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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
16 August 2007 (16.08.2007)

PCT

(10) International Publication Number
WO 2007/091092 A1

(51) International Patent Classification:
G06F 9/318 (2006.01) *G06F 9/38* (2006.01)

(21) International Application Number:
PCT/GB2007/000473

(22) International Filing Date: 9 February 2007 (09.02.2007)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
0602730.4 10 February 2006 (10.02.2006) GB

(71) Applicants *(for all designated States except US)*: **IMAGINATION TECHNOLOGIES LIMITED** [GB/GB]; Home Park Estate, Kings Langley, Hertfordshire WD4 8LZ (GB). **IMAGINATION TECHNOLOGIES LIMITED** [GB/GB]; Home Park Estate, Kings Langley, Hertfordshire, WD4 8LZ (GB).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **WEBBER, Andrew** [GB/GB]; 17 Fields End, Tring, Hertfordshire, HP12 5ER (GB). **WEBBER, Andrew** [GB/GB]; 17 Fields End, Tring, Hertfordshire, HP12 5ER (GB).

(74) Agent: **ROBSON, Aidan, John**; Reddie & Grose, 16 Theobalds Road, London WC1X 8PL (GB).

(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, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, 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, 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, HU, IE, IS, IT, LT, LU, LV, 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.

WO 2007/091092 A1

(54) Title: INSTRUCTION SETS FOR MICROPROCESSORS

(57) Abstract: A method and apparatus are provided for selecting between a plurality of instruction sets available to a microprocessor. An instruction fetch address is supplied. At least one predetermined bit of the instruction fetch address is used to select between the instruction sets. Once an instruction set has been selected instructions may be fetched and decoded with a decoding scheme appropriate to the instruction set.

Instruction Sets for Microprocessors

This invention relates to a microprocessor and to instruction sets which may be used by such a processor, and particularly to multithreaded processors and their instruction sets.

Background to the Invention

In our British patent no. GB-A-2311882 there is described a multithreaded processor. In this, a single processing unit has a plurality of inputs and outputs, corresponding to a plurality of processing threads which are to execute on the processor. The processor arbitrates between the threads to determine which one should be executed on each block cycle. This is typically done on a prioritisation basis. Further developments of this have concerned monitoring factors such as the time since execution commenced for a thread, and the time to a specific deadline by when the thread must execute. This idea can be embodied in processors directed to general processing but also in application specific processors such as Digital Signal Processors (DSP).

A number of different threads can be arranged to execute on one of these processors, but DSPs typically use between two and four threads. The number of threads is defined at the design and manufacture stage of the chip and the chip is configured with an appropriate number of inputs and outputs.

A typical processor uses a 32 bit instruction set which may be extended via template instructions which are used to retrieve additional instructions.

Some processors (not multithreaded) have been produced with smaller than standard instruction sets. This leads to a reduction in the code size of a program used by such a processor. The processor will be configured to switch between the large and reduced instruction sets using special instructions. Each time a new instruction set is added additional switching instructions have to be added to enable it to be accessed.

Many of the applications to which multithreaded processors could be put include embedded and low power requirements. Such requirements constrain the amount of memory available in the systems for data such as programs. As a

result, microprocessor manufacturers have aimed to improve their devices by compressing program code. This is most commonly achieved by supporting instruction subsets which can be implemented when a smaller instruction set is required. For example, a processor with a 32 bit instruction set may also be able to support a special 16 bit instruction set to allow programs to be made smaller.

Again, the switch between such instruction sets is handled by use of a special instruction to switch between the sets. Again this requires a special instruction and an additional clock cycle to perform the switching.

Summary of the Invention

Preferred embodiments of the present invention provide a processor which is able to support more than one instruction set but do not require additional instructions to switch between instructions sets. Accordingly, a specific combination of instruction address bits are used to identify an instruction as belonging to a specific set. As a result, in order to switch between instruction sets the system jumps between different areas of instruction memory to retrieve instructions from the appropriate set.

Preferably at least two instruction sets are provided.

