a plurality of on-chip devices; a DMA engine; and a CPU for executing code to set up the data transfer engine to perform a transfer, the set-up comprising indicating the address of a source and destination device. The integrated circuit also comprises timing means arranged to generate a trigger at a time after the execution of the set-up code; and transfer control means arranged to determine, at that time, an amount of data to be transferred. The DMA engine is arranged to receive the trigger from the timing means and an indication of the amount from the transfer control means, and to transfer that amount of data to the destination peripheral interface in response to the trigger.

## DMA DATA TRANSFER

Field of the Invention

- 5     The present invention relates to a data transfer engine for transferring data in a computer system.

Background

- 10     Direct Memory Access (DMA) refers to a feature in a computer system whereby data can be transferred directly between memory devices and/or memory-mapped peripheral devices without that data needing to pass via the Central Processing Unit (CPU).
- 15     Although DMA is now commonplace, without DMA the CPU would have to read data from the destination device into one or more of the CPU's operand registers, and then write that data from its operand registers to the destination device. This would be wasteful of processor resources, especially where several bytes are to be transferred, because the CPU would have to be occupied throughout the
- 20     entire transfer.

But using DMA, software running on the CPU simply sets up the DMA engine to transfer the data directly by programming it with the source address, destination address and the amount of data to be transferred. After the set up, the CPU can

25     then continue with other tasks whilst the DMA engine completes the transfer independently of the CPU.

However, the fact that the DMA transfer is set up in advance can introduce timing difficulties.

## Summary

To set up a DMA, the software programming the DMA must conventionally provide it with the relevant set-up information. However, although this information  
5 may be correct at the point in time when the DMA is set up, it could change between the point of set-up and the point at which the data is actually transferred. For example, the source may have additional data by that time or the destination may have reduced the amount of storage available for receiving the data.

10 According to one aspect of the present invention, there is provided an integrated circuit chip comprising: a plurality of addressable on-chip devices; a DMA data transfer engine for transferring data between the devices; and a central processing unit for executing transfer set-up code to set up the data transfer  
15 engine to perform a transfer, the set-up comprising indicating to the data transfer engine the address of a source device and the address of a destination device from said plurality of devices; timing means arranged to generate a trigger at a time after the execution of said transfer set-up code; and transfer control means arranged to determine, at said time, an amount of data to be transferred; wherein the DMA engine is arranged to receive said trigger from the timing means and an  
20 indication of said amount from the transfer control means, and to transfer said amount of data to the destination peripheral interface in response to said trigger.

Thus by generating a determination of the amount of data together with the trigger to start the DMA transfer, the present invention allows the amount of data  
25 to transfer to be determined at the point of performing of the transfer itself, and not at the point of DMA set up. The amount of data to transfer need not be indicated at the point of programming the DMA, so the software does not have to set up the amount in advance.

In a particularly advantageous application of the present invention, the devices include at least one peripheral interface to an external peripheral, and the destination device is one of the peripheral interfaces.

- 5     The inventor has recognised that the above timing issue is particularly problematic in the case of driver software which writes data to external peripherals. Such software typically only has a limited amount of time running on the CPU and so needs to set up transfers in advance and then allow hardware timers to determine when those transfers occur.

10

In embodiments, the destination peripheral interface may be an RF interface for use in wireless communications. The RF interface may be configured for communicating via a wireless cellular network.

- 15     The above timing issue may be particularly problematic in the case of RF driver software, especially for cellular communications, because of the continuous demands for output on the peripheral relative to the amount of processor time typically scheduled for the RF driver. Thus the present invention has a particularly advantageous application to wireless communications.

20

In further embodiments, the transfer control means may be configured to determine said amount in dependence on how much data is in the source device waiting to be transferred.

- 25     The transfer control means may be configured to determine said amount in dependence on space available to accept data in one or more registers of the destination peripheral interface.

- 30     Said timing means may be arranged to determine said time in dependence on an external timing event.

Said external timing event may be generated by a launching peripheral, other than the destination peripheral and a peripheral associated with the source device, and said control means may be arranged to determine said amount in dependence on an indication received from the launching peripheral

5

The timing means may be arranged to arbitrate between the timings of said transfer and at least one other transfer, and to generate said trigger in dependence on said arbitration.

- 10 Said timing means may be arranged to determine said time in dependence on a time specified by the central processing unit in said set up.

