



US008103889B2

(12) **United States Patent**
Naik

(10) **Patent No.:** **US 8,103,889 B2**
(45) **Date of Patent:** ***Jan. 24, 2012**

(54) **REDUCING POWER CONSUMPTION OF A MICROPROCESSOR**

(56) **References Cited**

(75) Inventor: **Parag Naik**, Karnataka State (IN)
(73) Assignee: **Tamiras Per Pte. Ltd., LLC**, Dover, DE (US)
(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 384 days.
This patent is subject to a terminal disclaimer.

U.S. PATENT DOCUMENTS

5,625,644	A *	4/1997	Myers	375/242
5,790,874	A	8/1998	Takano	
5,890,005	A *	3/1999	Lindholm	713/320
6,195,756	B1 *	2/2001	Hurd	713/340
6,335,718	B1 *	1/2002	Hong et al.	345/98
6,463,521	B1	10/2002	Long	
6,725,450	B1 *	4/2004	Takayama	717/139
6,909,358	B2 *	6/2005	Gass	340/146.2
6,938,248	B2 *	8/2005	Kitakami et al.	717/159
6,959,395	B2 *	10/2005	Ma	713/310
7,287,149	B2	10/2007	Okamoto	
7,386,844	B2 *	6/2008	Heishi et al.	717/161
7,409,001	B2 *	8/2008	Ionescu et al.	375/267
2002/0120830	A1	8/2002	Matsuo	
2004/0015922	A1 *	1/2004	Kitakami et al.	717/154
2005/0010830	A1	1/2005	Webster	
2005/0055563	A1 *	3/2005	Fischer et al.	713/194

* cited by examiner

(21) Appl. No.: **12/335,137**

(22) Filed: **Dec. 15, 2008**

(65) **Prior Publication Data**

US 2009/0177902 A1 Jul. 9, 2009

Related U.S. Application Data

(63) Continuation of application No. 11/251,586, filed on Oct. 13, 2005, now Pat. No. 7,480,809.

(60) Provisional application No. 60/619,060, filed on Oct. 15, 2004.

(51) **Int. Cl.**
G06F 1/00 (2006.01)

(52) **U.S. Cl.** 713/320; 717/139

(58) **Field of Classification Search** 713/300-340;
717/139

See application file for complete search history.

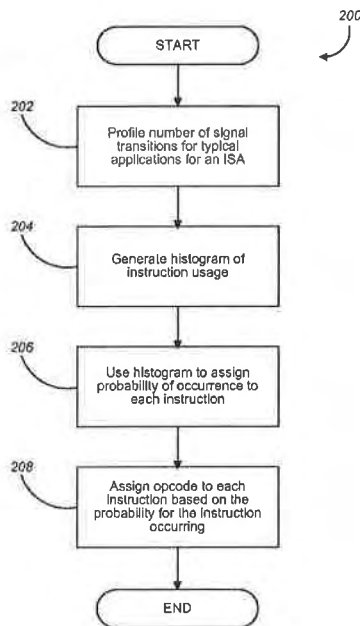
Primary Examiner — Paul R Myers

(74) *Attorney, Agent, or Firm* — Christensen O'Connor Johnson Kindness PLLC

(57) **ABSTRACT**

Methods and apparatus, including computer program products, implementing and using techniques for reducing the power consumption of a microprocessor. One or more signal transitions in an instruction set of a microprocessor are profiled. A probability of occurrence is assigned to each instruction in the instruction set. A binary operation code is assigned to each instruction, based on the probability of occurrence for the instruction. The instructions having the highest probability of occurrence are assigned operation codes that require fewer signal transitions. As a result, the power consumption of the microprocessor is reduced.