The invention may be embodied in a multithreaded processor.

A Brief Description of the Drawings

A preferred embodiment of the invention is now described in detail by way of example in which:

Figure 1 is a schematic block diagram of an instruction fetch and decode unit for use in an embodiment of the invention:

Figure 2 shows an alternative representation of an instruction fetch and decode unit for use in an embodiment of the invention;

Figure 3 shows the typical processor pipeline for an instruction; and

Figure 4 shows a bit map of the type which is used in an instruction fetch address for use in the arrangements of figure 1 and figure 2.

The processor pipeline for an instruction thread shown in figure 3 comprises five portions. First, an instruction fetch portion 1. This is typically a 32 bit fetch instruction which includes the address of an instruction to be fetched from instruction memory. This is followed by a decode step 2 in which the instruction retrieved from memory is decoded into machine code for execution on the microprocessor. An operand fetch 3 comprises a further fetch. This retrieves any data which is to be operated upon by the fetched and decoded instruction.

At 4, the Arithmetic Logic Unit (ALU) executes the fetched and decoded instruction on the operand and at 5 the result of the executed instruction is written to a destination in memory via the respective processor output.

In an embodiment of the invention, specific bits of the fetch instruction are used to indicate that a different instruction set is to be used. This instruction set will preferably be stored in the same instruction memory as a first instruction set. Two or more instruction sets can be stored and an appropriate number of bits used to select between them. In this example, bits 20 and 23 are used to select between two instruction sets. When both bits 20 and 23 are set to 1, the fetch instruction will retrieve the addressed instruction from the second set of instructions stored in instruction memory rather than the first instruction.

As both instruction sets are stored in instruction memory, the address portion of the fetch instruction defines where in the instruction memory the instructions come from, but the bits 20 and 23 are used when both set to 1 to select between the decoding which is applied to the fetched instruction. Thus when both bits are set to 1, an instruction decoder for the second instruction set is used. Any other combination will result in an instruction decoder for the first set of instructions. Obviously with two bits, a total of four states could be supported and therefore a total of four instruction sets could be selected between using these two bits. Additional bits could be included if further instruction sets are to be used. If only two instruction sets are implemented, then only a single set is required to select between them.

The way in which this fetch address shown in figure 2 affects the operation of the system is now explained with reference to figure 1. Figure 1 comprises an instruction memory 10. Instructions are retrieved from this in response to an

instruction fetch address 12 (as shown in figure 4). The instruction address is supplied to the instruction memory 10 which reads the appropriate instruction out and sends it to a fetched instruction unit 14.

At the same time, an AND gate 16 receives bits 20 and 23 of the instruction. The output of this is supplied to an instruction set type portion of the fetch instruction unit 14. Where more than two bits are used to select between instruction sets, more complex gating will be required, or possibly a multiplexer.

If the output of the AND gate is a 1 then this indicates that bits 20 and 23 are both set to 1 and the instruction set type therefore corresponds to the second of two instruction sets.

The fetched instruction unit 14 sends the fetched instruction to two instruction decoders A and B, 20 and 18 respectively. These simultaneously decode the fetched instruction and give alternative decoded instructions at their outputs.

When the combined address bits indicate that the instruction set type is the first instruction set A, the output of instruction decoder A 20 is required. When the instruction set type indicates that it is the second set of instructions that is required, the output of instruction decoder B 18 is required. Selection between the outputs of instruction decoders A and B, 20 18 is performed in a multiplexer 22. The selection input of this is controlled by an output of the instruction set type portion of the fetched instruction unit 14. As previously explained, instruction set type is determined by the combined address bits from AND gate 16.

The selected decoded instruction is then provided as an output from the multiplexer to the ALU for execution in a conventional manner.

An alternative to the arrangement of figure 1 would be to route only the fetched instruction data to the instruction decoder appropriate to the instruction set type. This would involve positioning a multiplexer between the fetched instruction data unit 14 and the two instruction decoders 20 and 18.

