TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

Intelligent interrupt distributor

Citation for published version (APA):
Fatemi, H., Kapoor, A., Pineda de Gyvez, J., & Echeverri Escobar, J. D. (2014). Intelligent interrupt distributor. (Patent No. *US2014181351*).
https://nl.espacenet.com/publicationDetails/originalDocument?CC=US&NR=2014181351A1&KC=A1&FT=D&ND=3&date=20140626&DB=&locale=nl_NL#

Document status and date:
Published: 26/06/2014

Document Version:
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Download date: 04. Oct. 2023

US 20140181351A1

(19) **United States**

(12) **Patent Application Publication**

Fatemi et al.

(10) **Pub. No.: US 2014/0181351 A1**

(43) **Pub. Date: Jun. 26, 2014**

(54) **INTELLIGENT INTERRUPT DISTRIBUTOR**

(71) Applicant: **NXP B.V.**, Eindhoven (NL)

(72) Inventors: **Hamed Fatemi**, Eindhoven (NL); **Ajay Kapoor**, Veldhoven (NL); **Jose de Jesus Pineda de Gyvez**, Eindhoven (NL); **Juan Diego Echeverri Escobar**, Eindhoven (NL)

(73) Assignee: **NXP B.V.**, Eindhoven (NL)

(21) Appl. No.: **13/725,698**

(22) Filed: **Dec. 21, 2012**

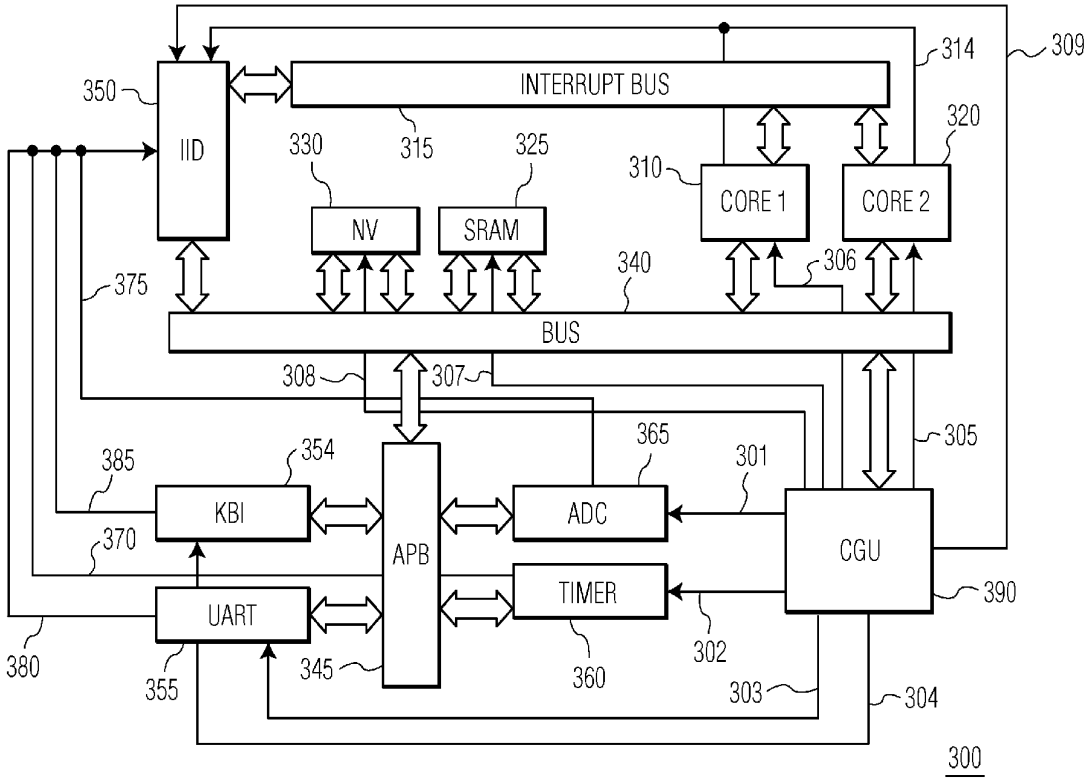
Publication Classification

(51) **Int. Cl.**
G06F 13/40 (2006.01)

(52) **U.S. Cl.**
CPC ***G06F 13/4068*** (2013.01)
USPC **710/313**

(57) **ABSTRACT**

An intelligent interrupt distributor balances interrupts (workload) in a highly parallelized system. The intelligent interrupt distributor distributes the interrupts between the processor cores. This allows lowering of voltage and frequency of individual processors and ensures that the overall system power consumption is reduced.



