



(43) International Publication Date
11 April 2013 (11.04.2013)

- (51) International Patent Classification:
G06F 1/00 (2006.01) G06F 9/06 (2006.01)
G06F 12/00 (2006.01)
- (21) International Application Number:
PCT/US2011/055122
- (22) International Filing Date:
6 October 2011 (06.10.2011)
- (25) Filing Language: English
- (26) Publication Language: English
- (71) Applicant (for all designated States except US): INTEL CORPORATION [US/US]; 2200 Mission College Boulevard, M/S RNB-4-150, Santa Clara, CA 95052 (US).
- (72) Inventors; and
- (75) Inventors/Applicants (for US only): GENDLER, Alexander; Hahashmonaim 39a/40, 27041 Kiriat Motzkin, Ha (IL). NOVAKOVSKY, Larisa; Yakinton 26/5, 34792 Haifa, Ha (IL). LEIFMAN, George; Avraham Sachnin 7, 32981 Haifa, Ha (IL). RIP, Dana; 1 Harduf St., 34747 Haifa, Ha (IL).
- (74) Agents: WEBSTER, Thomas, C. et al.; Blakely, Sokoloff, Taylor & Zafman LLP, 1279 Oakmead Parkway, Sunnyvale, CA 94085-4040 (US).

- (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, 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, 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, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, 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, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, 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, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— of inventorship (Rule 4.17(iv))

Published:

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

(54) Title: APPARATUS AND METHOD FOR DYNAMICALLY MANAGING MEMORY ACCESS BANDWIDTH IN MULTI-CORE PROCESSOR

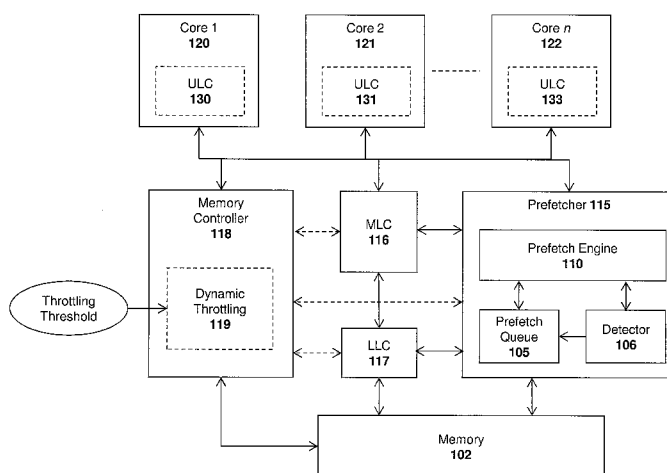


Fig. 1a

(57) Abstract: An apparatus and method are described for performing history-based prefetching. For example a method according to one embodiment comprises: determining if a previous access signature exists in memory for a memory page associated with a current stream; if the previous access signature exists, reading the previous access signature from memory; and issuing prefetch operations using the previous access signature.

WO 2013/052056 A1

APPARATUS AND METHOD FOR DYNAMICALLY MANAGING MEMORY ACCESS BANDWIDTH IN A MULTI-CORE PROCESSOR

Field of the Invention

[0001] This invention relates generally to the field of computer processors. More particularly, the invention relates to an apparatus and method for dynamically managing memory bandwidth in a multi-core processor.

Description of the Related Art

[0002] Many modern microprocessors have large instruction pipelines that facilitate high speed operation. "Fetched" program instructions enter the pipeline, undergo operations such as decoding and executing in intermediate stages of the pipeline, and are "retired" at the end of the pipeline. When the pipeline receives a valid instruction and the data needed to process the instruction each clock cycle, the pipeline remains full and performance is good. When valid instructions are not received each cycle and/or when the necessary data is not available the pipeline may stall and performance can suffer. For example, performance problems can result from branch instructions in program code. If a branch instruction is encountered in the program and the processing branches to the target address, a portion of the instruction pipeline may have to be flushed, resulting in a performance penalty. Moreover, even with sequentially executed (i.e., non-branch) instructions, modern microprocessors are much faster than the memory where the program is kept, meaning that the program's instructions and data cannot be read fast enough to keep the microprocessor busy.

[0003] System performance may be enhanced and effective memory access latency may be reduced by anticipating the needs of a processor. If the data and instructions needed by a processor in the near future are predicted, then the data and instructions can be fetched in advance or "prefetched",

such that the data/instructions are buffered/cached and available to the processor with low latency. A prefetcher that accurately predicts a READ request (such as, for example, for a branch instruction) and issues it in advance of an actual READ can thus, significantly improve system performance. Prefetchers can be implemented in a CPU or in a chipset, and prefetching schemes have been routinely used for both.

[0004] Prefetching may be performed at various levels of a CPU's cache hierarchy. For example, some current x86-based processors include a Level 2 ("L2" or "MLC") cache stream prefetcher to reduce the number of L2 and lower level (e.g., "L3" or "LLC") cache misses. The stream prefetcher predicts future accesses within a memory page based on the order of accesses within that page and the distance between subsequent accesses.