16 Claims, 3 Drawing Sheets



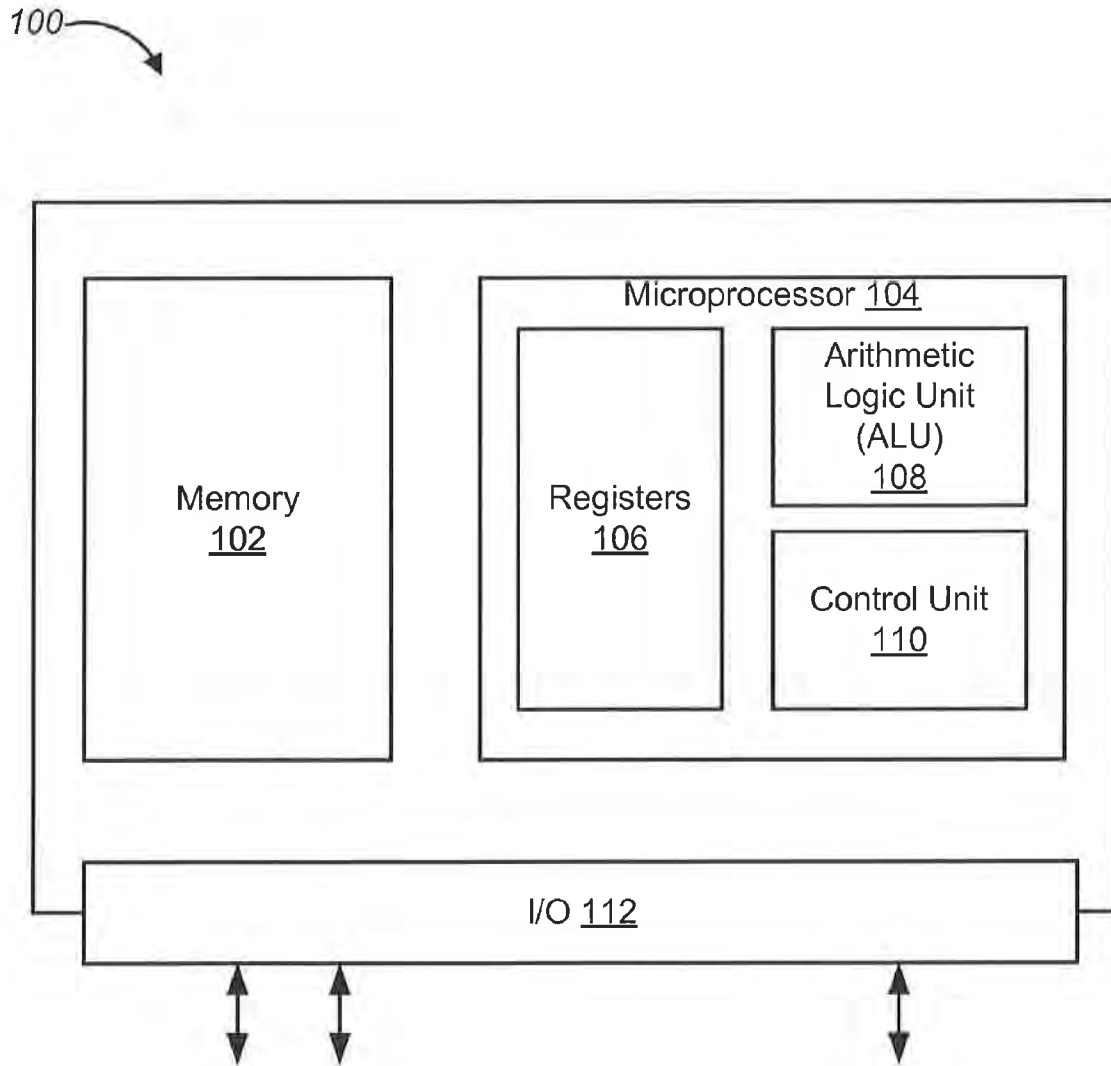


FIG. 1

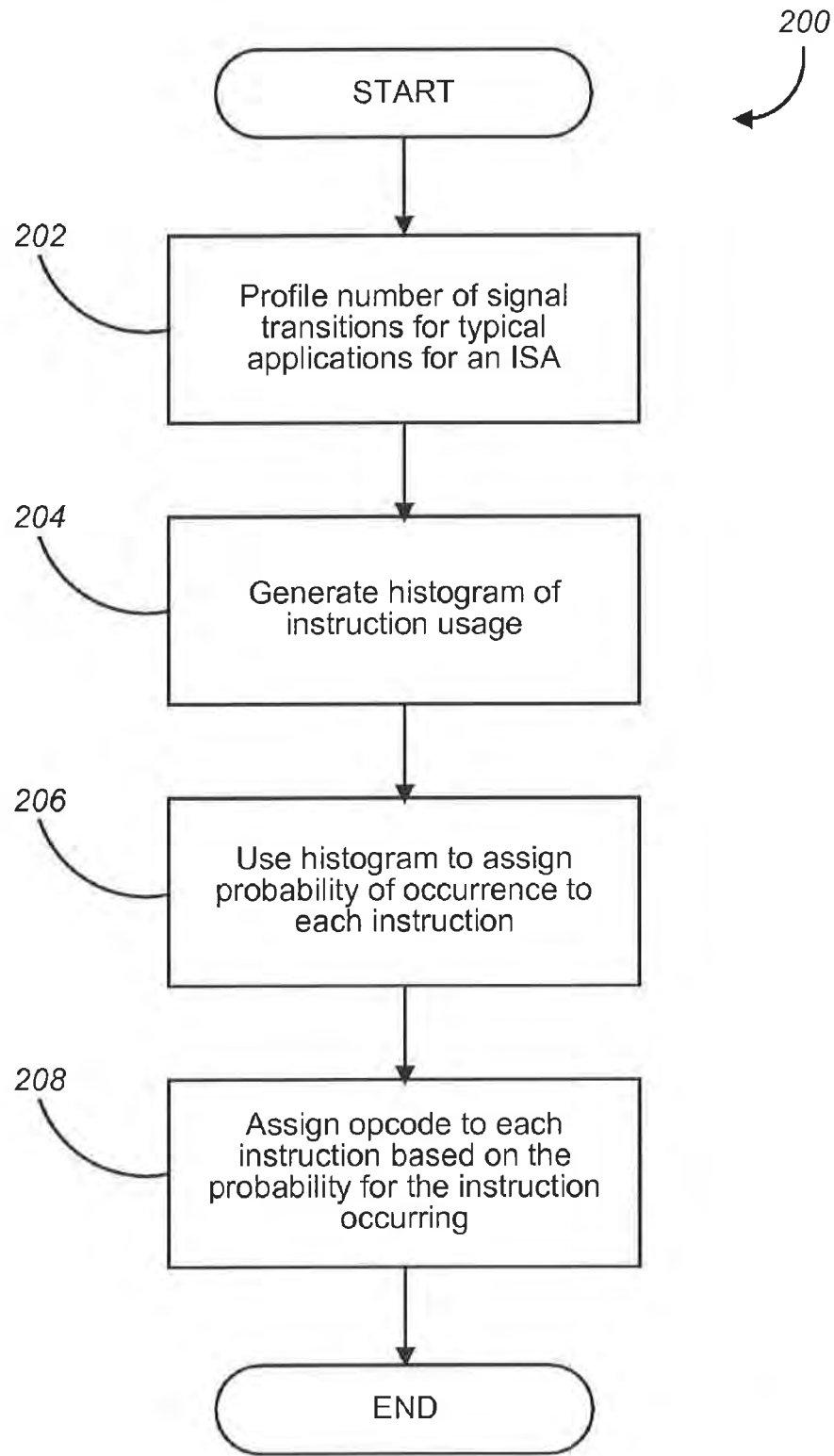


FIG. 2

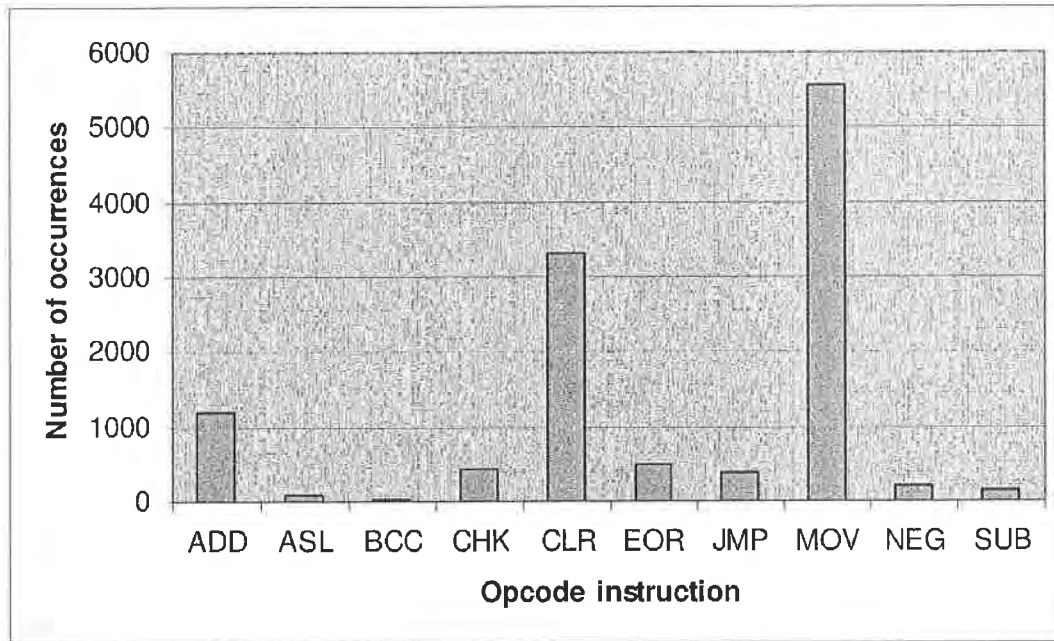


FIG. 3A

Opcode Instruction	Number of occurrences	Likelihood (%)
ADD	1200	10.13
ASL	75	0.63
BCC	20	13.33
CHK	436	3.68
CLR	3333	28.14
EOR	496	4.19
JMP	378	3.19
MOV	5555	46.91
NEG	200	1.69
SUB	150	1.27
TOTAL	11843	100

FIG. 3B

REDUCING POWER CONSUMPTION OF A MICROPROCESSOR

CROSS-REFERENCE TO RELATED APPLICATIONS

This application is a continuation application and claims priority under 35 U.S.C. §120 of U.S. patent application Ser. No. 11/251,586, filed Oct. 13, 2005 now U.S. Pat. No. 7,480,809, issued Jan. 20, 2009, which claimed benefit under 35 U.S.C. §119(e) from U.S. Provisional Application No. 60/619,060 entitled "METHOD OF ASSIGNING OPCODES IN AN ISA OF A MICROPROCESSOR FOR REDUCING POWER CONSUMPTION," filed Oct. 15, 2004, the entire disclosures of which are incorporated herein by reference for all purposes.

BACKGROUND

This invention relates to reducing the power consumption of a microprocessor.

Microprocessors used in computers and other types of computing devices are typically power-sensitive components. That is, due to the large number of operations that a microprocessor performs, it is desirable to keep its power consumption per operation as low as possible, in order to reduce the energy usage of the device incorporating the microprocessor. In a computer, operation codes (opcodes) operate on registers, values in memory, values stored on the stack, the I/O ports, the bus, and so on. The opcodes are used to perform arithmetic operations and to move and change values. The opcodes operate on operands. Microprocessors convert the opcodes into sets of binary bits and perform operations using these sets of bits. Each bit in a set can have an on-state (typically represented as '1') or an off-state (typically represented as '0'). Eight bits are usually referred to as a byte on most conventional processors, and two bytes are usually referred to as a word.