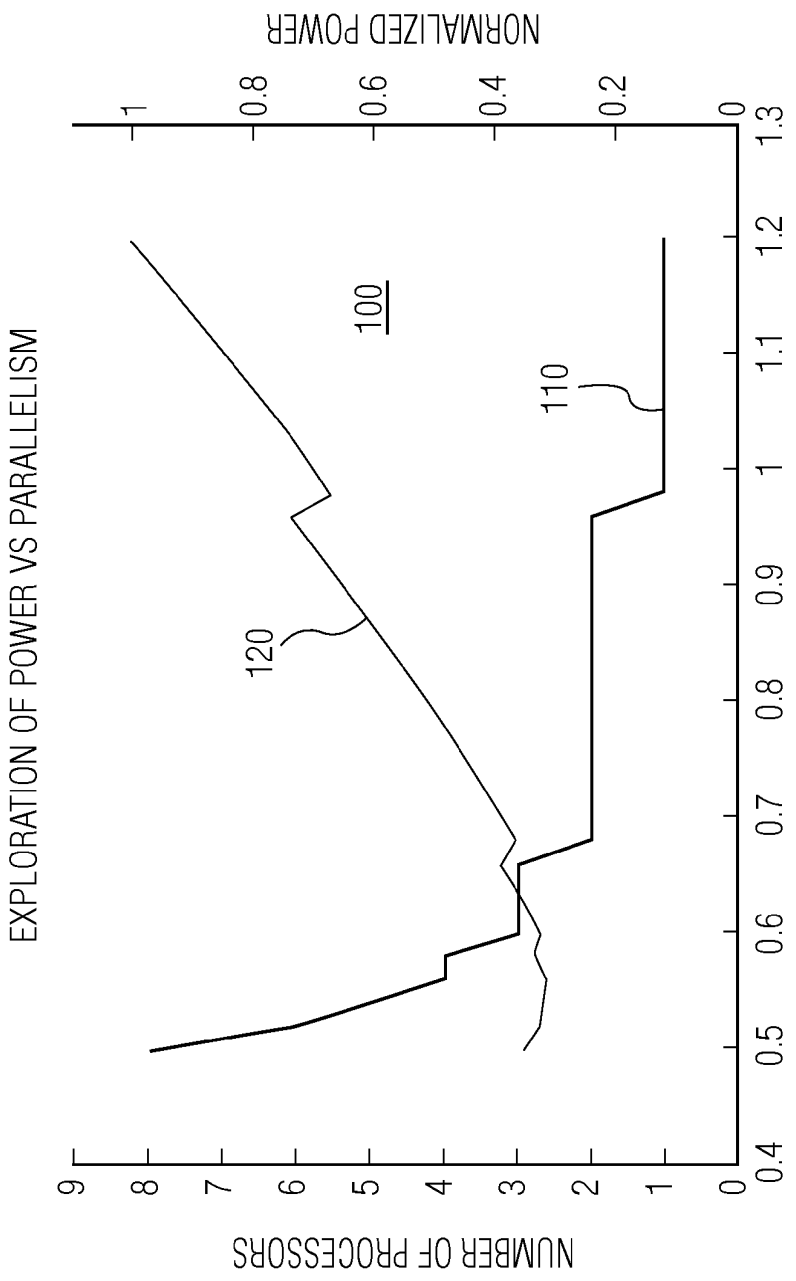

EXPLORATION OF POWER VS PARALLELISM
NUMBER OF PROCESSORS
NORMALIZED POWER
VDD [V]
100
110
120
0
1
2
3
4
5
6
7
8
9
0.2
0.4
0.6
0.8
1
0.4
0.5
0.6
0.7
0.8
0.9
1
1.1
1.2
1.3