In the implementation shown, the first instruction set would typically contain 32 bit wide instructions from the first set or 16 bit instructions from the second set. After decoding, all of the instructions are in 32 bit wide form, i.e. the 16 bit instructions

are used to form suitable instructions from the main processor instruction set, i.e. a subset of these instructions.

This is explained with reference to figure 4. As can be seen, a specific transformation is performed on the program counter address (PC) to obtain the instruction fetch address. The idea behind this transformation is that the PC addresses are always 32 bit word aligned (bottom bits both zero). If it is decided to switch to a second 16 bit wide instruction it is necessary to use a smaller 2 byte aligned word. This transformation is effectively a logical shift right by 1 of bit 22 a new bit zero being inserted at bit 22. This has the effect of incrementing the program counter by 16 bits at a time instead of 32 bits and thereby enabling 16 bit words to be retrieved from memory.

The arrangement of figure 2 shows an alternative representation of the instruction retrieval and decoding when using multiple instruction sets. The external instruction memory 10 is not shown here. An instruction fetch unit 12 again is used to send instruction fetch requests to the instruction memory 10. The thus retrieved instructions are supplied to a fetched instruction unit 14. The bits 20 and 22 are again combined in an AND gate 16 to determine the instruction set type and to control a multiplexer 22.

A fetched instruction can pass to multiplexer 22 via a short instruction pre-decode unit 24, or directly.

The output of the short instruction set pre-decode unit 24 which is used to operate on a reduced 16 bit instruction set is a 32 bit wide long instruction. Thus both inputs to multiplexer 22 are 32 bits wide. The multiplexer then selects between these and supplies the one determined by the instruction set type to a long instruction pre-decode unit 26 from where it is loaded to a long instruction register 28 and read out to a long instruction post-decode unit 30 from where it can be provided to the ALU for execution. Such an arrangement is used in a system where, for example, a new 16 bit instruction set has been added which has much in common with the subset of the regular 32 bit instruction set. For example, it could be an abridged form of a portion of the instruction set. What is implemented here is the use of one or two 16 bit instruction words to recreate instructions in the regular 32 bit form. For example, the most common

instructions are carefully chosen to fit into a set of instruction patterns that use just one 16 bit instruction word. However, to allow all the required regular 32 bit instructions to be encoded using smaller 16 bit words additional types of 16 bit instruction which can actually use two 16 bit words to form a single instruction are included. These two 16 bit word instructions can in this example have one of two forms. In the first set a second word is used to extend the instructions in a single 16 bit word instruction set. The second form is a new instruction pattern made from two 16 bit words. What is contemplated in the present example is that the 16 bit instruction set is tailored to produce all the bits of the 32 bit instruction set so that a much larger range of instructions can be supported. This is done by allowing more than one 16 bit instruction to be grouped together to form the 32 bit instruction. This is all done based upon the instruction data patterns and is in principle not related to the selection of instruction sets.

There is of course no requirement for a 16 bit instruction set to be a subset of the 32 bit instruction set. Completely unrelated instruction sets can be supported by the invention.

Whilst using such an arrangement ensures that all the instructions actually executed are the 32 bit wide instructions which feed into the standard instruction decoder of the system. However, this arrangement does add to latency since more clock cycles are taken to load instruction, or limits the cycle time.

Using the embodiments described above, it is possible to support a plurality of instruction sets used by a single program by using appropriate ones of the address bits to select between different instruction sets, and therefore to transfer control between different blocks of code.

It is possible for instruction sets to be reduced sets of an overall instruction set, or for instruction sets to be alternative instruction sets which only partially overlap, or in some circumstances may not overlap at all.

Embodiments of the invention can be implemented on a multithreaded microprocessor by appropriate modifications to instruction input pipelines of the types shown in figures 1 and 2.

CLAIMS

1. A method for selecting between a plurality of instruction sets available to a microprocessor comprising the steps of supplying an instruction fetch address, determining the status of at least one predetermined bit of the instruction fetch address, selecting between the instruction sets in dependence on the result of the determination fetching instructions from the selected instruction set, decoding fetched instructions with a decoding scheme appropriate to the instruction set, and supplying the decoded instructions for execution.