[0005] In a multi-core processor, each processor core must share a portion of the overall bandwidth for accesses to main memory (i.e., memory bandwidth is a shared resource). Consequently, there may be situations where overly-aggressive prefetching of one core consumes most of the shared memory bandwidth, thereby causing the demand requests of other cores to stall and reducing performance.

[0006] As such, what is needed is a more intelligent method for controlling prefetching aggressiveness to improve processor performance.

BRIEF DESCRIPTION OF THE DRAWINGS

[0007] A better understanding of the present invention can be obtained from the following detailed description in conjunction with the following drawings, in which:

[0008] **FIGS. 1a-b** illustrate one embodiment of a processor architecture for performing dynamic throttling of prefetch aggressiveness.

[0009] **FIG. 2** illustrates a method for performing dynamic throttling of prefetch aggressiveness.

[0010] **FIG. 3** illustrates a computer system on which embodiments of the invention may be implemented.

[0011] **FIG. 4** illustrates another computer system on which embodiments of the invention may be implemented.

DETAILED DESCRIPTION

[0012] In the following description, for the purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the embodiments of the invention described below. It will be apparent, however, to one skilled in the art that the embodiments of the invention may be practiced without some of these specific details. In other instances, well-known structures and devices are shown in block diagram form to avoid obscuring the underlying principles of the embodiments of the invention.

[0013] As mentioned above, limited memory bandwidth in multi-core architectures creates situations where the aggressive prefetching of one core consumes most of the memory bandwidth. As a result, the demand requests of other cores cannot be served, resulting in a performance hit.

[0014] One embodiment of the invention addresses these problems by controlling the aggressiveness of core prefetching. Specifically, in one embodiment, a throttling threshold value is set and prefetching is throttled down or disabled when the current ratio of the number of mid-level cache (MLC) hits over the number of demands for the current detector is below the specified throttling threshold value. Prefetching may be throttled back up when this ratio rises above the specified throttling threshold value.

[0015] **Figure 1a** illustrates an exemplary processor architecture on which embodiments of the invention may be implemented. The architecture includes a plurality of processor cores 120-122 each containing its own upper level cache (“ULC” or sometimes referred to as a level 1 (“L1”) cache) 130-133, respectively, for caching instructions and data. The architecture also includes a memory controller 118 with dynamic throttling logic 119 for implementing the dynamic throttling techniques described herein. A mid-level cache (“MLC” or sometimes referred to as a level 2 (“L2”) cache) and a lower level cache 117 are employed for caching instructions and data according to a specified cache management policy. The cache management policy may comprise an inclusive policy in which any cache line stored in a cache relatively higher in the hierarchy (e.g., the ULC) is also present in a cache further down the hierarchy (e.g., in the MLC 116 or LLC 117). Alternatively, an exclusive cache management policy may be implemented in which a cache line is stored in only one cache in the hierarchy at a time (excluding all other caches from storing the cache line). The underlying principles of the invention may be implemented on processors having either inclusive or exclusive cache management policies.

[0016] The architecture shown in **Figure 1a** also includes a prefetch unit 115 with a prefetch engine 110 which executes an algorithm for prefetching instructions from memory 102 and storing the prefetched instructions within a prefetch queue 105 from which they may be read into one of the various caches 116-117, 130-133 prior to execution by one of the cores 120-122. As it is well understood by those of skill in the art, the prefetch engine 110 implements an algorithm which attempts to predict the instructions which each core will require in the future and responsively pre-fetches those instructions from memory 102.

[0017] To this end, the prefetcher 115 includes detector logic 106 which may include multiple detectors for learning and identifying prefetch candidates. The detector 106 of one embodiment comprises a detector table, with each

entry in the table identifying a specified contiguous physical address region of memory 102 from which prefetch operations are to be executed. The detector identifies a particular region with a region address and includes state information for learning and identifying prefetch candidates.

[0018] In one embodiment, the dynamic throttling logic 119 controls the prefetch engine 110 to throttle up or down prefetch requests in response to a specified throttling threshold. Specifically, in one embodiment, the throttling threshold is set at one of the following values: (1) no throttle (throttling as described herein is disabled); (2) 25% or $\frac{1}{4}$ (low throttle); (3) 50% or $\frac{1}{2}$ (medium throttle); and (4) 75% or $\frac{3}{4}$ (high throttle). In one embodiment, the dynamic throttling logic 119 monitors the number of MLC cache hits in relation to the number of demands generated by the cores and, if the ratio of the number of MLC cache hits to the number of demands is below the current specified throttling threshold, then the dynamic throttling logic 119 signals to the prefetcher 115 to cease any new prefetch requests. In one embodiment, the above techniques are implemented only when the current detector has more than one outstanding demand.

[0019] It should be noted that the underlying principles of the invention are not limited to the particular cache layout shown in **Figure 1a**. For example, in an alternative embodiment, each processor core may have its own dedicated MLC and/or LLC. In yet another embodiment, a single ULC may be shared between the cores 120-122. Various other architectural modifications may be implemented while still complying with the underlying principles of the invention.