The data transfer engine may comprise a first DMA stage and a second DMA stage, and the first DMA stage may be arranged to supply data from the source device to the second DMA stage.

15

The invention is particularly but not exclusively advantageous for DMAs that are timed to start at an external timing event unknown to either the source or destination device. Further, the number of bytes to transfer may be generated by a launching peripheral which may be neither the source nor destination device.

20

According to another aspect of the present invention, there is provided a method of transferring data in an integrated circuit chip, the method comprising: executing transfer code to set up a DMA data transfer engine to perform a transfer, the set up comprising indicating to the data transfer engine the addresses of a source and destination device from a plurality of addressable on-chip devices; determining a time after the execution of said transfer code at which the transfer should occur and generating a trigger at said time; at said time, determining an amount of data to transfer; supplying said trigger and an indication of said amount to the data transfer engine; and transferring using the

25

30

DMA engine to transfer said amount of data from the source device to the destination device in dependence on receipt of said trigger by the DMA engine.

For a better understanding of the present invention and to show how the same may be carried into effect, reference will now be made by way of example to the accompanying drawings.

#### Brief Description of the Drawings

Figure 1 is a schematic block diagram of soft-modem computer system, Figure 2 is a schematic clock diagram of a DMA data transfer engine, and Figure 3 is a schematic block diagram of a lower level of a DMA engine.

#### Detailed Description of Preferred Embodiments

Figure 1 schematically illustrates an example of an integrated circuit package for use in a mobile terminal such as a mobile phone. The circuit 2 comprises a central processing unit (CPU) 4 to which is connected an instruction memory 10, a data memory 12, an instruction cache 6, and a data cache 8. Each of the instruction memory 10, data memory 12, instruction cache 6 and data cache 8 are connected to a direct memory access (DMA) data transfer engine 14, which in turn is connected to a system interconnect 16 comprising a data bus and an address bus.

The system interconnect 16 connects between the DMA data transfer engine 14, a memory controller 18, and various on-chip devices in the form of peripheral interfaces 20 and 22 which connect to external devices, i.e. external to the integrated circuit 2. The memory controller 18 connects to one or more external memory devices (not shown). For example, the memory controller 18 may support a connection to RAM such as SDRAM or mobile DDR, to flash memory such as NAND flash or NOR flash, and/or to a secure ROM. Examples of

peripheral interfaces include an analogue radio frequency (RF) interface 22 and one or more additional peripheral interfaces 20. Each of the one or more additional peripheral interfaces 20 connects to a respective external peripheral (also not shown). For example, the peripheral interfaces 20 may include a USIM  
5 interface 20a, a power management interface 20b, a UART interface 20c, an audio interface 20d, and/or a general purpose I/O interface 20e. The RF interface 22 connects with an external RF front-end and antenna (also not shown), and ultimately with a wireless cellular network over an air interface. In the case where there are a plurality of peripheral interfaces, some or all of these may be  
10 connected to the system interconnect 16 by a peripheral bus (also not shown).

In a preferred embodiment, the chip used is designed by Icera and sold under the trade name Livanto®. Such a chip has a specialised processor platform described for example in WO2006/117562.

15 In a preferred application of the present invention, the integrated circuit 2 is configured as a software modem, or "soft modem", for handling wireless communications with a wireless cellular network. The principle behind software modem is to perform a significant portion of the signal processing required for the  
20 wireless communications in a generic, programmable, reconfigurable processor, rather than in dedicated hardware.

Preferably, the software modem is a soft baseband modem. That is, on the receive side, all the radio functionality from receiving RF signals from the antenna  
25 up to and including mixing down to baseband is implemented in dedicated hardware. Similarly, on the transmit side, all the functionality from mixing up from baseband to outputting RF signals to the antenna is implemented in dedicated hardware. However, all functionality in the baseband domain is implemented in software stored in the instruction memory 10, data memory 12 and external  
30 memory, and executed by the processor 4.