The basic set of commands, or instructions, that a microprocessor understands is termed a command set, or an instruction set. Various types of microprocessors can have differently sized instruction sets, and this difference often constitutes one of the main differences between the microprocessors. For example, RISC (Reduced Instruction Set Computer) microprocessors have relatively small instruction sets, whereas CISC (Complex Instruction Set Computers) processors have relatively large instruction sets.

When designing an Instruction Set Architecture (ISA) for a microprocessor, little thought is typically given to how the respective instructions in the opcode that is used in the microprocessor are assigned to bit sets. The number of bits in an opcode instruction typically depends on the total number of instructions that can be carried out by the microprocessor, and which bits are assigned '0's and '1's, respectively, is usually arbitrarily determined. In most microprocessors, 6-8 bits correspond to an opcode instruction. There is always a one to one correspondence between a particular opcode instruction and a set of bits forming a corresponding machine code instruction.

One of the most common operations of a microprocessor is to fetch instructions from a memory external to the microprocessor for execution by the microprocessor. This is a power intensive operation, as the number of signal transitions is high. Thus, in order to reduce the power consumption of the microprocessor, it would be desirable to construct the ISA in

such a way that the number of signal transitions in its most common operations is as low as possible.

SUMMARY

In general, in one aspect, the invention provides methods and apparatus, including computer program products, implementing and using techniques for reducing the power consumption of a microprocessor. One or more signal transitions in an instruction set of a microprocessor are profiled. A probability of occurrence is assigned to each instruction in the instruction set. A binary operation code is assigned to each instruction, based on the probability of occurrence for the instruction. The instructions having the highest probability of occurrence are assigned operation codes that require fewer signal transitions. As a result, the power consumption of the microprocessor is reduced.

Advantageous implementations can include one or more of the following features. The profiling step can include compiling a set of microprocessor instructions into an assembly file, and executing the assembly file through a pattern matching software tool which calculates the occurrences of the instructions. A histogram showing the signal transitions occurring during operation of the microprocessor can be generated. The instruction with the highest probability of occurring can be assigned a binary operation code consisting of only zeros.

In general, in another aspect, the invention provides a microprocessor. The microprocessor includes a set of registers, an arithmetic logic unit, and a control unit. The set of registers stores temporarily data during operation of the microprocessor. The arithmetic logic unit executes arithmetic and logical operations of the microprocessor. The control unit interprets instructions contained in a program that is executed by the microprocessor. The instructions form a subset of the microprocessor's instruction set architecture, and each instruction in the microprocessor's instruction set architecture has a binary operation code that is assigned to the instruction, based on the probability of occurrence for the instruction during operation of the microprocessor.

The invention can be implemented to include one or more of the following advantages. By assigning opcodes in the ISA of the microprocessor such that the most common opcodes have the lowest number of signal transitions, the power usage of the microprocessor can be significantly reduced at an architectural level.

The details of one or more embodiments of the invention are set forth in the accompanying drawings and the description below. Other features, objects, and advantages of the invention will be apparent from the description and drawings, and from the claims.

DESCRIPTION OF DRAWINGS

FIG. 1 shows a schematic diagram of a computer incorporating a microprocessor in accordance with the invention.

FIG. 2 shows a schematic diagram of a process for assigning opcodes to instructions in a microprocessor at design time.

FIG. 3A shows an exemplary histogram illustrating the number of occurrences of various opcode instructions in a microprocessor.

FIG. 3B shows an exemplary table illustrating the likelihood of occurrences of various opcode instructions in a microprocessor, corresponding to the histogram in FIG. 3A.

Like reference symbols in the various drawings indicate like elements.

