

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
8 September 2006 (08.09.2006)

PCT

(10) International Publication Number
WO 2006/092792 A2

- (51) International Patent Classification: Not classified
- (21) International Application Number: PCT/IL2006/000280
- (22) International Filing Date: 1 March 2006 (01.03.2006)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
 - 60/657,412 2 March 2005 (02.03.2005) US
 - 60/657,414 2 March 2005 (02.03.2005) US
 - 60/667,022 1 April 2005 (01.04.2005) US

(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, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, 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).

(71) Applicant (for all designated States except US): MPLIC-ITY LTD. [IL/IL]; P.o. Box 58160, 61580 Tel Aviv (IL).

(72) Inventors; and

(75) Inventors/Applicants (for US only): DAGAN, Eran [IL/IL]; 7 Mevo Gilat Street, 69359 Tel Aviv (IL). VINITZKY, Gil [IL/IL]; 33 Ehad Haam Street, 58014 Azur (IL).

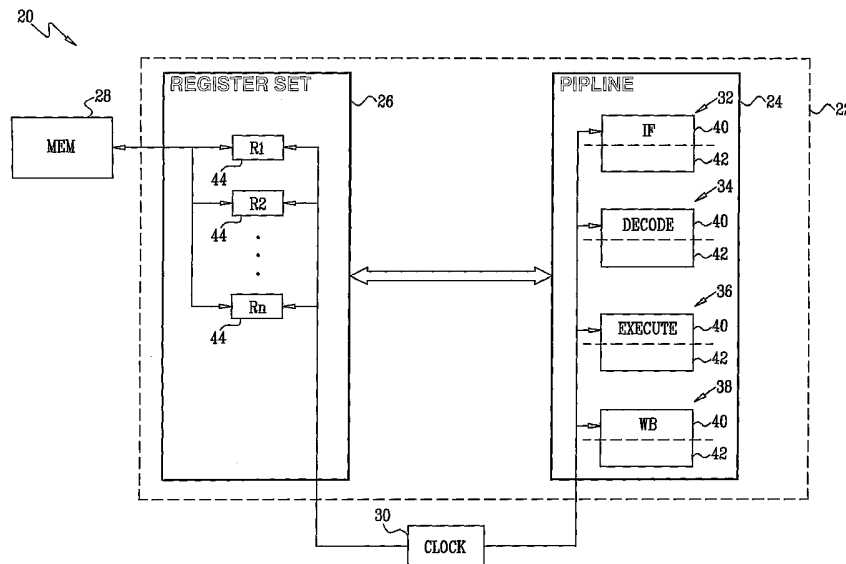
Published:

— without international search report and to be republished upon receipt of that report

(74) Agents: SANFORD T. COLB & CO. et al.; P.o. Box 2273, 76122 Rehovot (IL).

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.

(54) Title: EFFICIENT MACHINE STATE REPLICATION FOR MULTITHREADING



(57) Abstract: An electronic processing device (20) includes a digital processing circuit (24), which is configured to process multiple threads in alternation using respective context data of the multiple threads, and a clock circuit (30), which is operative to generate a clock signal for timing the alternation of the multiple threads. A register replication circuit (26), having a single clock input, is coupled to receive the clock signal, and comprises a main storage element (50) for holding and outputting to the digital processing circuit the context data of one thread and at least one shadow storage element (52) for holding the context data of at least one other thread, the main and shadow storage elements being connected in cascade so as to exchange the context data held in the main and shadow storage elements responsively to the clock signal received via the single clock input.

WO 2006/092792 A2

EFFICIENT MACHINE STATE REPLICATION FOR MULTITHREADING

CROSS-REFERENCE TO RELATED APPLICATIONS

This application claims the benefit of U.S. Provisional Patent Applications 60/657,412 and 60/657,414, both filed March 2, 2005, and of U.S. Provisional Patent Application
5 60/667,022, filed April 1, 2005. The disclosures of these related applications are incorporated herein by reference.

FIELD OF THE INVENTION

The present invention relates generally to integrated circuit architectures, and specifically to architectures and circuits that may be used to support multithreading in an
10 integrated digital processing device.