FIG. 1

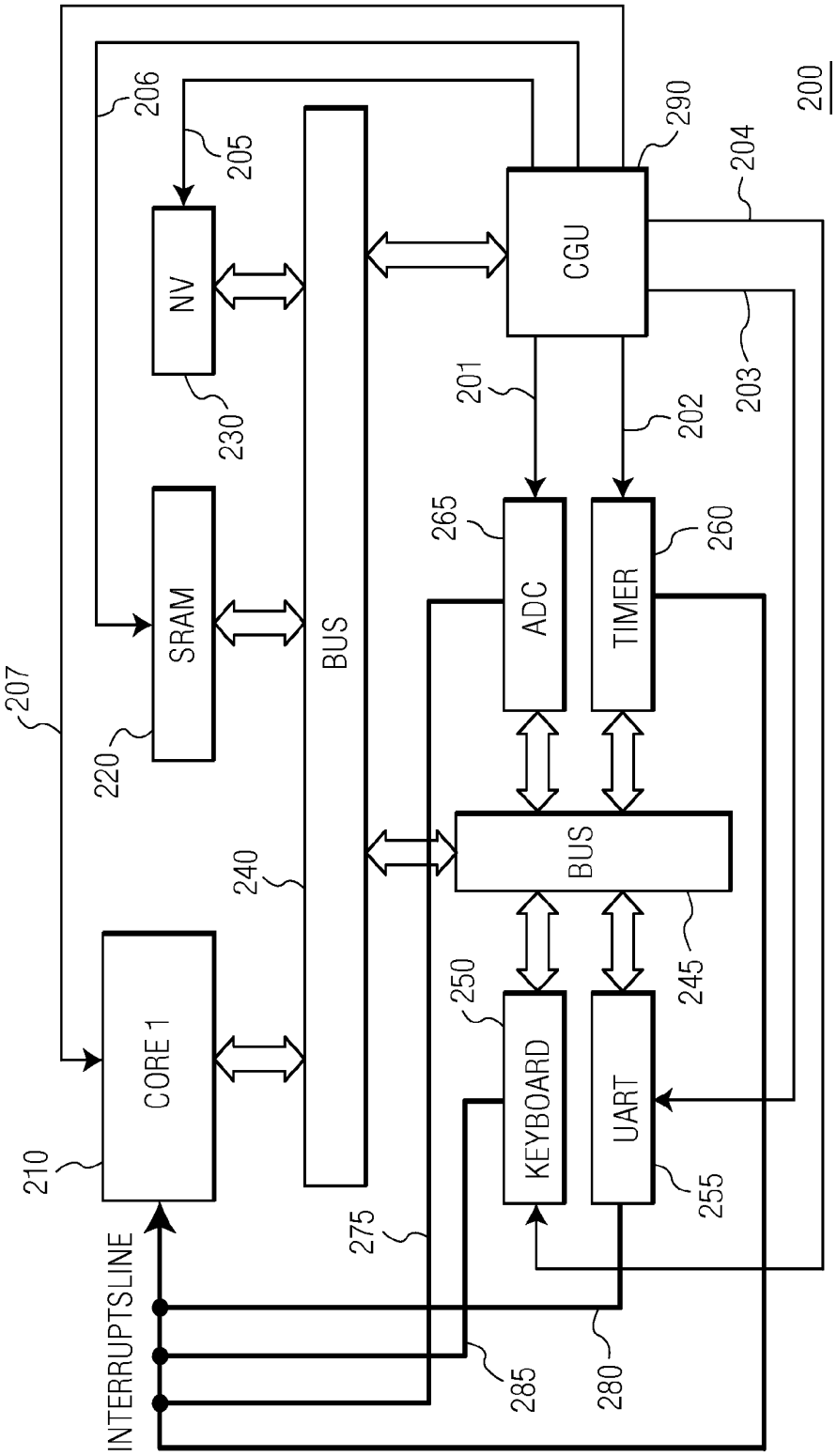


FIG. 2
PRIOR ART

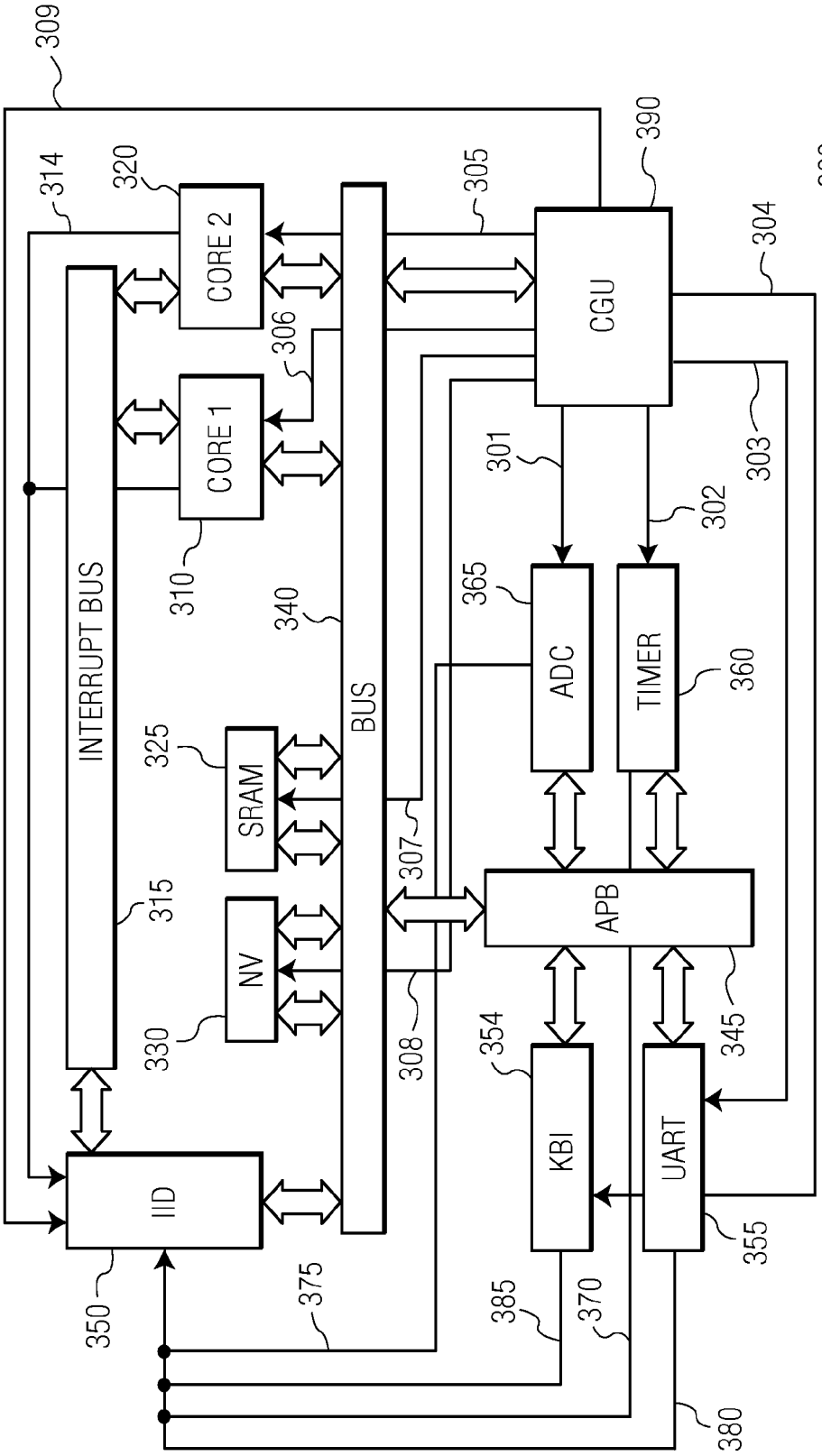

309
314
320
CORE 2
305
390
300
304
CGU
306
CORE 1
301
303
302
310
340
365
ADC
TIMER
360
INTERRUPT BUS
325
SRAM
BUS
307
315
APB
NV
330
308
354
345
KBI
UART
IID
355
350