In a preferred implementation, the dedicated hardware in the receive part of the RF interface 22 may comprise a low noise amplifier (LNA), mixers for downconversion of the received RF signals to intermediate frequency (IF) and for downconversion from IF to baseband, RF and IF filter stages, and an analogue to digital conversion (ADC) stage. An ADC is provided on each of in-phase and quadrature baseband branches for each of a plurality of receive diversity branches. The dedicated hardware in the transmit part of the RF interface 22 may comprise a digital to analogue conversion (DAC) stage, mixers for upconversion of the baseband signals to IF and for upconversion from IF to RF, RF and IF filter stages, and a power amplifier (PA). Optionally, some of these stages may be implemented in an external front-end (in which case the RF interface may not input and output RF signals per se, but is still referred to as an RF interface in the sense that it is configured to communicate up/downconverted or partially processes signals with the RF front-end for the ultimate purpose of RF communications). The "peripheral" to the RF interface is the antenna and any associated front-end required external to the chip 2. Details of the required hardware for performing such radio functions will be known to a person skilled in the art.

Received data is passed from the RF interface 22 to the processor 4 for signal processing, via the system interconnect 16, DMA data transfer engine 14 and data memory 12. Data to be transmitted is passed from the processor 4 to the RF interface 22 via the data memory 12, DMA data transfer engine 14 and system interconnect 16.

The software modem running on the processor 4 may then handle functions such as:

- Modulation and demodulation,
- Interleaving and de-interleaving,
- Rate matching and de-matching,
- Channel estimation,



- Equalisation,
- Rake processing,
- Bit log-likelihood ratio (LLR) calculation,
- Transmit diversity processing,
- 5 - Receive diversity processing,
- Multiple-Input Multiple-Output (MIMO) processing,
- Voice codecs,
- Link adaptation by power control or adaptive modulation and coding, and/or
- Cell measurements.

10

The DMA data transfer engine 14 is now discussed in more detail in relation to Figure 2. In embodiments, the data transfer engine 14 comprises a plurality of different hierarchical stages of DMA engine: a lower level DMA engine 26 referred to herein as HRL (Hardware Regulated Latency), and one or more

15 higher level DMA engines 24. The higher level DMA engine(s) 24 are arranged to receive data from any of the data cache 8, data memory 12, memory controller 18, RF interface 22 and additional peripheral interfaces 20 (via the system interconnect 16 if necessary); and to write data to the instruction cache 6, instruction memory 10, data cache 8, data memory 12 and memory controller 18.

- 20 The lower level HRL DMA engine 26 is an "add on" arranged specifically to write data to memory-addressable registers of the peripheral interfaces 20 and 22 via the system interconnect 16, i.e. to peripheral interfaces rather than storage memories. The data transfer engine 14 also comprises a timer 28 and transfer controller 29 connected to the DMA levels 24 and 26. The operation of the timer
- 25 28 and controller 29 is discussed below.

The structure is hierarchical in that the data buffers of a lower level DMA engine 26 are fed by a higher level DMA 24 engine.

- 30 In operation, the CPU 4 executes code which sets up a DMA transfer by writing a source and destination address to registers of a higher level DMA engine 24,

along with any timing conditions associated with the transfer. The CPU 4 may set up a number of such transfers between any of the different memory addressed devices 6, 8, 10, 12, 18, 20 and 22. These transfers may be timed to occur at certain times under the control of the timer 28, being triggered for example by an

external timing event or the elapsing of a certain predetermined time period. Further, since the DMA engine 14 only has a limited number of channels, the timings of such transfers may potentially conflict with one another and thus the timer 28 may also be configured to arbitrate between the timings of the transfers, for example based on the relative lateness of the transfers and/or according to a

The issue of timing is particularly relevant to driver software for a peripheral, which typically only has a limited amount of time (i.e. processor cycles) running on a CPU and so needs to set up one or more transfers in advance (before other tasks are scheduled, e.g. other driver software for other peripherals). Therefore following the set-up, the hardware timer 28 times when one transfer stops, when data buffers are reprogrammed for a new transfer, and when the new transfer is started. The system timer 28 ensures these register writes happen at the correct time, even though the driver software for that peripheral is no longer currently scheduled and being executed by the CPU 4.

RF drivers for the RF interface 22 of a soft modem, especially for wireless cellular communications, are particularly susceptible to these difficulties because of the continual demands for output via the RF interface 22 (paging, hand-over, cell measurements, voice data, etc.) relative to the amount of time scheduled for the RF driver on the CPU 4.

Conventionally the set up by the CPU 4 would also have to include writing an indication of the number of bytes to be transferred to the higher level DMA engine 24. However, as mentioned, circumstances may change between the time the transfer is set up and the time it is actually carried out. For example, the source

may have additional data to transfer or the destination may have changed the amount of storage available to receive the transferred data.