BACKGROUND OF THE INVENTION

Multithreading is commonly used to enhance the performance of modern microprocessors and programming languages. Multithreading may be defined as the logical separation of a processing task into independent threads, which are activated individually and
15 require limited interaction or synchronization between threads. In a pipelined processor, for example, the pipeline stages may be controlled to process two or more threads in alternation and thus use the pipeline resources more efficiently. Each thread has its own processing context, which is held in a set of registers that are accessed by the processor. Typically, a context switching circuit switches the register sets so that the processor is able to access the
20 appropriate context when processing each of the threads.

Various methods and circuits for context switching are known in the art. For example, U.S. Patent 5,142,677, whose disclosure is incorporated herein by reference, describes an electronic processor that is operable in alternative processing contexts identified by a context
25 signal. First and second registers are connected to the electronic processor to participate in one processing context while retaining information from another processing context until a return thereto. A context switching circuit is connected to the first and second registers and operates to selectively control input and output operations of the registers to and from the electronic processor depending on the processing context. Exemplary circuits for “zero-overhead interrupt context switching” are shown in Figs. 22 and 23.

30 As another example, U.S. Patent 6,247,040, whose disclosure is incorporated herein by reference, describes a method and structure for automated switching between multiple contexts

in a storage subsystem target device. The active context register set and inactive context register set are rapidly and automatically swapped by operation of a state machine model to resume or start processing of an inactive context. Additional inactive contexts are stored in a buffer memory associated with the target device controller.

5 U.S. Patent Application Publication US 2005/0081018 A1, whose disclosure is incorporated herein by reference, describes a register file bit that includes a primary latch and a secondary latch with a feedback path. A context switch mechanism allows a fast context switch when execution changes from one thread to the next. A bit value for a second thread of execution is stored in the primary latch, and is then transferred to the secondary latch. The bit
10 value for a first thread of execution is then written to the primary latch. When a context switch is needed (when the first thread stalls and the second thread needs to begin execution), the register file bit can perform a context switch from the first thread to the second thread in a single clock cycle. The register file bit contains a backup latch inside the register file itself so that minimal extra wire paths are needed to or from the existing register file.

15 U.S. Patent Application Publication US 2003/0046517 A1, whose disclosure is incorporated herein by reference, describes apparatus for facilitating multithreading in a computer processor pipeline. The pipeline is controlled by a control mechanism, which is statically scheduled to execute multiple threads in round-robin succession. This static scheduling eliminates the need for communication between stages of the pipeline.

20 U.S. Patent Application Publication US 2003/0135716 A1, whose disclosure is incorporated herein by reference, describes a method for converting a computer processor configuration having a k-phased pipeline into a virtual multithreaded processor. For this purpose, each pipeline phase of the processor configuration is divided into a plurality of sub-phases, and at least one virtual pipeline with k sub-phases is created within the pipeline. In this
25 manner, a single physical processor can be made to operate as multiple virtual processors, each equivalent to the original processor.

SUMMARY OF THE INVENTION

Embodiments of the present invention provide methods and circuitry that may be used in efficient multithreaded designs, including conversion of single-thread designs to multithread
30 operation. These embodiments may be applied not only to custom processing units and pipelines, but also in semi-custom integrated circuit devices, such as structured application-

specific integrated circuits (ASICs) and gate arrays, including field-programmable gate arrays (FPGAs).

In some embodiments of the present invention, multithreading circuitry comprises a novel register replication circuit, comprising a main storage element in cascade with one or more shadow storage elements. The main storage element holds and outputs the context of one thread, while each shadow storage element holds the context of another thread that is waiting to be output. The register replication circuit is configured so that a single clock line may be used to drive all the elements of the circuit. This novel design approach simplifies the timing of the circuitry and reduces chip size and power consumption.

There is therefore provided, in accordance with an embodiment of the present invention, an electronic processing device, including:

a digital processing circuit, which is configured to process multiple threads in alternation using respective context data of the multiple threads;

a clock circuit, which is operative to generate a clock signal for timing the alternation of the multiple threads; and

a register replication circuit, which has a single clock input, which is coupled to receive the clock signal, and includes a main storage element for holding and outputting to the digital processing circuit the context data of one thread and at least one shadow storage element for holding the context data of at least one other thread, the main and shadow storage elements being connected in cascade so as to exchange the context data held in the main and shadow storage elements responsively to the clock signal received via the single clock input.

In a disclosed embodiment, the digital processing circuit includes one or more pipeline stages, which are split into multiple phases corresponding to the multiple threads.