2. A method according to claim 1 in which the selecting step comprises supplying the fetched instruction to at least two decoders each having their outputs connected to a multiplexer, and selecting the multiplexer output with the at least one predetermined bit of the instruction fetch address.

3. A method according to claim 1 or 2 in which a first one of the instruction sets has a smaller number of bits than a second one of the instruction sets.

4. A method according to claim 3 in which the first instruction set is a subset of the second instruction set.

5. Apparatus for selecting between a plurality of instruction sets available to a microprocessor comprising

means for supplying an instruction fetch address;

means for determining the status of at least one predetermined bit of the instruction fetch address;

means for selecting between the instruction sets in dependence on the result of the determination;

means for fetching instructions from the selected instruction set;

means for decoding the fetched instructions with a decoding scheme appropriate to the instruction set; and

means for supplying decoded instructions for execution.

6. Apparatus according to claim 5 in which the means for selecting comprises means for supplying the fetched instruction to at least two decoders each having their outputs connected to a multiplexer, the multiplexer having a

select input receiving the at least one predetermined bit of the instruction fetch address and operable to select an output in dependence on the at least one predetermined bit of the instruction fetch address.

7. Apparatus according to claim 5 or 6 in which a first one of the instruction sets has a smaller number of bits than the second of the instruction sets.

8. Apparatus according to claim 3 in which the first instruction set is a subset of the second instruction set.

9. A method for selecting between a plurality of instruction sets available to a microprocessor comprising the steps of supplying an instruction fetch address, determining the status of at least one predetermined bit of the instruction fetch address, selecting between the instruction sets in dependence on the result of the determination fetching instructions from the selected instruction set, decoding fetched instructions with a decoding scheme appropriate to the instruction step, and supplying the decoded instructions for execution, wherein the at least one predetermined bit of the instruction fetch address is positioned part-way through the instruction fetch address.

10. A method according to claim 9 in which the at least one bit of the instruction fetch address is spaced from the least most significant bit and the most significant bit of the instruction fetch address by a plurality of bits.

11. A method according to claim 10 in which the instruction fetch address comprises a 32 bit instruction fetch address and there are two predetermined bits positioned at bits 20 and 23 of the instruction fetch address.

12. Apparatus for selecting between a plurality of instruction sets available to a microprocessor comprising means for supplying an instruction fetch address, means for determining the status of at least one predetermined bit of the instruction fetch address, means for selecting between the instruction sets in dependence on the result of the determination, means for fetching instructions from the selected instruction set, means for decoding the fetched instructions with a decoding scheme appropriate to the instruction set, and means for supplying decoded instructions for execution, wherein the at least one predetermined bit of

the instruction fetch address is spaced positional partway through the instruction fetch address from the least significant bit and most significant bit of the instruction fetch address by a plurality of bits.

13. Apparatus according to claim 12 in which the at least one bit of the instruction fetch address is spaced from the at least significant bit and the most significant bits of the instruction fetch address by a plurality of bits.

14. Apparatus according to claim 13 in which the instruction fetch address bit is a 32 bit word and there are two predetermined bits positioned at bits 20 and 23 of the instruction fetch address.