375
385
370
380

FIG. 3

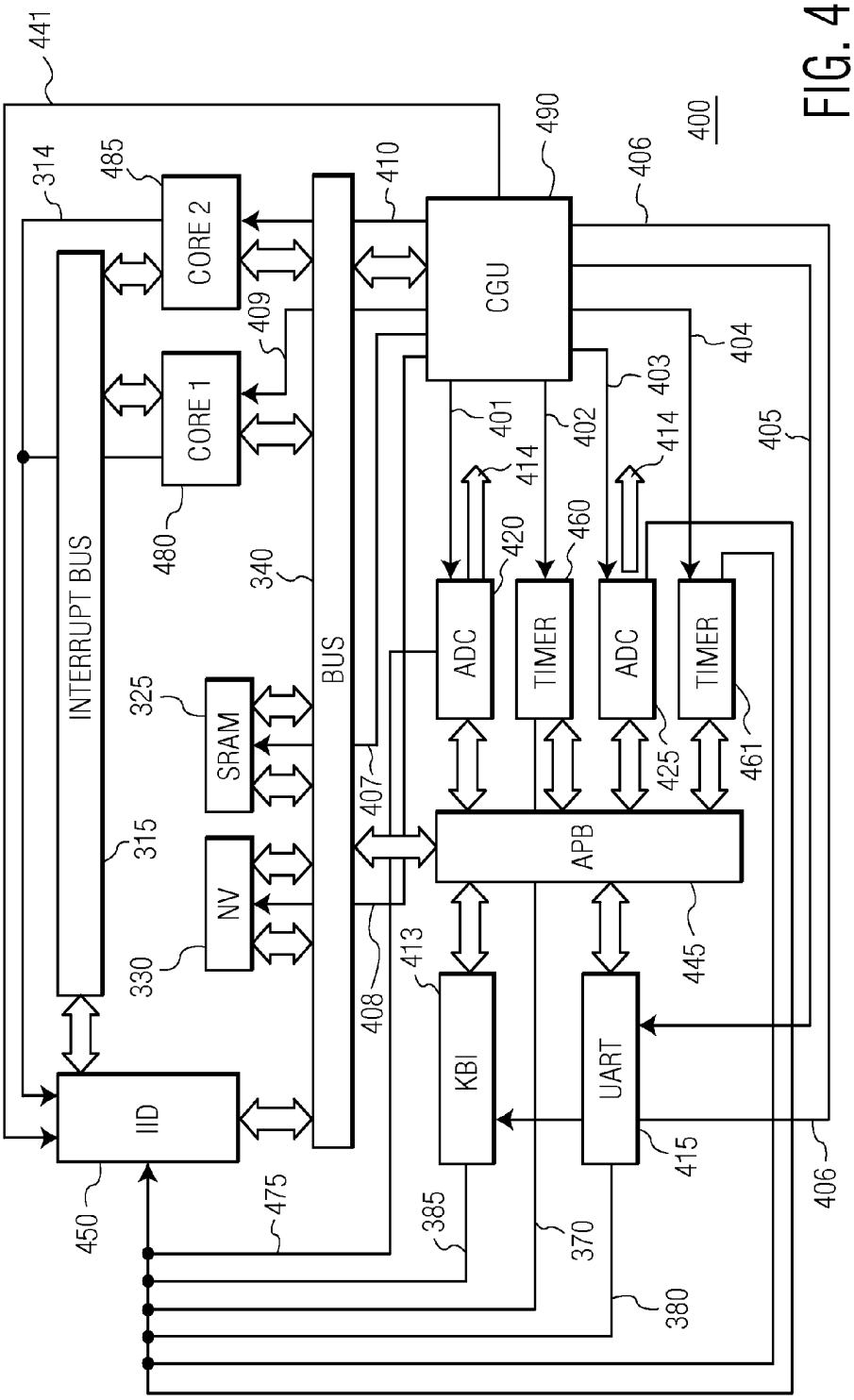

441
314
485
CORE 2
CORE 1
480
INTERRUPT BUS
315
325
SRAM
330
NV
340
BUS
410
409
CGU
490
406
400
404
403
401
402
414
405
420
460
ADC
TIMER
ADC
TIMER
425
461
407
APB
445
408
413
KBI
UART
415
IID
450
475
385
370
380