In some embodiments, the main and shadow storage elements respectively include main and shadow flip-flops; each having a respective input and a respective output, wherein the output of the main flip-flop is coupled to the input of the shadow flip-flop, and the output of the shadow flip-flop is coupled to the input of the main flip-flop. Additionally or alternatively, the at least one shadow storage element includes a plurality of cascaded shadow storage elements, which include at least first and second shadow storage elements, each of the main and shadow storage elements having a respective input and a respective output, and the output of the main storage element is coupled to the input of the first shadow storage element, while the output of the second shadow storage element is coupled to the input of the main storage element.

Further additionally or alternatively, each of the main and shadow storage elements has a respective input and a respective output, and the register replication circuit includes multiplexing logic, which is coupled between the output of the at least one shadow storage element and the input of the main storage element and is further coupled to receive input data and an enable signal, and which is operative, responsively to the enable signal, to write the input data to the main storage element instead of the context data held in the at least one shadow storage element. In a disclosed embodiment, the multiplexing logic is further coupled to connect, responsively to a control signal, the output of at least one of the main and shadow storage elements to the input of the same at least one of the main and shadow storage elements so as to cause the at least one of the main and shadow storage elements to hold the same context data over multiple cycles of the clock signal.

In some embodiments, the digital processing circuit is fabricated as part of a semi-custom integrated circuit (IC) device having user-configurable cells, and the register replication circuit is implemented in the user-configurable cells. In one embodiment, the semi-custom IC includes a structured application-specific integrated circuit (ASIC). In another embodiment, the semi-custom IC includes a field-programmable gate array (FPGA).

There is also provided, in accordance with an embodiment of the present invention, a method for producing an electronic processing device, including: -

configuring a digital processing circuit to process multiple threads in alternation using respective context data of the multiple threads;

generating a clock signal for timing the alternation of the multiple threads;

coupling a register replication circuit to receive the clock signal via a single clock input, the register replication circuit including a main storage element for holding and outputting to the digital processing circuit the context data of one thread and at least one shadow storage element for holding the context data of at least one other thread; and

connecting the main and shadow storage elements in cascade so as to exchange the context data held in the main and shadow storage elements responsively to the clock signal received via the single clock input.

There is additionally provided, in accordance with an embodiment of the present invention, a method for producing an electronic processing device, including:

providing a device design for a multithreaded digital processing circuit for implementation as part of a semi-custom integrated circuit (IC), which includes predefined functional components including user-configurable cells;

incorporating in the device design sufficient registers and thread scheduling logic so as to support multithreaded operation of the digital processing circuit;

porting the device design to the functional components of the semi-custom IC, so as to generate a gate-level design in which the user-configurable cells are configured to implement at

5 least the thread scheduling logic; and

implementing the gate-level design in the semi-custom IC.

In one embodiment, the semi-custom IC includes a structured application-specific IC (ASIC), which includes pre-designed blocks and the user-configurable cells. In another embodiment, the semi-custom IC includes a field-programmable gate array (FPGA), and

10 providing the device design includes implementing the digital processing circuit using the user-configurable cells.

In some embodiments, incorporating the registers and thread scheduling logic includes adding a register replication circuit for holding context data of multiple threads processed by the digital processing circuit, the register replication circuit including a main storage element

15 for holding and outputting to the digital processing circuit the context data of one thread and at least one shadow storage element for holding the context data of at least one other thread. Typically, incorporating the registers and thread scheduling logic includes providing a clock signal for timing alternation of the multiple threads in the digital processing circuit, and coupling the register replication circuit to receive the clock signal via a single clock input.

20 There is further provided, in accordance with an embodiment of the present invention, an electronic processing device, including a semi-custom integrated circuit (IC), which includes predefined functional components including user-configurable cells,

wherein a device design including a multithreaded digital processing circuit is implemented in the semi-custom IC, and the design includes sufficient registers and thread

25 scheduling logic to support multithreaded operation of the digital processing circuit, and

wherein the device design is ported to the functional components of the semi-custom IC so as to generate a gate-level design in which the user-configurable cells are configured to implement at least the thread scheduling logic, and the gate-level design is implemented in the

semi-custom IC.