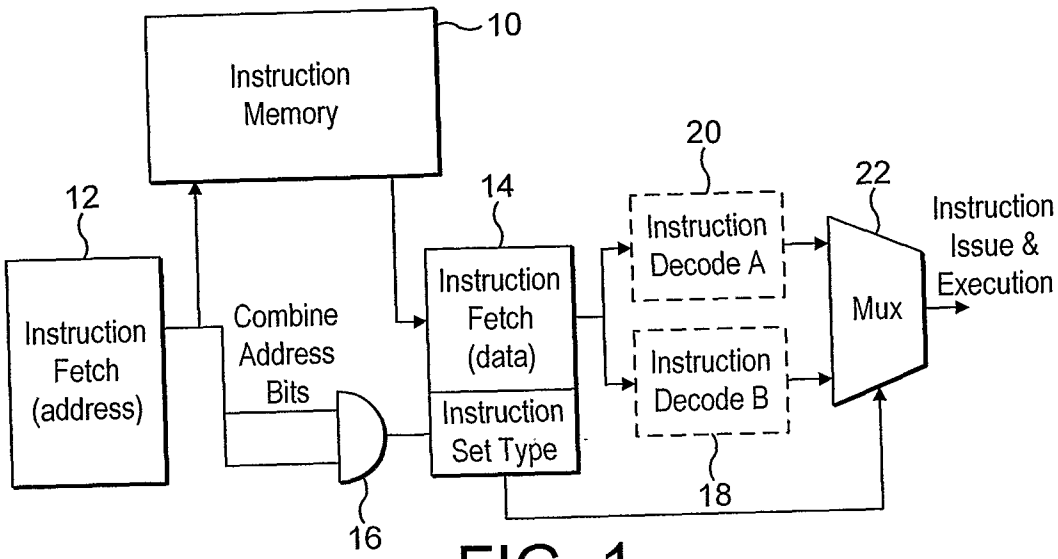
Instruction Memory
10
12
Instruction Fetch (address)
Combine Address Bits
14
Instruction Fetch (data)
Instruction Set Type
16
20
Instruction Decode A
Instruction Decode B
18
22
Mux
Instruction Issue & Execution

FIG. 1

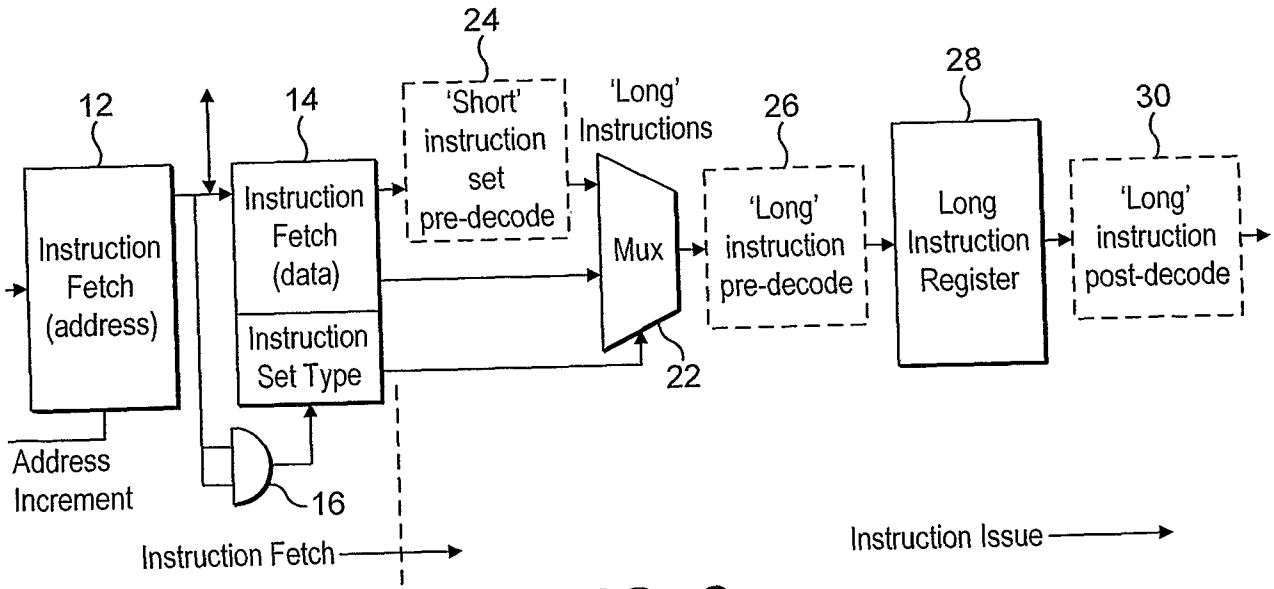
24
'Short' instruction set pre-decode
12
Instruction Fetch (address)
14
Instruction Fetch (data)
Instruction Set Type
Address Increment
16
Instruction Fetch
'Long' Instructions
Mux
22
26
'Long' instruction pre-decode
28
Long Instruction Register
30
'Long' instruction post-decode
Instruction Issue