FIG. 4

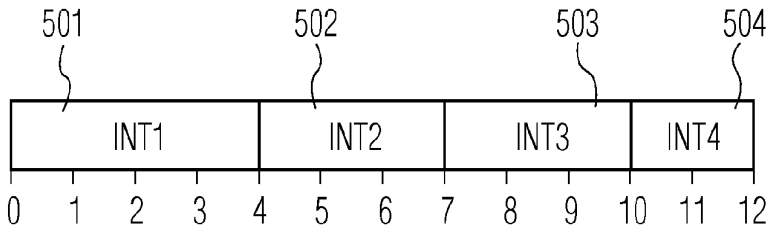


FIG. 5
PRIOR ART

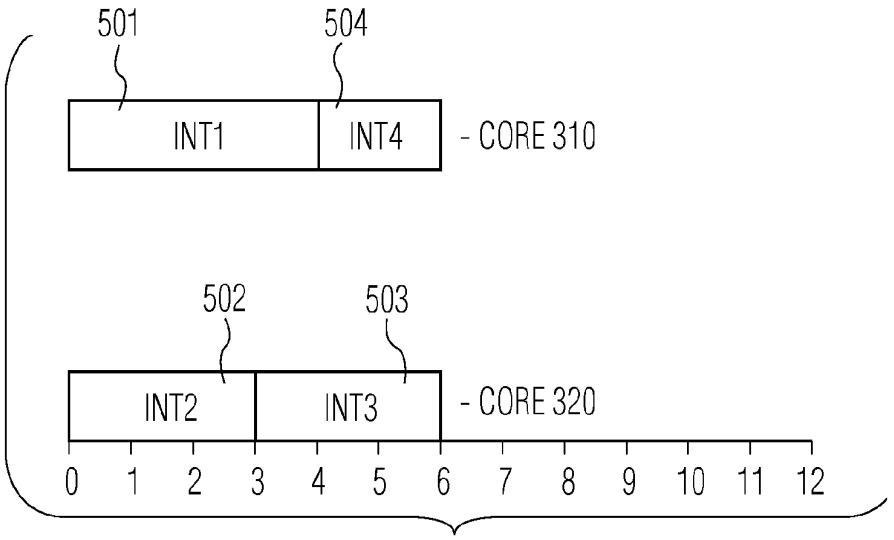


FIG. 6

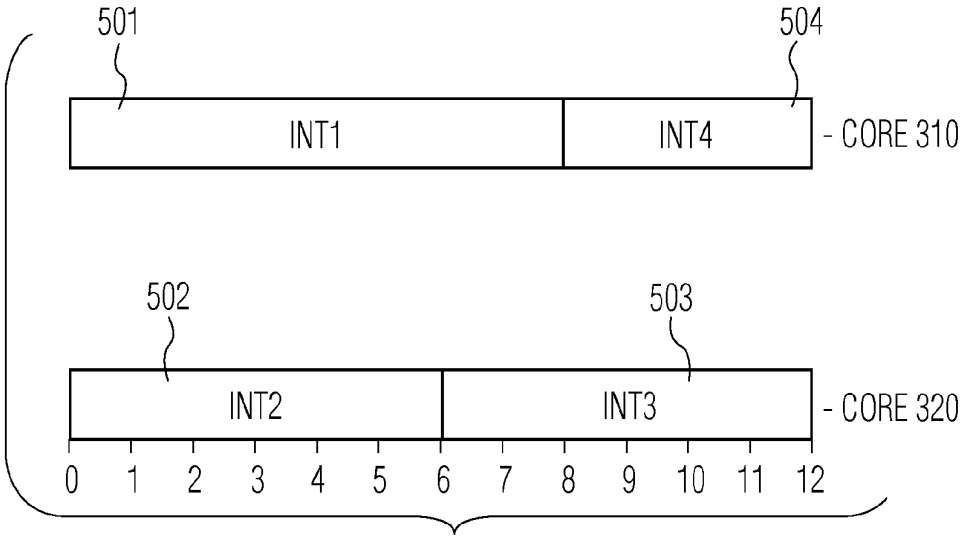


FIG. 7

INTELLIGENT INTERRUPT DISTRIBUTOR

BACKGROUND

[0001] Typically, computing systems such as desktop computers and mainframes are designed to provide the highest possible throughput. However, in the last decade or so, the proliferation of mobile computing systems such as laptops, smartphones and tablets which typically place a premium on long battery life has shifted the design focus towards optimizing both speed and battery lifetime. Mobile computing systems incorporate the minimization of power consumption as an important design parameter. The advent of E-metering, microcontrollers, sensors and smartcards has made minimization of power consumption an even more important feature.

[0002] In typical microprocessor or microcontroller applications, the microprocessor or microcontroller gathers information from various sources to make a decision or measurement, for example, encephalography, security or sensor applications. Most of the information gathered reaches the microprocessor via an interrupt. Various techniques at both the architecture and circuit level have been investigated to maximize throughput and minimize latency of the computing system. These techniques typically lead to an increase in the total power dissipation of the system. In order to compensate for the increased power dissipation, techniques have been introduced to reduce system power consumption such as body biasing and clock gating, for example.

[0003] The performance of general purpose microcontroller or microprocessor systems is typically limited by the number of interrupts that need to be handled simultaneously. The design of these microcontroller systems typically requires a certain throughput to be able to handle the required number of simultaneous interrupts. To maintain adequate throughput requires a minimum supply voltage to be provided to the microcontroller system which then determines the power consumption of the microcontroller system.