30 The present invention will be more fully understood from the following detailed description of the embodiments thereof, taken together with the drawings in which:

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram that schematically illustrates elements of a microprocessor that is configured for multithreading, in accordance with an embodiment of the present invention;

5 Fig. 2 is an electrical schematic diagram showing details of a register replication circuit, in accordance with an embodiment of the present invention;

Fig. 3 is a timing diagram that schematically illustrates signals in the circuit of Fig. 2, in accordance with an embodiment of the present invention;

Fig. 4 is an electrical schematic diagram showing details of a register replication circuit, in accordance with another embodiment of the present invention;

10 Fig. 5 is an electrical schematic diagram showing details of a register replication circuit, in accordance with yet another embodiment of the present invention;

Fig. 6 is a block diagram that schematically illustrates a structured ASIC device that is configured for multithreading, in accordance with an embodiment of the present invention;

15 Fig. 7 is a flow chart that schematically illustrates a method for producing a structured ASIC device with multithreading capability, in accordance with an embodiment of the present invention; and

Fig. 8 is a block diagram that schematically illustrates a FPGA that is configured for multithreading, in accordance with an embodiment of the present invention.

DETAILED DESCRIPTION OF EMBODIMENTS

20 Fig. 1 is a block diagram that schematically illustrates elements of a microprocessor 20, in accordance with an embodiment of the present invention. This is a simplified view, which is meant only to aid in understanding the principles of the present invention, and thus includes only those elements that are relevant to the operation of these principles. Incorporation of these elements in an actual microprocessor (or in any other synchronous programmable or non-

25 programmable design) will be apparent to those skilled in the art based upon the description that follows. Although a particular pipeline architecture is shown in Fig. 1, this architecture is chosen simply for convenience and clarity of explanation, and the principles of the present invention may similarly be applied in substantially any architectural framework that supports multithreading.

30 Microprocessor 20 comprises a processing core 22, which comprises a processing pipeline 24 and a register set 26. The core elements communicate with a memory 28 and a clock circuit 30, as well as with other elements not shown in the figure. Pipeline 24 comprises

a sequence of stages including an instruction fetcher (IF) 32, a decoder 34, an execution engine 36, and a writeback (WB) stage 38.

In order to configure pipeline 24 for multithreading while maintaining the original design frequency of the microprocessor (i.e., with each thread running at the original design frequency), each stage of the pipeline is split into first and second sub-stages (or phases) 40 and 42. Splitting operations of this sort are described in the above-mentioned U.S. Patent Application Publications US 2003/0046517 A1 and US 2003/0135716 A1. Typically, a logic storage element is inserted in the design between the two sub-stages. During a given clock cycle, sub-stage 40 can then process an instruction belonging to a first thread, while sub-stage 42 processes an instruction belonging to another thread. During the next clock cycle, sub-stage 42 completes the processing of the instruction belonging to the first thread, while sub-stage 40 begins processing the next instruction of the other thread. Clock circuit 30 may drive pipeline 24 at twice the nominal clock speed of microprocessor 20, so that both threads are processed at the nominal, single-thread throughput of the pipeline.

Although this example relates to interleaved dual-thread operation, each stage in the pipeline may alternatively be split into three or more sub-stages, so as to permit a larger number of threads to be processed concurrently. Further alternatively, the design may be adapted for multithreading using the circuitry described hereinbelow without splitting the pipeline stages, in which case each thread in the multithreaded processor will run at half (or less) of the original design frequency.

Each of the threads that is processed by pipeline 24 has its own context, which is held in register set 26 and accessed by the pipeline stages during processing. To enable the interleaving of the threads in the pipeline, the register set comprises register replication circuits 44, corresponding to the original registers (R1, R2, ..., Rn) of the original microprocessor design. Each circuit 44 holds the contexts of both of the executing threads and switches the context that is made available to the pipeline stages at the (accelerated) clock rate of the pipeline. For proper multithread operation, the context switching performed by the register replication circuits must be carefully synchronized with the pipeline.

Fig. 2 is an electrical circuit diagram that schematically shows details of register replication circuit 44, in accordance with an embodiment of the present invention. This figure (as well as Figs. 4 and 5, described hereinbelow) shows a single cell of circuit 44, which is designed to store a single bit of the context of both of the executing threads. This cell is

typically replicated multiple times in each circuit 44, depending on the number of bits held in the corresponding register.