[0020] As illustrated in **Figure 1b**, in one embodiment, the prefetch queue 105 comprises an output queue 141 and a super queue 142. Prefetched instructions flow along the prefetch pipeline from the detector 106 to the output queue 141, to the super queue 142. In one embodiment, various points in the prefetching pipeline may be controlled to control prefetch

aggressiveness. For example, as indicated in **Figure 1b**, prefetch parameters may be controlled at the detector 106. The output queue 141 may also be decreased in size or blocked and/or the output of the super queue 142 may be dropped.

[0021] A method according to one embodiment of the invention is illustrated in **Figure 2**. The method may be implemented using the microprocessor architecture shown in **Figures 1a-b** but is not necessarily limited to any particular microprocessor architecture.

[0022] At 201 a determination is made as to whether the current prefetch detector has more than one demand pending. If not, then the current throttling threshold is set to No Throttle at 206 (i.e., because if only a single demand is pending then the problems associated with aggressive prefetching described above are not present). If, however, the current detector has more than one demand, then at 202, the throttling threshold may be set at (1) 25% or $\frac{1}{4}$ (low throttle); (2) 50% or $\frac{1}{2}$ (medium throttle); or (3) 75% or $\frac{3}{4}$ (high throttle).

[0023] At 203, the ratio of the number of MLC hits to the number of MLC demands is calculated and, at 204, this ratio is compared to the current throttling threshold. If the ratio is lower than the current throttling threshold, then at 205, steps are taken to throttle down prefetch requests. For example, in one embodiment, the prefetch unit will not issue new requests if the ratio of the number of MLC hits to the number of MLC demands is below the threshold.

[0024] In one embodiment, to reduce additional pressure on the memory controller, least recently used (LRU) hints are disabled from the cache management policy if the throttle level is set at low, medium or high. LRU hints are typically employed to identify least recently used cache lines for eviction. Disabling LRU hints in this embodiment will have the effect of

reducing traffic on the communication ring connecting the cores 120-122 and help balance the system.

[0025] The following additional prefetch parameters may set in one embodiment of the invention:

- The value of “double_mlc_window_watermark” may be set higher to cause the issuance of more MLC prefetch requests. In one embodiment, the double_mlc_window_watermark variable multiplies the possible number of prefetch request with parking in both the MLC 116 and LLC 117s.
- The value of “llc_only_watermark” may be set higher, thereby forcing all prefetch request parking to the LLC 117 only.
- Kick start may send 6 instead of 4 requests.

[0026] In one embodiment, the foregoing parameters are set as follows for each of the throttle thresholds:

No Throttle: In one embodiment, the no throttle condition is implemented with “double_mlc_window_watermark” set to its higher value (e.g., 11), with “llc_only_watermark” set to its higher value (e.g., 14), and with 6 kick start requests.

Low Throttle: In one embodiment, the low throttle condition is implemented with “double_mlc_window_watermark” set to its standard value (e.g., 6), with “llc_only_watermark” set to its standard value (e.g., 12), and with 4 kick start requests. In one embodiment, if the number of demands for the detector is higher than the threshold (default 2), then the MLC hit/demand ratio is checked to determine if it is below the $\frac{1}{4}$ threshold throttle value, as described above.

Medium Throttle: In one embodiment, the medium throttle condition is implemented with “double_mlc_window_watermark” set to its standard value (e.g., 6), with “llc_only_watermark” set to its standard value (e.g., 12), and

with 4 kick start requests. In one embodiment, if the number of demands for the detector is higher than the threshold (default 2), then the MLC hit/demand ratio is checked to determine if it is below the $\frac{1}{2}$ threshold throttle value, as described above.

High Throttle: In one embodiment, the high throttle condition is implemented with “double_mlc_window_watermark” set to its standard value (e.g., 6), with “llc_only_watermark” set to its standard value (e.g., 12), and with 4 kick start requests. In one embodiment, if the number of demands for the detector is higher than the threshold (default 2), then the MLC hit/demand ratio is checked to determine if it is below the $\frac{3}{4}$ threshold throttle value, as described above.

[0027] The specific values set forth above are used merely for the purposes of illustration of one specific embodiment of the invention. It should be noted, however, that the underlying principles of the invention are not limited to an implementation having these particular values.

[0028] Referring now to **Figure 3**, shown is a block diagram of a computer system 300 in accordance with one embodiment of the present invention. The system 300 may include one or more processing elements 310, 315, which are coupled to graphics memory controller hub (GMCH) 320. The optional nature of additional processing elements 315 is denoted in FIG. 3 with broken lines.

[0029] Each processing element may be a single core or may, alternatively, include multiple cores. The processing elements may, optionally, include other on-die elements besides processing cores, such as integrated memory controller and/or integrated I/O control logic. Also, for at least one embodiment, the core(s) of the processing elements may be multithreaded in that they may include more than one hardware thread context per core.

[0030] **Figure 3** illustrates that the GMCH 320 may be coupled to a memory 340 that may be, for example, a dynamic random access memory (DRAM). The DRAM may, for at least one embodiment, be associated with a non-volatile cache.