DETAILED DESCRIPTION

Reference will now be made in detail to a particular embodiment of the invention an example of which is illustrated in the accompanying drawings. While the invention will be described in conjunction with the particular embodiment, it will be understood that it is not intended to limit the invention to the described embodiment. To the contrary, it is intended to cover alternatives, modifications, and equivalents as may be included within the spirit and scope of the invention as defined by the appended claims.

FIG. 1 shows a schematic block diagram of the functional units of a computer (100). As can be seen in FIG. 1, the computer (100) includes three main components: a microprocessor (104), a memory (102), and an input/output (I/O) system (112). The microprocessor (104) includes three main parts: the arithmetic logic unit (ALU) (108) where all arithmetic and logical operations take place, the registers (106) where data is temporarily stored, and the control unit (110), which interprets the instructions contained in the program that is being executed by the microprocessor (104). The memory (102) contains the program (i.e., a sequence of instructions) and the data to be processed by the microprocessor (104). The content in the memory (102) can be read as well as modified by the microprocessor (104). The I/O units (112) are used by the computer (100) to communicate with the outside world, as is well known by those of ordinary skill in the art.

As was described above, the microprocessor (104) performs operations using binary bits, and there is a one-to-one correspondence between a particular bit combination and a specific opcode instruction. That is, whenever a particular instruction is to be performed, the microprocessor (104) typically must change the states of one or more of the 6-8 bits to select the proper combination of zeroes and ones that form the opcode instruction. This switching of bits consumes power. Furthermore, in most conventional microprocessors, about 20% of the available instructions in the opcode set are performed about 80% of the time. Thus, by assigning specific bit combinations to the most common instructions, which require as few switches of bits as possible, the power consumption of the microprocessor (104) can be significantly reduced. A process (200) for reducing the power consumption of a microprocessor will now be described with reference to FIG. 2. In the implementation shown in FIG. 2, the process (200) is conducted at design time of the microprocessor.

As shown in FIG. 2, the process (200) starts by profiling the number of signal transitions for the various operations in an ISA (step 202). In an implementation for an existing microprocessor, the profiling can be done by taking a sample set of instructions that the microprocessor would execute when executing a program. This code is compiled into an assembly file, which is executed through a pattern matching software tool. If no compiler is available for the processor architecture, profiles are generated for other, similar, architectures. A histogram is then generated of the instruction usage of the microprocessor (step 204), for example, using the pattern matching software tool recited above, which calculates the occurrences of the instructions and plots the histogram. An exemplary histogram can be seen in FIG. 3A, where the number of occurrences for each opcode instruction in the microprocessor's instruction set has been plotted. It should be noted that the number of opcode instructions and the number of occurrences illustrated in FIG. 3A have been significantly reduced,

in order to more clearly illustrate the principles of the invention. As can be seen in FIG. 3A, the most common opcode instructions are "BCC", "CLR", and "MOV."

Returning now to FIG. 2, based on the histogram, a probability of occurrence is assigned to each instruction (step 206). FIG. 3B shows a table corresponding to the histogram in FIG. 3A, and where the probability has been calculated in the right hand column. As can be seen in FIG. 3B, the probabilities for the three most common instructions are 46.91% for the "MOV" instruction, 28.14% for the "CLR" instruction, and 13.33% for the "BCC" instruction. Again, it should be noted that this is merely a hypothetical example illustrating the principles of the invention. In a situation where no compiler is available, the histogram is generated based on the selected similar processor architectures and a first guess for opcode assignments is made as described below. Subsequently this guess can be refined when a compiler becomes available on the new architecture.

Again returning to FIG. 2, in the last step of the process (200) an opcode is assigned to each instruction set, based on the probability for the instruction occurring (step 208), which completes the process. For example, if the profiling in step (202) shows that the 'MOV' instruction has the highest likelihood of occurring, then the assigning step (206) will assign bits '000000' or '111111' to the 'MOV' instruction, assuming that the opcode range is 6 bits, since neither of these assignments involve any signal transitions. The second most likely instruction can be encoded using the bit assignment that was not used by the most likely instruction, that is, '000000' or '111111,' respectively. The third most likely instruction can be encoded such that only a single signal transitions occurs, and so on.