5 Circuit 44 comprises a main storage element 50 and a shadow storage element 52. In this embodiment, the storage elements comprise flip-flops, although other suitable devices may similarly be used for this purpose. The two flip-flops are both driven by a single clock line (CLK). During each clock cycle, the context bit held in main storage element 50 is available for output (DOUT), to be used by the active thread in pipeline 24. Shadow storage element 52 meanwhile holds the corresponding context bit for the other thread. The contents of the main and shadow storage elements are exchanged at each clock cycle, so that the contexts of the two
10 executing threads are available in alternation at the output of circuit 44. The use of a single clock line for both the main and shadow registers simplifies the timing of register set 26 and reduces the chip area and power that must be consumed for this purpose. Another advantage of circuit 44 is that there is no thread-selection multiplexer in the critical output path, so that timing is simplified, and the output DOUT is ready at the beginning of each clock cycle.

15 A multiplexer 54 permits new data to be loaded into main storage element 50. Normally, when the multiplexer is in state 0, the contents of shadow storage element 52 are transferred to main storage element 50 at each clock cycle. When the multiplexer is enabled to state 1, however, a new input bit on data input DIN is loaded into main storage element 50. The main storage element may have other inputs and features (not shown in the figure), such as
20 a reset input. The shadow storage element, however, may comprise a flip-flop of minimal size and complexity, since the only data and control connections it requires are the clock and the D and Q terminals, which are connected inside circuit 44. This architectural characteristic is useful in reducing the size and power consumption of circuit 44.

25 Fig. 3 is a timing diagram that schematically illustrates signals in circuit 44, in accordance with an embodiment of the present invention. In this embodiment, it is assumed that circuit 44 is used in processing of two threads, identified as thread 0 and thread 1, which access the contents of main storage element 50 during alternating clock cycles, labeled phase 0 and phase 1. At the initial phase 0, data D0 are valid for thread 0 and the enable select control (EN) of multiplexer 54 is active. As a result, the multiplexer selects D0 from input DIN, and
30 passes D0 for sampling by main storage element 50. At phase 1, the sampled D0 is transmitted to output DOUT and is sampled by shadow storage element 52. At the next phase 0, if control EN is inactive, multiplexer 54 restores D0 into main storage element 50.

Fig. 4 is an electrical schematic diagram that schematically shows details of a register replication circuit 60, in accordance with another embodiment of the present invention. Whereas the architecture of circuit 44, shown in the preceding figures, is designed for context switching between threads on every clock cycle, the architecture of circuit 60 is designed to satisfy the requirements of other multithreading schemes with more flexible timing requirements. Like circuit 44, circuit 60 has a single clock input for driving both the main and shadow storage elements.

In addition to main storage element 50, shadow storage element 52, and multiplexer 54, which operate in the manner describe above, circuit 60 comprises further multiplexing logic in the form of multiplexers 62 and 64, for determining when context data should be exchanged between the storage elements. When the KEEP input to circuit 60 is active, shadow storage element 52 continues to hold the same data over successive clock cycles, rather than accepting the data from main storage element 50. Similarly, when the BYPASS input is active and the ENABLE input is inactive, main storage element 50 continues to hold the same data over successive clock cycles, rather than accepting the data from shadow storage element 52. Otherwise, the main and shadow storage elements exchange data and receive new data as described above with reference to Fig. 2.

The multiplexing logic in circuit 60 thus determines which thread context will be stored in each of the main and shadow storage elements. In this manner, one of the threads may be held in "sleep mode" while the other thread remains active. Pipeline 24 may then be used to process the active thread at accelerated speed. The sleep mode may be invoked, for example, when one of the threads stalls following a cache miss or enters a wait state for some other reason. In this case, when the wait state is resolved (when the required data arrive from memory, for instance), the sleeping thread is reactivated. The scheduling capability provided by the multiplexing logic in circuit 60 can also be used to allocate additional resources (in the form of extra computing cycles) to a thread with a greater computational load, or to save power by maintaining an inactive thread in sleep mode.

Fig. 5 is an electrical schematic diagram that schematically shows details of a register replication circuit 70, in accordance with yet another embodiment of the present invention. Although the circuits of the preceding embodiments are directed to dual-threaded architectures, the principles of these embodiments may be extended in a straightforward manner to N-threaded designs, wherein $N \geq 3$. Circuit 70 is one example of such a design. It is similar in design to circuit 44, but includes three shadow storage elements 72, 74 and 76 in cascade with

main storage element 50. All the storage elements are driven by the same clock input. This arrangement will serve four interleaved threads.