[0031] The GMCH 320 may be a chipset, or a portion of a chipset. The GMCH 320 may communicate with the processor(s) 310, 315 and control interaction between the processor(s) 310, 315 and memory 340. The GMCH 320 may also act as an accelerated bus interface between the processor(s) 310, 315 and other elements of the system 300. For at least one embodiment, the GMCH 320 communicates with the processor(s) 310, 315 via a multi-drop bus, such as a frontside bus (FSB) 395.

[0032] Furthermore, GMCH 320 is coupled to a display 340 (such as a flat panel display). GMCH 320 may include an integrated graphics accelerator. GMCH 320 is further coupled to an input/output (I/O) controller hub (ICH) 350, which may be used to couple various peripheral devices to system 300. Shown for example in the embodiment of FIG. 3 is an external graphics device 360, which may be a discrete graphics device coupled to ICH 350, along with another peripheral device 370.

[0033] Alternatively, additional or different processing elements may also be present in the system 300. For example, additional processing element(s) 315 may include additional processors(s) that are the same as processor 310, additional processor(s) that are heterogeneous or asymmetric to processor 310, accelerators (such as, e.g., graphics accelerators or digital signal processing (DSP) units), field programmable gate arrays, or any other processing element. There can be a variety of differences between the physical resources 310, 315 in terms of a spectrum of metrics of merit including architectural, microarchitectural, thermal, power consumption characteristics, and the like. These differences may effectively manifest themselves as asymmetry and heterogeneity amongst the processing

elements 310, 315. For at least one embodiment, the various processing elements 310, 315 may reside in the same die package.

[0034] **Figure 4** is a block diagram illustrating another exemplary data processing system which may be used in some embodiments of the invention. For example, the data processing system 400 may be a handheld computer, a personal digital assistant (PDA), a mobile telephone, a portable gaming system, a portable media player, a tablet or a handheld computing device which may include a mobile telephone, a media player, and/or a gaming system. As another example, the data processing system 400 may be a network computer or an embedded processing device within another device.

[0035] According to one embodiment of the invention, the exemplary architecture of the data processing system 900 may be used for the mobile devices described above. The data processing system 900 includes the processing system 420, which may include one or more microprocessors and/or a system on an integrated circuit. The processing system 420 is coupled with a memory 910, a power supply 425 (which includes one or more batteries) an audio input/output 440, a display controller and display device 460, optional input/output 450, input device(s) 470, and wireless transceiver(s) 430. It will be appreciated that additional components, not shown in **Figure 4**, may also be a part of the data processing system 400 in certain embodiments of the invention, and in certain embodiments of the invention fewer components than shown in **Figure 4** may be used. In addition, it will be appreciated that one or more buses, not shown in **Figure 4**, may be used to interconnect the various components as is well known in the art.

[0036] The memory 410 may store data and/or programs for execution by the data processing system 400. The audio input/output 440 may include a microphone and/or a speaker to, for example, play music and/or provide

telephony functionality through the speaker and microphone. The display controller and display device 460 may include a graphical user interface (GUI). The wireless (e.g., RF) transceivers 430 (e.g., a WiFi transceiver, an infrared transceiver, a Bluetooth transceiver, a wireless cellular telephony transceiver, etc.) may be used to communicate with other data processing systems. The one or more input devices 470 allow a user to provide input to the system. These input devices may be a keypad, keyboard, touch panel, multi touch panel, etc. The optional other input/output 450 may be a connector for a dock.

[0037] Other embodiments of the invention may be implemented on cellular phones and pagers (e.g., in which the software is embedded in a microchip), handheld computing devices (e.g., personal digital assistants, smartphones), and/or touch-tone telephones. It should be noted, however, that the underlying principles of the invention are not limited to any particular type of communication device or communication medium.

[0038] Embodiments of the invention may include various steps, which have been described above. The steps may be embodied in machine-executable instructions which may be used to cause a general-purpose or special-purpose processor to perform the steps. Alternatively, these steps may be performed by specific hardware components that contain hardwired logic for performing the steps, or by any combination of programmed computer components and custom hardware components.

[0039] Elements of the present invention may also be provided as a computer program product which may include a machine-readable medium having stored thereon instructions which may be used to program a computer (or other electronic device) to perform a process. The machine-readable medium may include, but is not limited to, floppy diskettes, optical disks, CD-ROMs, and magneto-optical disks, ROMs, RAMs, EPROMs, EEPROMs, magnet or optical cards, propagation media or other type of

media/machine-readable medium suitable for storing electronic instructions. For example, the present invention may be downloaded as a computer program product, wherein the program may be transferred from a remote computer (e.g., a server) to a requesting computer (e.g., a client) by way of data signals embodied in a carrier wave or other propagation medium via a communication link (e.g., a modem or network connection).

[0040] Throughout this detailed description, for the purposes of explanation, numerous specific details were set forth in order to provide a thorough understanding of the present invention. It will be apparent, however, to one skilled in the art that the invention may be practiced without some of these specific details. In certain instances, well known structures and functions were not described in elaborate detail in order to avoid obscuring the subject matter of the present invention. Accordingly, the scope and spirit of the invention should be judged in terms of the claims which follow.