SUMMARY

[0004] In accordance with the invention, power efficient computation is achieved while maintaining overall system throughput. This may be achieved by appropriately managing the computer system's operating voltage and frequency. To compensate for the loss of throughput due to the lowered operating voltage and frequency, processor parallelization is introduced into the system architecture by having more than one processor. An Intelligent Interrupt Distributer (IID) is provided in a computer system architecture in accordance with the invention to balance interrupts among the processors. In accordance with the invention, the computer system may be configured for either throughput optimization or reduced power consumption. If the voltage and frequency are not reduced, the throughput is increased because more than one processor is working However, the voltage and frequency may be appropriately reduced so that throughput remains the same as in the single processor configuration. Additionally, in accordance with the invention, the maximum throughput and minimum power mode can be configured to comply with the application requirements.

BRIEF DESCRIPTION OF THE DRAWINGS

[0005] FIG. **1** shows the overall reduction in power consumption in accordance with the invention.

[0006] FIG. **2** shows a prior art embodiment.

[0007] FIG. **3** shows an embodiment in accordance with the invention.

[0008] FIG. **4** shows an embodiment in accordance with the invention.

[0009] FIG. **5** shows interrupt scheduling in accordance with FIG. **2**.

[0010] FIG. **6** shows interrupt scheduling for an embodiment in accordance with the invention.

[0011] FIG. **7** shows interrupt scheduling for an embodiment in accordance with the invention.

DETAILED DESCRIPTION

[0012] In an embodiment in accordance with the invention, the minimum operating voltage is reduced by using an IID to distribute interrupts among multiple processors in a computer system while the computer system appears as a single processor system to the user. No change to the binary code is typically needed. In accordance with the invention, the computer system may be a microcontroller or microprocessor system, for example. The IID incorporates both static and dynamic tuning of the computer system voltage and frequency. The concept of the IID is based on the sharing of interrupts among the multiple processors. If the processor is in idle mode and not busy then the IID schedules the incoming interrupt to that processor. Power-aware scheduling algorithms for interrupts with and without priority constraints are used. Power-aware interrupt scheduling with priority constraints means that when multiple interrupts arrive at the IID, the interrupts are scheduled according to a predefined interrupt priority typically defined by the programmer. The IID receives all interrupts and distributes the interrupts among the multiple processors based on availability. This distribution of the interrupts among the multiple processors by the IID recovers time not used by one processor to reduce the total energy consumption of the system. In summary, the IID detects the interrupts from the peripheral devices, distributes the interrupts to the processors and adjusts the supply voltage going to the processors and adjusts the operating frequency of the processors.

[0013] The scaling (reduction) of voltage results in the reduction of the throughput in a processor. Hence, if one reduces the supply voltage to a processor in a system, the resulting reduction in throughput in the processor needs to be compensated for. In an embodiment in accordance with the invention, compensation is achieved by having processors in parallel. The number of processors (N) needed to compensate for a given reduction in throughput is given by the following equation:

$$N_{@Freq2} = \left\lceil \frac{Freq_{Vdd1}}{Freq_{Vdd2}} \right\rceil = \left\lceil \frac{V_{dd2}}{(V_{dd2} - V_{th})^{\alpha}} \frac{(V_{dd1} - V_{th})^{\alpha}}{V_{dd1}} \right\rceil \quad (1)$$

where $N_{@Freq1}=1$, Freq1 is the original frequency, V_{dd1} is the original supply voltage, V_{th} is the threshold voltage which is one characteristic of the transistors and the threshold voltage is defined as the minimum voltage that required to turn the transistor ON. Freq2 is the reduced frequency at the scaled supply voltage V_{dd2}. $\lceil\ \rceil$ is the ceiling function. The exponent a accounts for the velocity saturation of the transistors and may take on any value between one, complete velocity satu-

ration and two, no velocity saturation. As the number of processors operating in parallel is increased, there will be a capacitance overhead due to multiplexing. See, for example, A. P. Chandrakasan and R. W. Brodersen, *Low Power Digital CMOS Design,* Boston: Kluwer Academic Publishers (Now Springer), 1995 incorporated herein by reference.

[0014] Total switching capacitance in the multi-processor system, where N is the number of parallel processors is given by:

$$\frac{C_{new}}{C_{old}} = [N + \lambda(N-1)] \quad (2)$$

with C_{new} and C_{old} representing the switching capacitance of the scaled voltage system and the original voltage system, respectively. λ represents the overhead of the additional hardware (multiplexing, registers etc—see A. P. Chandrakasan and R. W. Brodersen incorporated by reference above). The scaled voltage system will run at N times lower frequency. Therefore, total power consumption in the system can be given by:

$$\frac{P_{Vdd2}}{P_{Vdd1}} = [N + \lambda(N-1)]\frac{V_{dd2}^2}{V_{dd1}^2} \times \frac{1}{N} \quad (3)$$

where P_{Vdd2} is the power consumption in the scaled voltage system with N processors and P_{Vdd1} is the power consumption in the original voltage system with one processor. FIG. **1** shows the overall reduction in power consumption achieved by this method. Graph **100** is based on data from CMOS90 process where the nominal supply voltage, V_{dd}=1.2V. The x-axis shows the supply voltage V_{dd}. Curve 110 is plotted against the y-axis on the left side. The y-axis on the left shows the number of processors needed as the supply voltage is decreased below 1.2 V to maintain the same throughput. For example, curve **110** shows that increasing the number of processors to 2 reduces the voltage from ~1.2 V to ~0.7 V and reduces the power consumption to 0.4 (normalized-a factor of 2.5 reduction). Curve **120** shows the power reduction (y-axis on the right) when voltage is reduced. The y-axis on the right shows the power consumption in the scaled voltage system as a ratio of the power consumed in the original voltage system.