Fig. 6 is a block diagram that schematically illustrates a structured ASIC device 80, which is configured for multithreaded operation in accordance with an embodiment of the present invention. The type of structured ASIC shown in this figure is an integrated circuit that contains predefined functional components, including pre-designed blocks and user-configurable areas 96, which can be connected to the pre-designed blocks. The functions and connectivity of the user-configurable areas can be redefined by changing the upper layers of the chip. The user-configurable areas may use "tiles," look-up tables (LUTs), as in a FPGA, or predefined logic gates. Device 80 is shown solely by way of example, and the principles of the present invention may similarly be applied to substantially any type of semi-custom integrated circuit, including other types of ASICs (structured or non-structured) and other gate arrays, including both field-programmable and factory-programmable types.

In the example shown in Fig. 6, the pre-designed blocks of device 80 include a processing unit 82 (such as a CPU), as well as a DMA controller 84, a PLL 86, SRAM 99 and analog circuits 90. Other pre-designed blocks include I/O interfaces 92 and a data bus 94. As noted above, the functional components of the ASIC also include user-configurable cells, which are available in user areas 96 for use by designers in implementing their own logic blocks and interconnects. In the example shown in Fig. 6, a portion of user area 96 has been configured by a user to host a digital signal processor (DSP) 97.

ASIC devices and programmable gate arrays are generally not designed for multithreaded operation because of the added complexity involved. Multithreading requires additional clock and control lines, which can take up considerable area when implemented in the predefined gate structure of the ASIC or programmable gate array. The efficient register replication circuits described above, however, are themselves area-efficient and require a small number of clock and control lines in comparison with multithreading architectures that are known in the art. Furthermore, semi-custom device architectures commonly include a large number of flip-flops in order to achieve enhanced timing performance. When a user design is ported to the device, many of these flip-flops typically turn out to be redundant. These extra flip-flops may be used advantageously for storing several machine states in multithreaded implementations. Therefore, embodiments of the present invention make multithreaded ASIC and other gate array designs practical.

In the example shown in Fig. 6, sets of register replication circuits 98 and associated clock and control connections are implemented in user areas 96, adjacent to pre-designed processing unit 82 and to DSP 97. (Alternatively, register replication circuits may be associated only with a pre-designed block or only with a user-defined processing circuit.)

5 Register replication circuits 98 typically have a form similar to the register replication circuits shown in the preceding figures. These elements permit the processing unit to be configured for multithreaded operation in the manner described above. The user of device 80 is thus able to obtain enhanced performance from the existing components of the ASIC device, at relatively low cost in terms of chip real estate and power consumption. Alternatively, device 80 may be
10 configured for multithreaded operation using other context storage and switching circuits, such as those described in some of the patents cited in the Background of the invention. In this latter case, the design may still gain some benefit from multithreading, though typically at higher cost in terms of real estate and/or power consumption.

Fig. 7 is a flow chart that schematically illustrates a method for producing a structured
15 ASIC with a multithreaded design, in accordance with an embodiment of the present invention. Although this method is described, for the sake of clarity, with specific reference to device 80, it may similarly be applied, *mutatis mutandis*, to other types of semi-custom integrated circuits. The method takes as its starting point an initial micro-architectural design that is to be ported to the structured ASIC device, which is provided at an initial design step 100. The design may be
20 prepared in a suitable design language, such as register transfer language (RTL), or it may have already been synthesized in the form of a netlist.

The initial micro-architectural design may be either single-threaded or multithreaded, as determined at a design classification step 101. If the design is single-threaded, it is converted to a multithreaded design in the steps that follow. In the alternative, an experienced designer may
25 use the techniques described herein to create a multithreaded design *ab initio*, in which case the process skips ahead to step 110.

The design blocks that are to be modified for multithreaded operation are subjected to static timing analysis, at a timing testing step 102. In each processing stage that is to be converted to multithreaded operation, the designer inserts a splitter, to divide the stage into two
30 (or more) successive phases, at a splitting step 104. The static timing analysis indicates where to place the splitters in order to achieve optimal timing performance with a minimal number of added splitters. (Step 104 assumes that the micro-architectural design and the structured ASIC device itself are capable of acceleration by splitting the processing stages. In the alternative,

step 104 may be omitted, and the design may be adapted for multithreading by means of the steps that follow without necessarily splitting processing stages.)