This issue is particularly (but not exclusively) important for DMA transfers that are  
5 timed to start at an external timing event generated by an external peripheral, unknown to either the source or destination peripheral interface, because the timing of the event cannot be known relative to the scheduling of the driver.

Accordingly, embodiments of the present invention are provided with transfer  
10 control logic 29 which is configured to determine the amount of data to be transferred, with the determination being performed at the point in time at which the transfer actually takes place rather than when it is set up by the CPU 4. (Of course, this may not be achieved at the exact moment of the transfer, but the point is that it is performed in association with the transfer rather than the set-up).

15 The timer 28 supplies the trigger and the controller 29 supplies an indication of the number of bytes determined to the HRL 26, which writes that number of bytes to the peripheral 20 or 22 in question in dependence on receipt of the trigger.

The control logic 29 could be configured to make the determination based on the  
20 availability of data at the source or on the space available at the time of the transfer. The amount could even be determined based on an input from a launching peripheral other than the source and destination peripheral.

The HRL 26 and its interface with a higher level DMA 24, timer 28 and controller  
25 29 are now discussed in further detail in relation Figure 3.

The HRL 26 comprises an address decoder 32 having an input 33 connected to the higher level DMA engine 24. The HRL 26 further comprises a plurality N of queue blocks 30(1)...30(N), each block having a respective data queue  
30 comprising a set of first-in-first-out (FIFO) data buffers 40(1)...40(N) and a respective corresponding address queue comprising a set of FIFO address

buffers 42(1)...42(N). Each data queue 40(1)...40(N) and address queue 42(1)...42(N) has a respective input connected to the address decoder 32. For each data queue 40(1)...40(N), the address decoder 32 is also provided with a route 44(1)...44(N) to retrieve data from a source device 8, 12, 18, 20, 22 via the higher level DMA 24 and pass it to the respective data queue 40(1)...40(N). The HRL 26 further comprises a round-robin arbitration multiplexer 38, with an output of each of the data queues 40(1)...40(N) and address queues 42(1)...42(N) being connected to a respective input of the multiplexer 38. The multiplexer 38 has an output 46 connected to the system interconnect 16.

In addition, each queue block 30(1)...30(N) comprises a respective counter 34(1)...34(N), each with an output connected to a respective control input of the multiplexer 38. Each counter 34(1)...34(N) also has an input connected to the timer 28 and controller 29 by a respective control bus 36(1)...36(N) referred to herein as a SIC (Simple InterConnect) interface or bus. Each SIC control bus 36(1)...36(N) preferably comprises a single trigger wire from the timer 28 and a seven-bit wide count bus from the controller 29. In embodiments, it is this control interface 36 which advantageously allows the timing of a DMA write to a peripheral interface to be detached from the timing of the set-up of that transfer by the CPU 4.

In operation, the higher level DMA engine 24 passes the source and destination addresses to the address decoder 32 of the HRL 26b. The address decoder 32 finds a free queue block 30, each block 30 being for transfer to a different destination, and passes the destination addresses into the address queue 42 of that block. The address decoder 32 also uses the source address to request the corresponding source data from the source device 8, 12, 18, 20 or 22 via route 44 through the higher DMA engine 24, and passes the retrieved data to the data queue 40. Each entry in the data queue is preferably a thirty-two bit word wide. Thus the queues store pairs of data words and corresponding destination

addresses for writing the data to destination devices, fed by the higher level DMA engine 24.

5 If there is no data setup in the queues 40, then the HRL waits for the data. The assumption is that data will be available before the trigger starts the HRL writing to the destination, but this is not mandatory. If the queues 40 are empty then the DMA request signals to the higher DMA engine will be asserted so data should be send down to the HRL soon after.

10 When the timer 28 determines that it is time for a write to a certain peripheral to occur, as discussed above, it supplies a trigger signal to the counter 34 of the appropriate queue block 30 via the trigger wire of the corresponding SIC control bus 36. Along with the trigger, the control logic 29 also supplies a count of the number of bytes to be transferred at the time at which the trigger is generated.

15 The counter 34 then counts out that number of data bytes from the data queue 40, paired with its corresponding destination address.

The round-robin arbitration multiplexer 38 outputs the data bytes and corresponding destination addresses onto the data and address buses of the

20 system interconnect 16, cycling in a round-robin manner between the outputs of whichever queue blocks 30 have data waiting to output.