[0015] FIG. **2** shows typical prior art single processor system **200** having single processor processor **210**, SRAM **220**, nonvolatile memory **230**, bus **240** for connecting nonvolatile memory **230** and SRAM **220** to processor **210**. Additionally, keyboard **250**, Universal Asynchronous Receiver/Transmitter (UART) **255**, Timer **260** and Analog-to-Digital Converter (ADC) **265** are connected to bus **245**. Clock Generation Unit (CGU) **290** connected to bus **245** is the generating clock for single processor system **200**. Note CGU **290** provides a fixed clock in the context of single processor system **200**. Peripheral interrupt line **285** directly connects keyboard **250** to processor **210**. Peripheral interrupt line **280** directly connects UART **255** to processor **210**. Peripheral interrupt line **275** directly connects ADC **265** to processor **210**. Peripheral interrupt line **270** directly connects timer **260** to processor **210**.

[0016] FIG. **3** shows an embodiment in accordance with the invention. Multiprocessor system **300** has processors **310** and **320** connected to bus **340**. Dual port SRAM **325** and dual port nonvolatile memory **330** are also connected to bus **340**. Bus **340** supports two masters, i.e. processors **310** and **320**. Additionally, IID **350** is connected to bus **340**. IID **350** distributes the interrupts between processors **310** and **320** on interrupt bus **315**. The width of interrupt bus **315** is the total number of interrupts supported by core **310** and core **320**. Based on the interrupts received, IID **350** adjusts both the frequency and the voltage. Line **304** carries the return from interrupt signal to IID **350** from processors **310** and **320** which indicates the completion of the interrupt to IID **350**. Clock Generation Unit (CGU) **390** provides dynamic clock gating and scaling for multiprocessor system **300** and is connected to bus **340**. IID **350** sends commands to CGU **390** to adjust the clock (frequency) for processor **310** and **320**. Dedicated clock lines **301**, **302**, **303**, **304**, **305**, **306**, **307**, **308** and **309** connect from CGU **390** to ADC **365**, Timer **360**, UART **355**, KBI **354**, Core **320**, Core **310**, SRAM **325**, nonvolatile memory **330** and IID **350**, respectively, to provide clock signals.

[0017] Keyboard Interface (KBI) **354**, Universal Asynchronous Receiver/Transmitter (UART) **355**, ADC **365** and Timer **360** are all connected to Advance Peripheral Bus (APB) **345** which is connected to bus **340**. Peripheral interrupt line **385** directly connects keyboard **355** to IID **350**. Peripheral interrupt line **380** directly connects UART **355** to IID **350**. Peripheral interrupt line **375** directly connects ADC **365** to IID **350**. Peripheral interrupt line **370** directly connects timer **360** to IID **350**. Note, that unlike in FIG. **2**, all peripheral interrupt lines **370**, **375**, **380** and **385** directly connect to IID **350** and not to cores **310**, **320**. Dual port SRAM **325** is used as it typically consumes less power than 2 single port SRAMs. Dual port nonvolatile memory (NV) **330** is used so that both processors **310** and **320** can execute interrupts.

[0018] Multiprocessor system **300** remains a "single processor system" from the point of view of the user. This means that the binary code for single processor system **200** typically does not need to be modified for execution on multiprocessor system **300**. IID **350** schedules interrupts between processors **310** and **320** by examining the workload of processors **310** and **320**. If processor **310** or **320** is free, the coming interrupt is scheduled for the free processor. Therefore, the hardware changes introduced in multiprocessor system **300** are typically transparent to the user and the user can typically replace single processor system **200** with multiprocessor system **300** without any modifications.

[0019] FIG. **4** shows multiprocessor system **400** in an embodiment in accordance with the invention with analog to digital converter (ADC) **420** having connection **412** to external a first external temperature sensor (not shown) and ADC **425** having connection **414** to second external temperature sensor (not shown). Dedicated clock lines **401**, **402**, **403**, **404**, **405**, **406**, **407**, **408**, **409**, **410** and **411** connect from CGU **490** to ADC **420**, Timer **460**, ADC **425**, Timer **461**, UART **415**, KBI **413**, SRAM **325**, nonvolatile memory **330**, Core **480**, Core **485** and IID **450**, respectively. KBI **413** is connected to a keyboard (not shown) and UART **415** is connected to the user (not shown). KBI **413**, ADCs **420**, **421**, Timers **460**, **461** and UART **415** are all connected to APB bus **445**. When a temperature sample is available from one of the two temperature sensors, the relevant ADC, ADC **420** or ADC **425** sends an interrupt on peripheral interrupt **475** or **476**, respectively, to either processor **480** or processor **485** which is intercepted by IID **450**. IID **450** routes the interrupt to the first available processor and the temperature sample is stored in memory SRAM **325** by either processor **480** or processor **485**. Every 1 ms, for example, the temperature samples stored in SRAM