In place of each register used by the processing stage, the designer inserts a register replication circuit, such as circuits 98 (Fig. 6), at a register replication step 106. The designer also adds clock lines and thread scheduling logic and connections in order to synchronize the split processing stage and the corresponding register replication circuits, at a scheduling design step 108.

The splitters, register replication circuits, and logic and connections are added to the original netlist. These elements are then converted to gate-level designs, using appropriate gates in user area 96 of the ASIC device, at a logic synthesis step 110. This step is typically performed automatically, using available tools that are known in the art, such as the "Blast Create" and "Blast Fusion" products offered by Magma Design Automation (Santa Clara, California), or the "Amplify" product offered by Synplicity (Sunnyvale, California). The ASIC manufacturer uses the resulting gate-level design to generate production masks, at a mask generation step 112. In the structured ASIC process, the lower-layer masks are fixed in advance, and usually only the upper metal layers are customized at step 112. These masks are then used in fabricating the customized ASIC device, at a production step 114.

Fig. 8 is a block diagram that schematically illustrates a FPGA device 120, which is configured for multithreaded operation in accordance with an embodiment of the present invention. Device 120 may be, for example, a member of the "Virtex" family of FPGAs offered by Xilinx (San Jose, California). The device comprises an array of cells 122, which the user may configure in software to implement substantially any suitable logic design. Typically, as noted above, the FPGA device hardware includes a large number of redundant flip-flops, which may be exploited advantageously in creating an efficient multithreaded implementation at steps 104-108 (Fig. 7). In addition, device 120 typically comprises dedicated areas for functions such as I/O 124 and memory 126.

In the example shown in Fig. 8, the user has designed a logic processing unit 128, and this unit has been split for multithreaded operation in the manner described hereinabove. The user design also includes register replication circuits 130 and thread scheduling logic 132. The user typically designs these elements in a high-level design language, such as RTL. Automated logic synthesis software, which may be provided by the FPGA manufacturer or by a third party, ports the design to specific cells 122 of the FPGA itself, as shown in Fig. 8.

The use of multithreading in this fashion enables the user to get enhanced performance from the FPGA, at reduced cost in terms of cell utilization. For example, in the initial design, logic processing unit 128 may have been duplicated in its entirety to enable the device to process two data flows in parallel. Multithreading both data flows through a single logic processing unit means that the second logic processing unit can be eliminated from the design. The number of additional cells required by register replication circuits 130 and thread scheduling logic 132 is typically much smaller than the number of cells that would have been required for replication of the logic processing unit. Furthermore, the bottleneck in FPGA-based designs is usually in the logic LUT utilization and not in flip-flop utilization. The user may then be able to use a smaller FPGA or add further functionality to the chosen FPGA, and may thus reduce the cost or enhance the performance of the product in question.

Although the examples described above relate to certain specific types of devices, the principles of the present invention may advantageously be applied in semi-custom devices of other types, as well as in full-custom processors. It will thus be appreciated that the embodiments described above are cited by way of example, and that the present invention is not limited to what has been particularly shown and described hereinabove. Rather, the scope of the present invention includes both combinations and subcombinations of the various features described hereinabove, as well as variations and modifications thereof which would occur to persons skilled in the art upon reading the foregoing description and which are not disclosed in the prior art.

CLAIMS

1. An electronic processing device, comprising:
 - a digital processing circuit, which is configured to process multiple threads in alternation using respective context data of the multiple threads;
 - 5 a clock circuit, which is operative to generate a clock signal for timing the alternation of the multiple threads; and
 - a register replication circuit, which has a single clock input, which is coupled to receive the clock signal, and comprises a main storage element for holding and outputting to the digital processing circuit the context data of one thread and at least one shadow storage element for
10 holding the context data of at least one other thread, the main and shadow storage elements being connected in cascade so as to exchange the context data held in the main and shadow storage elements responsively to the clock signal received via the single clock input.
2. The device according to claim 1, wherein the digital processing circuit comprises one or more pipeline stages, which are split into multiple phases corresponding to the multiple threads.
- 15 3. The device according to claim 1, wherein the main and shadow storage elements respectively comprise main and shadow flip-flops, each having a respective input and a respective output, and wherein the output of the main flip-flop is coupled to the input of the shadow flip-flop, and the output of the shadow flip-flop is coupled to the input of the main flip-flop.
- 20 4. The device according to claim 1, wherein the at least one shadow storage element comprises a plurality of cascaded shadow storage elements, which comprise at least first and second shadow storage elements, each of the main and shadow storage elements having a respective input and a respective output, and wherein the output of the main storage element is coupled to the input of the first shadow storage element, while the output of the second shadow
25 storage element is coupled to the input of the main storage element.
5. The device according to claim 1, wherein each of the main and shadow storage elements has a respective input and a respective output, and wherein the register replication circuit comprises multiplexing logic, which is coupled between the output of the at least one shadow storage element and the input of the main storage element and is further coupled to receive
30 input data and an enable signal, and which is operative, responsively to the enable signal, to

write the input data to the main storage element instead of the context data held in the at least one shadow storage element.