As the person skilled in the art realizes, this architectural solution may allow the power consumption of the microprocessor to be significantly reduced and can be implemented at design time of the microprocessor. In another implementation, existing processors can be made compatible with the invention by adding extra circuitry that converts the existing bit assignments for the various instructions in the opcode set into bit assignments in accordance with the present invention.

The invention can be implemented in digital electronic circuitry, or in computer hardware, firmware, software, or in combinations of them. An apparatus of the invention can be implemented in a computer program product tangibly embodied in a machine-readable storage device for execution by a programmable processor; and method steps of the invention can be performed by a programmable processor executing a program of instructions to perform functions of the invention by operating on input data and generating output. The invention can be implemented advantageously in one or more computer programs that are executable on a programmable system including at least one programmable processor coupled to receive data and instructions from, and to transmit data and instructions to, a data storage system, at least one input device, and at least one output device. Each computer program can be implemented in a high-level procedural or object-oriented programming language, or in assembly or machine language if desired; and in any case, the language can be a compiled or interpreted language. Suitable processors include, by way of example, both general and special purpose microprocessors. Generally, a processor will receive instructions and data from a read-only memory and/or a random access memory. Generally, a computer will include one or more mass storage devices for storing data files; such devices include magnetic disks, such as internal hard disks and removable disks; magneto-optical disks; and optical disks. Storage devices suitable for tangibly embodying com-

5

puter program instructions and data include all forms of non-volatile memory, including by way of example semiconductor memory devices, such as EPROM, EEPROM, and flash memory devices; magnetic disks such as internal hard disks and removable disks; magneto-optical disks; and CD-ROM disks. Any of the foregoing can be supplemented by, or incorporated in, ASICs (application-specific integrated circuits).

To provide for interaction with a user, the invention can be implemented on a computer system having a display device such as a monitor or LCD screen for displaying information to the user. The user can provide input to the computer system through various input devices such as a keyboard and a pointing device, such as a mouse, a trackball, a microphone, a touch-sensitive display, a transducer card reader, a magnetic or paper tape reader, a tablet, a stylus, a voice or handwriting recognizer, or any other well-known input device such as, of course, other computers. The computer system can be programmed to provide a graphical user interface through which computer programs interact with users.

Finally, the processor optionally can be coupled to a computer or telecommunications network, for example, an Internet network, or an intranet network, using a network connection, through which the processor can receive information from the network, or might output information to the network in the course of performing the above-described method steps. Such information, which is often represented as a sequence of instructions to be executed using the processor, may be received from and outputted to the network, for example, in the form of a computer data signal embodied in a carrier wave. The above-described devices and materials will be familiar to those of skill in the computer hardware and software arts.

A number of implementations of the invention have been described. Nevertheless, it will be understood that various modifications may be made without departing from the spirit and scope of the invention. For example, a similar concept can be applied to data and not only to instructions. For example, if data represented by 16 bits, the frequency of use for all possible bit combinations (i.e., 2^{16} combinations) can be studied in a similar fashion and be used to create various power-saving data retrieval schemes. Furthermore, the invention has been described by way of example with respect to a microprocessor and a memory, but it should be realized that the inventive concept can be extended to any two semiconductor devices that exchange a limited instruction set, for which a frequency of use can be determined. Accordingly, other embodiments are within the scope of the following claims.