An example use the HRL is where the SIC signal is generated from a peripheral called the Cellular Timer (CET) that is in a different clock domain. The source is

25 actually a CPU write (instead of writing to the destination peripheral directly, the CPU writes the address and data to the HRL queues instead). The destination is the RF interface FIFO configuration register. Neither the CPU or RF interface know when the write will be scheduled so it's setup in advance and the CET signals to perform the write at the correct time. The CET is asynchronous to the

30 target and destination devices.

It will be appreciated that the above embodiments are described only by way of example. For example, it is not necessary to use different levels of DMA, and the timer and transfer controller could be used to control transfers in a single DMA engine. Further, transfers could be triggered by other timing events and/or the amount of data to be transferred could be determined based on other criteria. Other variations and uses of the present invention may be apparent to a person skilled in the art given the disclosure herein. The scope of the invention is not limited by the described embodiments, but only by the following claims.

Claims

1. An integrated circuit chip comprising:  
a plurality of addressable on-chip devices;  
5 a DMA data transfer engine for transferring data between the devices; and  
a central processing unit for executing transfer set-up code to set up the  
data transfer engine to perform a transfer, the set-up comprising indicating to the  
data transfer engine the address of a source device and the address of a  
destination device from said plurality of devices;  
10 timing means arranged to generate a trigger at a time after the execution  
of said transfer set-up code; and  
transfer control means arranged to determine, at said time, an amount of  
data to be transferred;  
wherein the DMA engine is arranged to receive said trigger from the timing  
15 means and an indication of said amount from the transfer control means, and to  
transfer said amount of data to the destination peripheral interface in response to  
said trigger.
2. The device according to claim 1, wherein the devices include at least one  
20 peripheral interface to an external peripheral, and the destination device is one of  
the peripheral interfaces.
3. The circuit according to claim 2, wherein the destination peripheral  
interface is an RF interface for use in wireless communications.  
25
4. The circuit according to claim 3, wherein the RF interface is configured for  
communicating via a wireless cellular network.
5. The circuit according to any preceding claim, wherein the transfer control  
30 means is configured to determine said amount in dependence on how much data  
is in the source device waiting to be transferred.

6. The circuit according to any preceding claim, wherein the transfer control means is configured to determine said amount in dependence on space available to accept data in one or more registers of the destination peripheral interface.

5

7. The circuit according to preceding claim, wherein said timing means is arranged to determine said time in dependence on an external timing event.

8. The circuit according to claim 7, wherein said external timing event is generated by a launching peripheral, other than the destination peripheral and a peripheral associated with the source device, and said control means is arranged to determine said amount in dependence on an indication received from the launching peripheral

10

9. The circuit according to any preceding claim, wherein the timing means is arranged to arbitrate between the timings of said transfer and at least one other transfer, and to generate said trigger in dependence on said arbitration.

15

10. The circuit according to any preceding claim, wherein said timing means is arranged to determine said time in dependence on a time specified by the central processing unit in said set up.

20

11. The circuit according to any preceding claim wherein the data transfer engine comprises a first DMA stage and a second DMA stage, the first DMA stage being arranged to supply data from the source device to the second DMA stage.

25

12. A method of transferring data in an integrated circuit chip, the method comprising:

executing transfer code to set up a DMA data transfer engine to perform a transfer, the set up comprising indicating to the data transfer engine the

30



addresses of a source and destination device from a plurality of addressable on-chip devices;

determining a time after the execution of said transfer code at which the transfer should occur and generating a trigger at said time;

5 at said time, determining an amount of data to transfer;

supplying said trigger and an indication of said amount to the data transfer engine; and

transferring using the DMA engine to transfer said amount of data from the source device to the destination device in dependence on receipt of said trigger  
10 by the DMA engine.

13. The method of claim 12, wherein the devices include at least one peripheral interface to an external peripheral, and the destination device is one of the peripheral interfaces.

15

14. The method of claim 13, wherein the destination peripheral interface is an RF interface for use in wireless communications.

15. The method of claim 14, comprising using the RF interface to  
20 communicate via a wireless cellular network.

16. The method of any of claims 12 to 15, wherein the determining of said amount comprises determining the amount in dependence on how much data is in the source device waiting to be transferred.

25

17. The method of any of claims 12 to 16, wherein the determining of said amount comprises determining the amount in dependence on space available to accept data in one or more registers of the destination peripheral interface.