CLAIMS

What is claimed is:

1. A method for dynamically adjusting prefetch requests to improve performance in a multi-core processor comprising:
 - setting a current throttling threshold to one of a plurality of selectable threshold levels;
 - determining a ratio of a number of current mid-level cache (MLC) hits to MLC demands;
 - throttling down prefetch requests if the ratio of the number of current MLC hits to MLC demands is below the currently selected throttling threshold level.

2. The method as in claim 1 wherein the plurality of selectable threshold levels includes a low throttle level of 25% or $\frac{1}{4}$, a medium throttle level of 50% or $\frac{1}{2}$ and the high throttle level comprises 75% or $\frac{3}{4}$.

3. The method as in claim 2 further comprising:
 - disabling least recently used (LRU) hints when the current threshold level is set at a low throttle level, medium throttle level, or high throttle level.

4. The method as in claim 1 further comprising:
 - determining if the current prefetch detector has more than one demand pending; and
 - if not, then setting the current throttling threshold level to No Throttle.

5. An apparatus for dynamically adjusting prefetch requests to improve performance in a multi-core processor comprising:
 - a mid-level cache (MLC) for caching instructions and data according to a specified cache management policy;
 - a prefetcher unit to prefetch instructions from memory, the instructions to be prefetched being identified by a prefetch detector;

a memory controller with dynamic throttling logic to perform the operations of:

setting a current throttling threshold to one of a plurality of selectable threshold levels;

determining a ratio of a number of MLC hits to MLC demands; and

throttling down prefetch requests if the ratio of the number of current MLC hits to MLC demands is below the currently selected throttling threshold level.

6. The apparatus as in claim 5 wherein the plurality of selectable threshold levels includes a low throttle level of 25% or $\frac{1}{4}$, a medium throttle level of 50% or $\frac{1}{2}$ and the high throttle level comprises 75% or $\frac{3}{4}$.

7. The apparatus as in claim 6 wherein the memory controller disables least recently used (LRU) hints when the current threshold level is set at a low throttle level, medium throttle level, or high throttle level.

8. The method as in claim 1 wherein memory controller is configured to perform the additional operations of:

determining if the current prefetch detector has more than one demand pending; and

if not, then setting the current throttling threshold level to No Throttle.

9. A computer system comprising:

a display device;

a memory for storing instructions;

a multi-core processor for processing the instructions, the multi-core processor dynamically adjusting prefetch requests to improve performance by performing the operations of:

setting a current throttling threshold to one of a plurality of selectable threshold levels;

determining a ratio of a number of current mid-level cache (MLC) hits to MLC demands;

throttling down prefetch requests if the ratio of the number of current MLC hits to MLC demands is below the currently selected throttling threshold level.

10. The system as in claim 9 wherein the plurality of selectable threshold levels includes a low throttle level of 25% or $\frac{1}{4}$, a medium throttle level of 50% or $\frac{1}{2}$ and the high throttle level comprises 75% or $\frac{3}{4}$.

11. The system as in claim 10 wherein the multi-core processor disables the least recently used (LRU) hints when the current threshold level is set at a low throttle level, medium throttle level, or high throttle level.

12. The system as in claim 10 wherein the multi-core processor performs the additional operations of:

determining if the current prefetch detector has more than one demand pending; and

if not, then setting the current throttling threshold level to No Throttle.