The invention claimed is:

1. A computer-implemented method, comprising:
 - profiling one or more signal transitions in an instruction set of a microprocessor, the instruction set including a plurality of instructions, wherein each instruction includes a plurality of bits;
 - assigning a respective probability of occurrence to at least two instructions in the instruction set; and
 - assigning a respective binary operation code to each of the at least two instructions in the instruction set based on the probability of occurrence for the instruction, wherein the instruction having the highest probability of occurrence is assigned a binary operation code comprising fewer signal transitions between adjacent bits.
2. The method of claim 1, wherein said profiling includes:
 - compiling a set of microprocessor instructions into an assembly file; and

6

executing the assembly file through a pattern matching software tool which calculates the occurrences of the instructions.

3. The method of claim 1, further comprising:

- generating a histogram showing the signal transitions occurring during operation of the microprocessor.

4. The method of claim 1, wherein the instruction having the highest probability of occurrence is assigned a binary operation code consisting of zeros but not ones or ones but not zeros.

5. The method of claim 1, wherein the probability of occurrence is an absolute probability of occurrence.

6. The article of manufacture of claim 1, wherein the probability of occurrence is an absolute probability of occurrence.

7. An article of manufacture including a computer-readable medium having instructions stored thereon that, in response to execution by a computing device, cause the computing device to perform operations comprising:

- profiling one or more signal transitions in an instruction set of a microprocessor, the instruction set including a plurality of instructions, wherein each instruction includes a plurality of bits;

- assigning a respective probability of occurrence to at least two instructions in the instruction set; and

- assigning a respective binary operation code to each of the at least two instructions in the instruction set based on the probability of occurrence for the instruction, wherein the instruction of the at least two instructions having the highest probability of occurrence is assigned a binary operation code comprising fewer signal transitions between adjacent bits.

8. The article of manufacture of claim 7, wherein the instructions further cause the computing device to perform operations comprising:

- compiling a set of microprocessor instructions into an assembly file; and

- executing the assembly file through a pattern matching software tool which calculates the occurrences of the instructions.

9. The article of manufacture of claim 7, wherein the instructions further cause the computing device to perform operations comprising:

- generating a histogram showing the signal transitions occurring during operation of the microprocessor.

10. The article of manufacture of claim 7, wherein the instruction *n* the instruction set having the highest probability of occurrence is assigned a binary operation code consisting of zeros but not ones or ones but not zeros.

11. A computer, comprising:

- a microprocessor including a control unit; and
- a memory storing a program thereon, wherein the program includes a set of instructions that are executable by the microprocessor, the set of instructions including a plurality of instructions, wherein each instruction includes a plurality of bits;

wherein the control unit of the microprocessor is communicatively coupled to the memory and configured to interpret the set of instructions contained in the program, and wherein at least two instructions in the set of instructions are each assigned a respective binary operation code that is based on a probability that the respective instruction will occur during operation of the microprocessor, wherein the instruction of the at least two instructions having the highest probability of occurrence is assigned a binary operation code comprising fewer signal transitions between adjacent bits.

7

12. The computer of claim 11, wherein the instruction having the highest probability of occurrence is assigned a binary operation code that uses fewer bit transitions.

13. The computer of claim 12, wherein the instruction having the highest probability of occurrence is assigned a binary operation code consisting of zeros but not ones or ones but not zeros.

14. The microprocessor, comprising:

a control unit configured to interpret a plurality of instructions contained in a set of instructions that are executable by the microprocessor, wherein each of the instructions includes a plurality of bits;

wherein the control unit is further configured to assign respective operation codes to each of the instructions in the set of the instructions based on a probability that the

8

respective instructions will be executed by the microprocessor during operation of the microprocessor; and wherein instructions having a higher probability of being executed by the microprocessor are assigned respective operation codes that use fewer transitions between bits than instructions having a lower probability of being executed by the microprocessor.

15. The microprocessor of claim 14, wherein the control unit is configured to assign an operation code consisting of zeros but not ones or ones but not zeros to the instruction having the highest probability of being executed by the microprocessor.

16. The microprocessor of claim 14, wherein the probability of occurrence is an absolute probability of occurrence.

* * * * *