FIG. 2

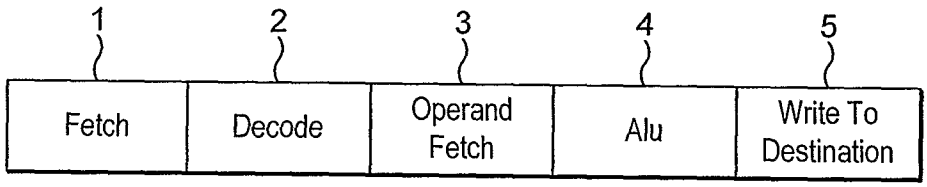
1
2
3
4
5
Fetch
Decode
Operand Fetch
Alu
Write To Destination

FIG. 3

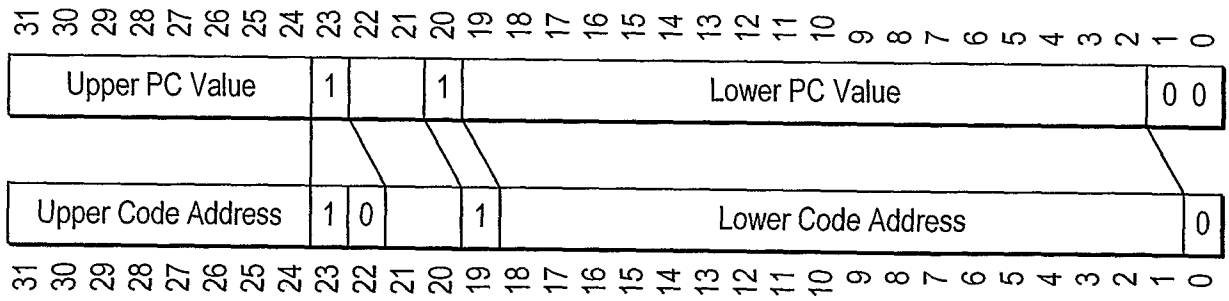
31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Upper PC Value
1
1
Lower PC Value
0 0
Upper Code Address
1
0
1
Lower Code Address
0
31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

FIG. 4

INTERNATIONAL SEARCH REPORT

International application No
PCT/GB2007/000473

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F9/318 G06F9/38

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

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 1 089 167 A (HITACHI LTD [JP]) 4 April 2001 (2001-04-04) paragraph [0025] - paragraph [0027] paragraph [0043] paragraph [0045] paragraph [0050] figures 2,4 claims 2-6	1-14
X	EP 0 324 308 B1 (IBM [US]) 27 March 1996 (1996-03-27) the whole document	1-14
A	WO 99/18486 A (KONINKL PHILIPS ELECTRONICS NV [NL]; PHILIPS SVENSKA AB [SE]) 15 April 1999 (1999-04-15)	

☐ 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	Date of mailing of the international search report
23 May 2007	06/06/2007

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	Moraiti, Marina

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/GB2007/000473

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 1089167 A	04-04-2001	JP 2001142692 A	25-05-2001
		KR 20010050792 A	25-06-2001
		TW 525087 B	21-03-2003
		US 2005262329 A1	24-11-2005
EP 0324308 B1	27-03-1996	DE 3855158 D1	02-05-1996
		DE 3855158 T2	10-10-1996
		EP 0324308 A2	19-07-1989
		JP 1201729 A	14-08-1989
		US 5115500 A	19-05-1992
WO 9918486 A	15-04-1999	JP 3816961 B2	30-08-2006
		JP 2001508909 T	03-07-2001
		US 6298434 B1	02-10-2001