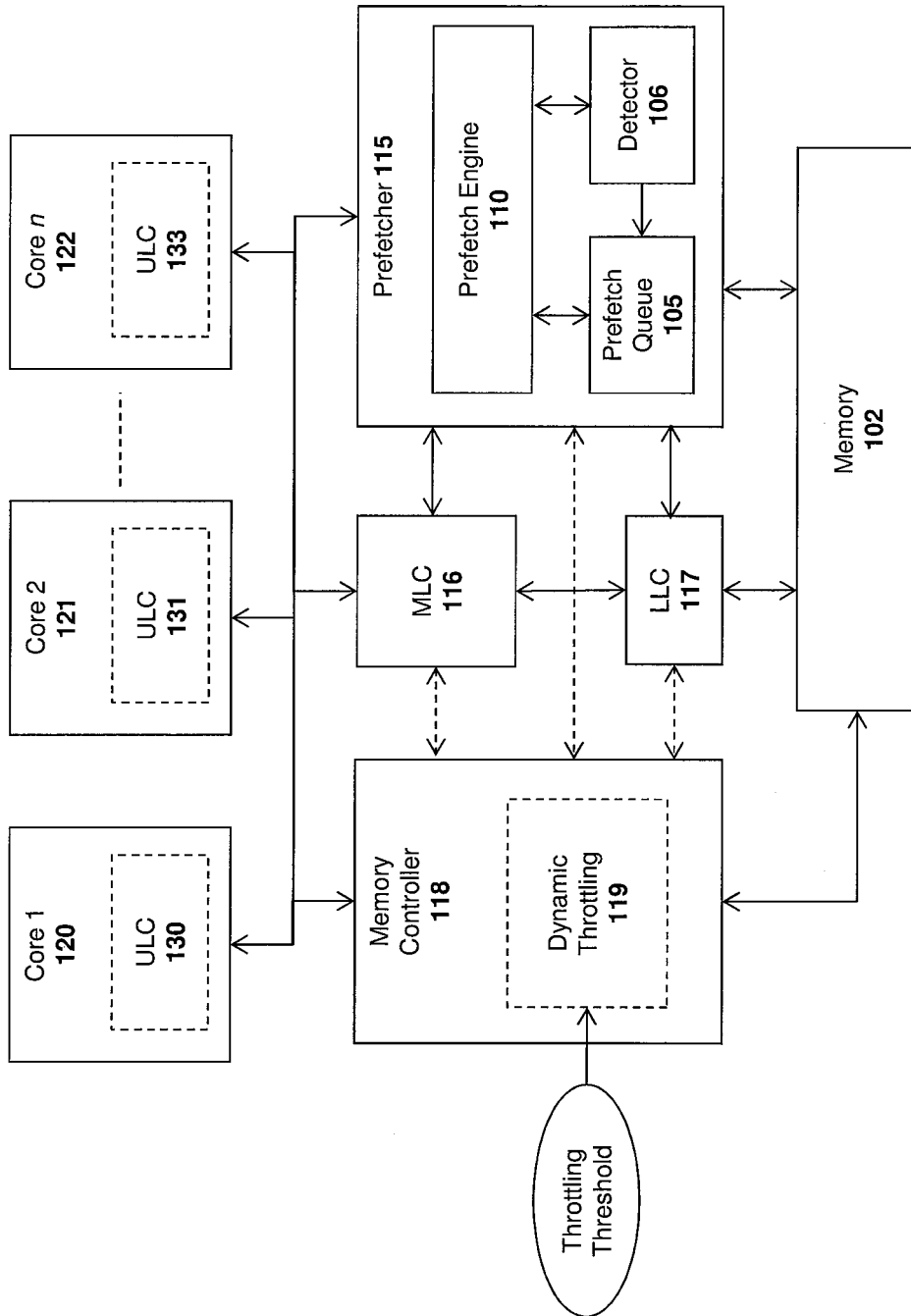


Fig. 1a

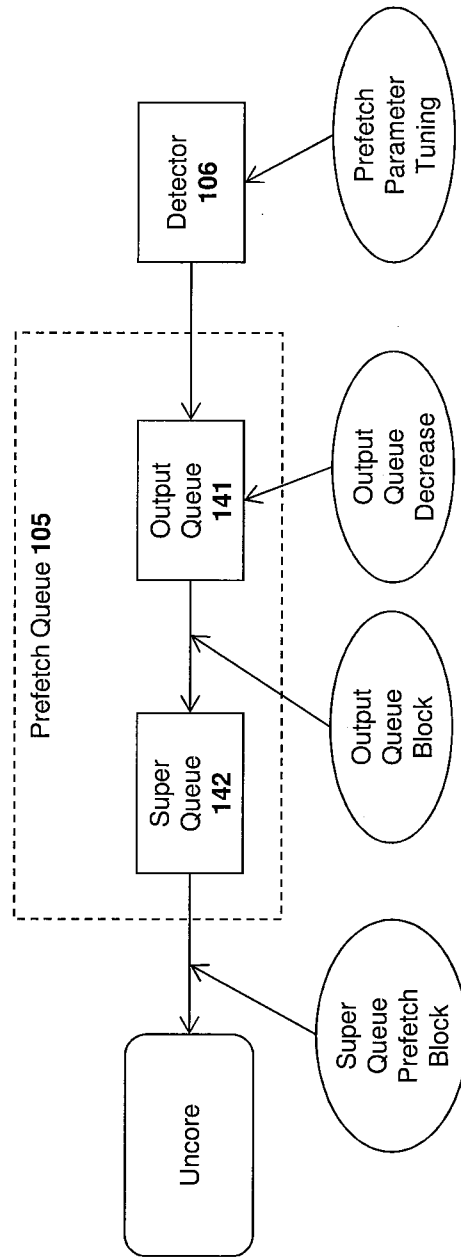


Fig. 1b

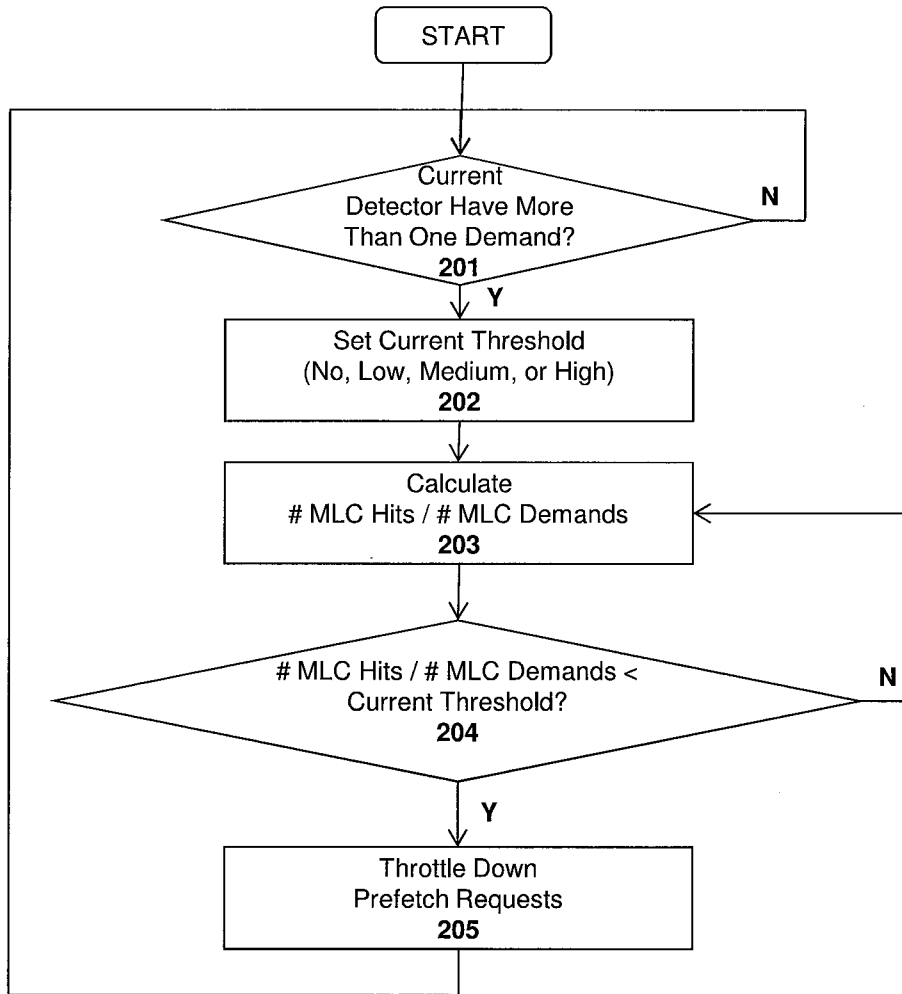


Fig. 2

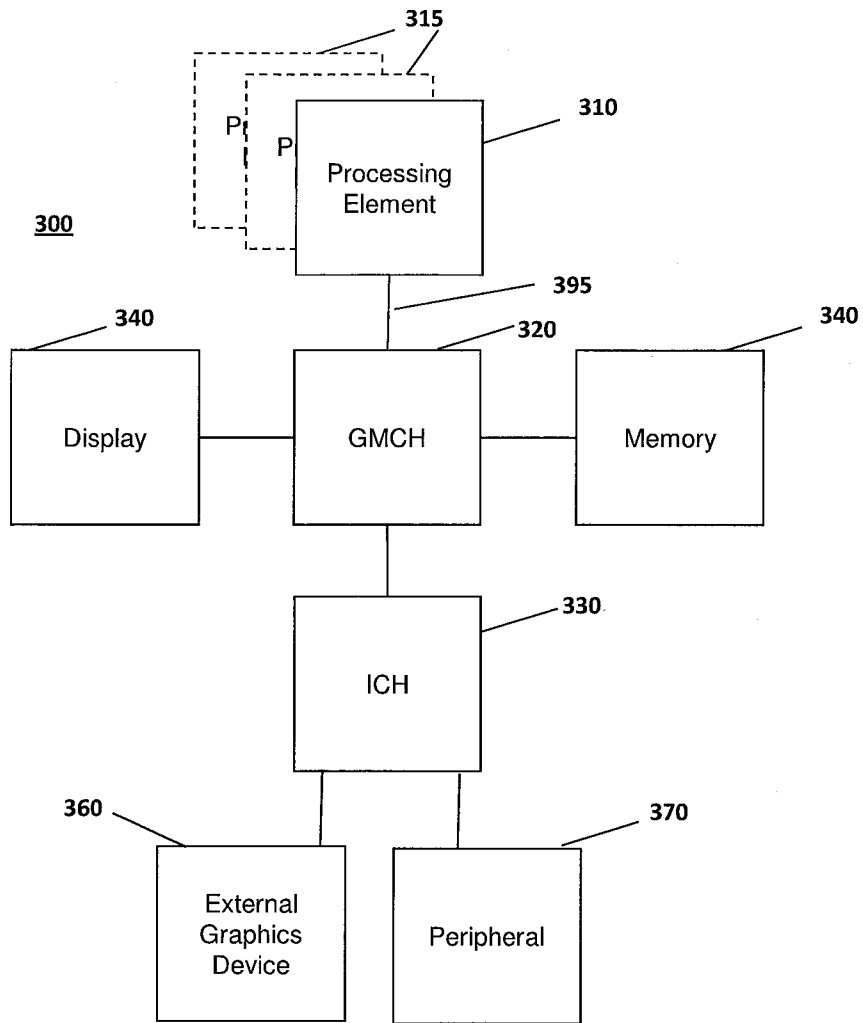


Fig. 3

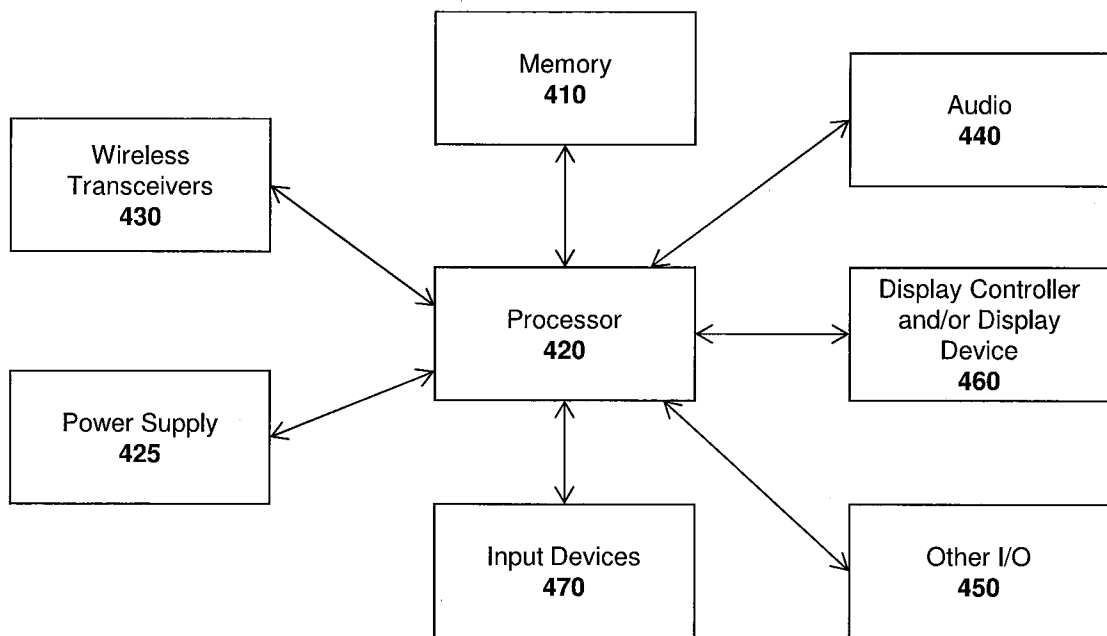


Fig. 4

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2011/055122**A. CLASSIFICATION OF SUBJECT MATTER***G06F 1/00(2006.01)i, G06F 12/00(2006.01)i, G06F 9/06(2006.01)i*

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 1/00; G06F 9/30; G06F 12/00; G06F 12/08

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

Korean utility models and applications for utility models

Japanese utility models and applications for utility models

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

eKOMPASS(KIPO internal) & Keywords: "CACHE" , "HIT" , "RATIO" , "PREFETCH"

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2008-0229071 A1 (SHIOYA KATSUHIKO et al.) 18 September 2008 See abstract, paragraphs [0029]-[0051], and figures 2,3,7	1,5,9