18. The method according to any of claims 12 to 17, wherein the determining of said time comprises determining the time in dependence on an external timing event.

5 19. The method of any of claims 12 to 18, comprising generating said external timing event using a launching peripheral, other than the destination peripheral and a peripheral associated with the source device, wherein the determining of said amount comprises determining the amount in dependence on an indication received from the launching peripheral.

10

20. The method of any of claims 12 to 19, comprising arbitrating between the timings of said transfer and at least one other transfer, and generating said trigger in dependence on said arbitration.

15 21. The method of any of claims 12 to 20, wherein the determining of said time comprises determining the time in dependence on a time specified by the central processing unit in said set up.

20 22. The method of any preceding claim, wherein the data transfer engine comprises a first DMA stage and a second DMA stage, and the method comprises supplying data from the source device to the second DMA stage.

Figure 1

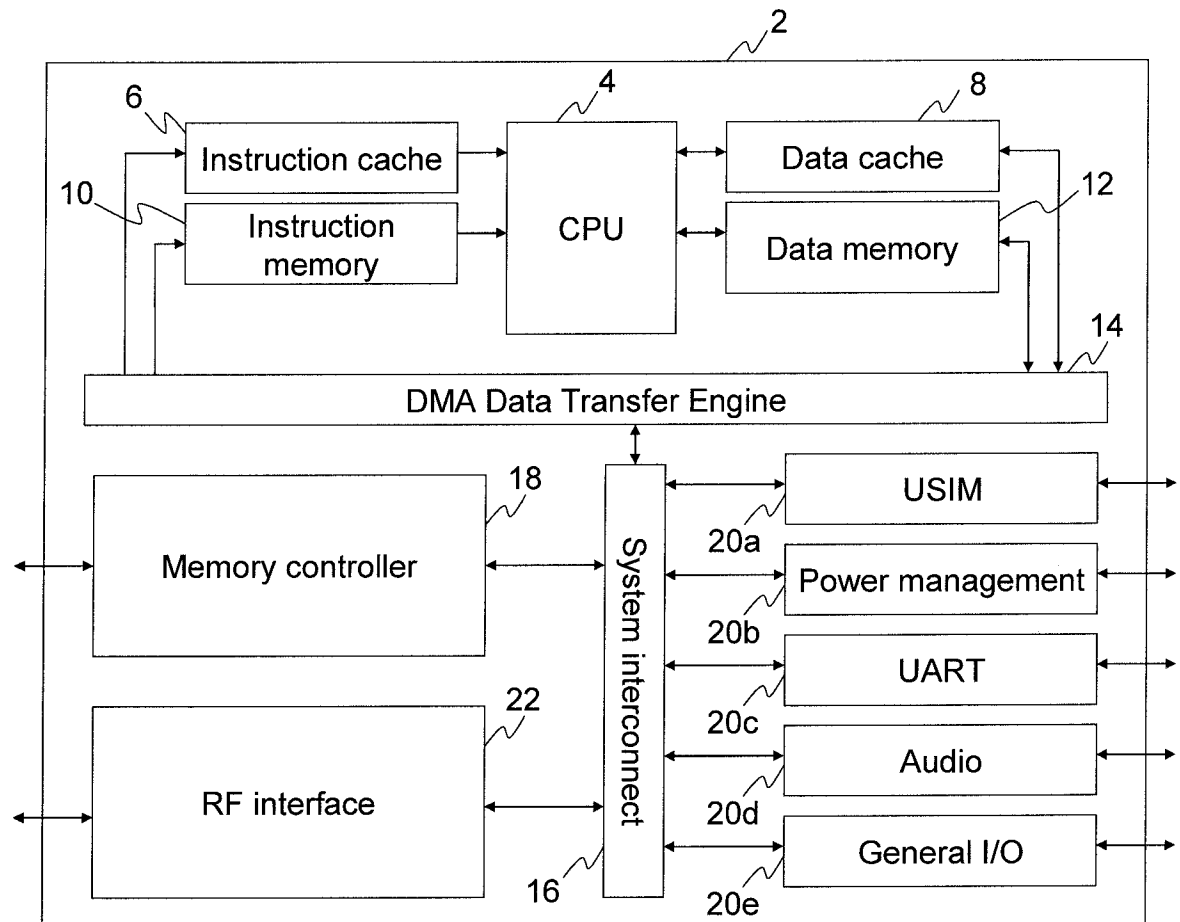


Figure 2

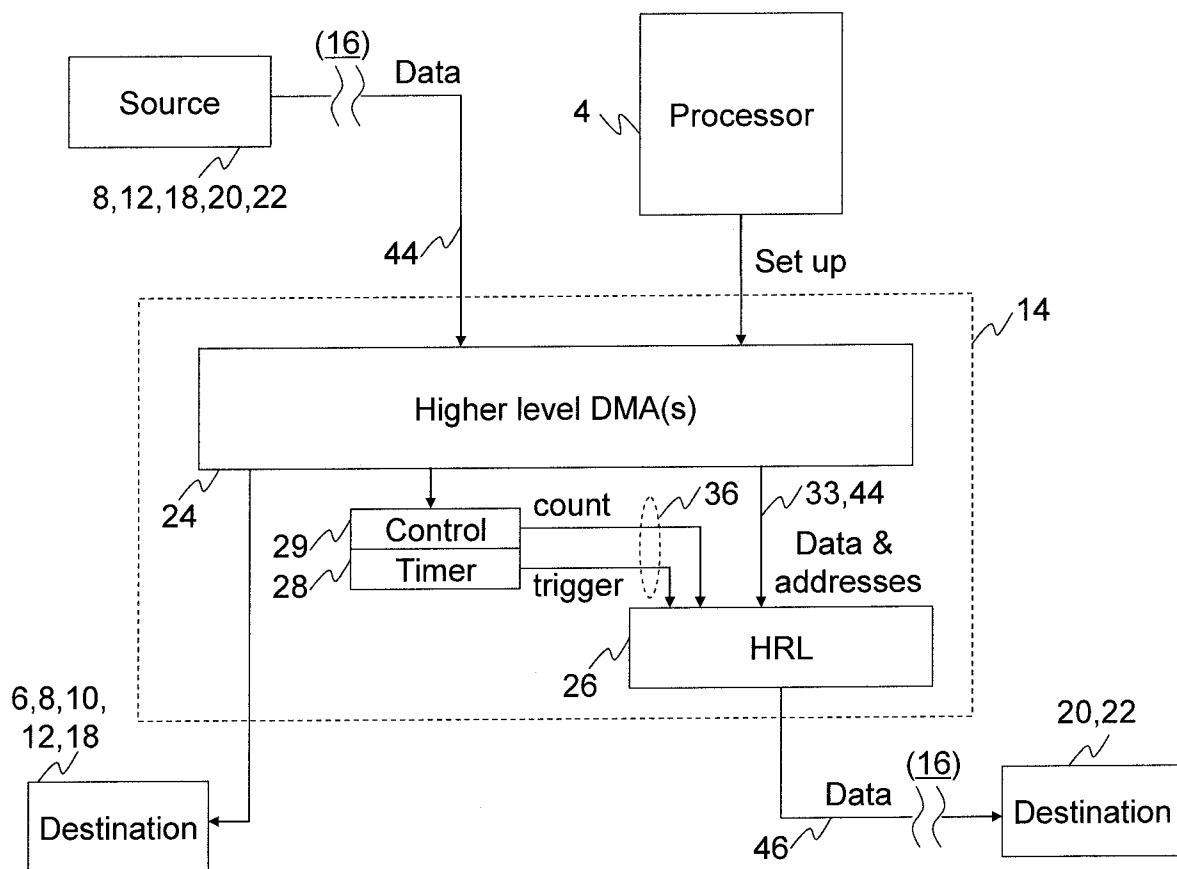
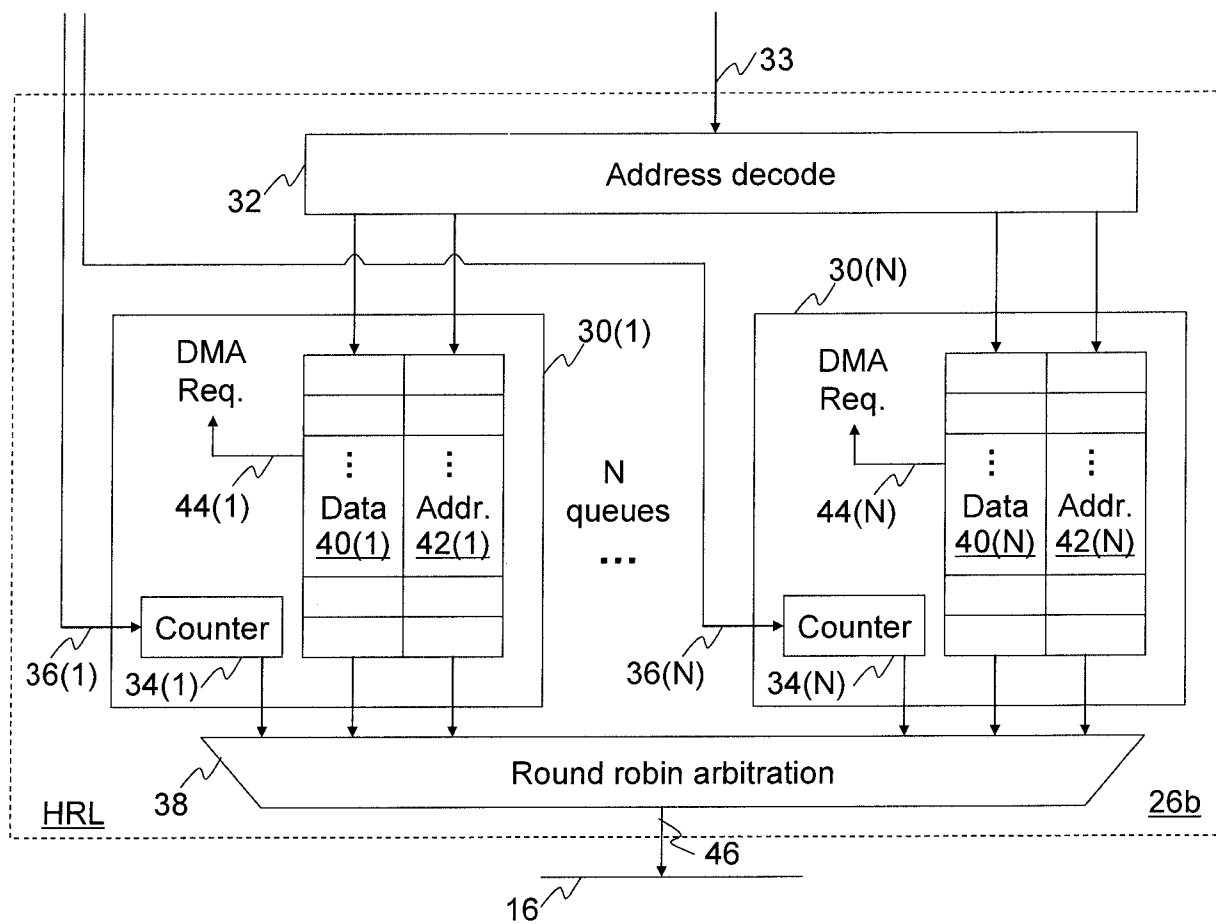