325 are filtered and restored in SRAM 325. When requested via KBI 413 by the user on the keyboard, the last filtered temperature samples are provided to the user via UART 415. All interrupt service routines (ISR) or interrupt handlers are stored in the dual port nonvolatile memory 330. If there were "n" processors and each processor had a nonvolatile memory, then one could use an "n" port nonvolatile memory or a separate nonvolatile memory for each processor. When processor 480 or processor 485 completes the execution of an interrupt, the return from interrupt signal 304 is set to 1. IID 450 checks return from interrupt signal 304 to identify which processor, processor 480 or processor 485 is free or idle. If both processors 480 and 485 are free or idle, the interrupt is sent to processor 480, for example.

[0020] With reference to FIG. 3, IID 350 primarily operates to schedule interrupts between processor 310 and processor 320. In an example shown in FIG. 5, assume that four interrupts 501, 502, 503 and 504 are coming to IID 350 and the execution times are 4 seconds, 3 seconds, 3 seconds and 2 seconds, respectively. Total execution time for the 4 interrupts in single processor system 200 on processor 210 is 12 seconds.

[0021] FIG. 6 shows the scheduling of the same interrupts 501, 502, 503 and 504 on multi-processor system 300. IID 350 first sends interrupt 501 to processor 310 and interrupt 502 to processor 320. Because interrupt 502 is completed after 3 seconds, the next interrupt, interrupt 503 is sent to processor 320 by IID 350 because processor 310 is still busy. Interrupt 504 is sent to processor 310 by IID 350. From FIG. 6 it can be seen that the total time required to handle the 4 interrupts in multiprocessor system 300 is 6 seconds, i.e. half the time required in single processor system 200.

[0022] However, in accordance with the invention, the purpose of having a multi-processor system is to reduce the power consumption while keeping the throughput the same as in single processor system 200 (12 seconds in this example). FIG. 7 shows an embodiment in accordance with the invention. Here, IID 350 reduces the operating frequency (the voltage is also reduced from 1.2 to 0.7) of processor 310 and processor 320 by a factor of two which results in interrupts 501 and 504 taking 12 seconds to complete and in interrupts 502 and 503 also taking 12 seconds to complete in parallel. To the user, the embodiment in FIG. 7 still appears to be single processor system 200. However, the embodiment in FIG. 7 uses less power than the embodiment in FIG. 5. FIG. 1 shows that the power consumption of the embodiment in FIG. 7 is about one third of the power consumption of the embodiment shown in FIG. 5 though both take the same total execution time.

[0023] Note that if the object is to increase throughput, it is advantageous to increase the number of processors but that two processors is typically the optimum solution for reducing power consumption in accordance with the invention.

1. A multiprocessor system comprising:
a first processor;
a second processor coupled to the first processor by a first bus;
a second bus coupled to the first bus;
a clock gating unit directly coupled to the first processor and the second processor;
a first and second peripheral device coupled to the second bus; and
an intelligent interrupt distributor coupled to the first bus and directly coupled to the first and second processor by an interrupt bus and directly coupled to the first and second peripheral device by a first and second peripheral interrupt line and directly coupled to the clock gating unit such that an interrupt received on the first or the second peripheral interrupt line is distributed via the interrupt line to one of the first and the second processors which is in an idle state.

2. The multiprocessor system of claim 1 wherein the intelligent interrupt distributor adjusts an operating frequency and voltage of the first and the second processors by sending commands to the clock gating unit to maintain a throughput that is the same as that for an equivalent single processor system.

3. The multiprocessor system of claim 1 further comprising a nonvolatile memory capable of storing interrupt service routines and coupled to the first bus.

4. The multiprocessor system of claim 3 wherein the nonvolatile memory is a multiport memory.

5. The multiprocessor system of claim 1 further comprising an SRAM memory coupled to the first bus.

6. The multiprocessor system of claim 5 where the SRAM memory is a multiport memory.

7. The multiprocessor system of claim 1 where the second bus is an Advanced Peripheral Bus.

8. The multiprocessor system of claim 1 where the first peripheral device is an analog to digital converter.

9. The multiprocessor system of claim 1 where the second peripheral device is a UART.

10. The multiprocessor system of claim 1 where the intelligent interrupt distributor statically tunes the operating frequency and voltage of the multiprocessor system.

11. A method for making a multiprocessor system comprising:
providing a first processor;
providing a second processor coupled to the first processor by a first bus;
providing a second bus coupled to the first bus;
providing a clock gating unit directly coupled to the first processor and the second processor;
providing a first and second peripheral device coupled to the second bus; and
an intelligent interrupt distributor coupled to the first bus and directly coupled to the first and second processor by an interrupt bus and directly coupled to the first and second peripheral device by a first and second peripheral interrupt line and directly coupled to the clock gating unit such that an interrupt received on the first or the second peripheral interrupt line is distributed via the interrupt line to one of the first and the second processors which is in an idle state.

* * * * *