Y	US 2009-0019229 A1 (MORROW MICHAEL WILLIAM et al.) 15 January 2009 See paragraphs [0007],[0019]-[0020], claim 1, and figure 6	1,5,9
A	US 2005-0257005 A1 (JOSEPH JEDDELOH) 17 November 2005 See abstract, paragraphs [0033],[0043] and figure 3	1-12
A	US 2004-0205298 A1 (BRIAN S. BEARDEN et al.) 14 October 2004 See paragraphs [0059],[0100],[0106]-[0107], claim 1, and figure 8	1-12

 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 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

22 MAY 2012 (22.05.2012)

Date of mailing of the international search report

24 MAY 2012 (24.05.2012)

Name and mailing address of the ISA/KR

Korean Intellectual Property Office
Government Complex-Daejeon, 189 Cheongsu-ro,
Seo-gu, Daejeon 302-701, Republic of Korea

Facsimile No. 82-42-472-7140

Authorized officer

YUN, BYEONG SOO

Telephone No. 82-42-481-8530



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2011/055122

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2008-0229071 A1	18.09.2008	JP 2008-225915 A	25.09.2008
US 2009-0019229 A1	15.01.2009	CN 101689147 A	31.03.2010
		EP 2015189 A1	14.01.2009
		JP 2010-532904 A	14.10.2010
		JP 2010-532904 T	14.10.2010
		KR 10-1095217 B1	20.12.2011
		KR20100022518A	02.03.2010
		US 7917702 B2	29.03.2011
		WO 2009-009704 A1	15.01.2009
US 2005-0257005 A1	17.11.2005	CN 101390060 A	18.03.2009
		EP 1756718 A2	28.02.2007
		EP 1756718 A4	26.11.2008
		JP 04-769797 B2	24.06.2011
		JP 2007-537541 A	20.12.2007
		JP 2007-537541 T	20.12.2007
		JP 4769797 B2	07.09.2011
		KR 10-0813422 B1	13.03.2008
		KR20070012857A	29.01.2007
		TW 1309772A	11.05.2009
		US 2007-0033353 A1	08.02.2007
		US 2008-0133853 A1	05.06.2008
		US 2008-0307023 A1	11.12.2008
		US 7162567 B2	09.01.2007
		US 7353320 B2	01.04.2008
		US 7562178 B2	14.07.2009
		WO 2005-114427 A2	01.12.2005
		WO 2005-114427 A3	01.12.2005
US 2004-0205298 A1	14.10.2004	US 7146467 B2	05.12.2006