Figure 3



# INTERNATIONAL SEARCH REPORT

International application No

PCT/EP2008/066781

## A. CLASSIFICATION OF SUBJECT MATTER

INV. 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)

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 2006/010264 A1 (RADER SHEILA M [US] ET AL) 12 January 2006 (2006-01-12)	1-4, 7-15, 18-22
Y	abstract; figures 1-6	5,6,16, 17
	paragraphs [0025] - [0042], [0063] - [0076]	
Y	US 5 905 911 A (SHIMIZU TOSHIYUKI [JP]) 18 May 1999 (1999-05-18)	5,6,16, 17
	abstract; figures 1,6	
	column 1, line 32 - column 2, line 63	
	column 3, line 42 - column 5, line 49	
A	EP 0 997 822 A (NIPPON ELECTRIC CO [JP]) 3 May 2000 (2000-05-03)	5,6,16, 17
	abstract; figures 1,6	
	paragraphs [0001], [0016] - [0038]	

☐ 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 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

6 April 2009

Date of mailing of the international search report

21/04/2009

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2  
NL - 2280 HV Rijswijk  
Tel. (+31-70) 340-2040,  
Fax: (+31-70) 340-3016

Authorized officer

París Martín, Laura

# INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/EP2008/066781

Patent document cited in search report		Publication date	Patent family member(s)	Publication date
US 2006010264	A1	12-01-2006	NONE	
US 5905911	A	18-05-1999	NONE	
EP 0997822	A	03-05-2000	CN 1252566 A	10-05-2000
			JP 3206568 B2	10-09-2001
			JP 2000132497 A	12-05-2000