6. The device according to claim 5, wherein the multiplexing logic is further coupled to connect, responsively to a control signal, the output of at least one of the main and shadow storage elements to the input of the same at least one of the main and shadow storage elements so as to cause the at least one of the main and shadow storage elements to hold the same context data over multiple cycles of the clock signal.

7. The device according to any of the preceding claims, wherein the digital processing circuit is fabricated as part of a semi-custom integrated circuit (IC) device having user-configurable cells, and wherein the register replication circuit is implemented in the user-configurable cells.

8. The device according to claim 7, wherein the semi-custom IC comprises a structured application-specific integrated circuit (ASIC).

9. The device according to claim 7, wherein the semi-custom IC comprises a field-programmable gate array (FPGA).

10. A method for producing an electronic processing device, comprising:
configuring a digital processing circuit to process multiple threads in alternation using respective context data of the multiple threads;
generating a clock signal for timing the alternation of the multiple threads;
coupling a register replication circuit to receive the clock signal via a single clock input, the register replication circuit comprising a main storage element for holding and outputting to the digital processing circuit the context data of one thread and at least one shadow storage element for holding the context data of at least one other thread; and
connecting the main and shadow storage elements in cascade so as to exchange the context data held in the main and shadow storage elements responsively to the clock signal received via the single clock input.

11. The method according to claim 10, wherein the digital processing circuit comprises one or more pipeline stages, and wherein configuring the digital processing circuit comprises splitting the pipeline stages into multiple phases corresponding to the multiple threads.

12. The method according to claim 10, wherein the main and shadow storage elements respectively comprise main and shadow flip-flops, each having a respective input and a

respective output, and wherein the output of the main flip-flop is coupled to the input of the shadow flip-flop, and the output of the shadow flip-flop is coupled to the input of the main flip-flop.

5 13. The method according to claim 10, wherein coupling the register replication circuit comprises cascading a plurality of shadow storage elements, which comprise at least first and second shadow storage elements, each of the main and shadow storage elements having a respective input and a respective output, wherein the output of the main storage element is coupled to the input of the first shadow storage element, while the output of the second shadow storage element is coupled to the input of the main storage element.

10 14. The method according to claim 10, wherein each of the main and shadow storage elements has a respective input and a respective output, and wherein coupling the register replication circuit comprises coupling multiplexing logic between the output of the at least one shadow storage element and the input of the main storage element, and connecting the multiplexing logic to receive input data and an enable signal and, responsively to the enable
15 signal, to write the input data to the main storage element instead of the context data held in the at least one shadow storage element.

15. The method according to claim 14, wherein coupling the multiplexing logic comprises connecting, responsively to a control signal, the output of at least one of the main and shadow storage elements to the input of the same at least one of the main and shadow storage elements
20 so as to cause the at least one of the main and shadow storage elements to hold the same context data over multiple cycles of the clock signal.

16. The method according to any of claims 10-15, wherein the digital processing circuit is fabricated as part of a semi-custom integrated circuit (IC) method having user-configurable cells, and wherein coupling the register replication circuit comprises implementing the register
25 replication circuit in the user-configurable cells.

17. The method according to claim 16, wherein the semi-custom IC comprises a structured application-specific integrated circuit (ASIC).

18. The method according to claim 16, wherein the semi-custom IC comprises a field-programmable gate array (FPGA).

30 19. A method for producing an electronic processing device, comprising:

providing a device design for a multithreaded digital processing circuit for implementation as part of a semi-custom integrated circuit (IC), which comprises predefined functional components including user-configurable cells;

5 incorporating in the device design sufficient registers and thread scheduling logic so as to support multithreaded operation of the digital processing circuit;

porting the device design to the functional components of the semi-custom IC, so as to generate a gate-level design in which the user-configurable cells are configured to implement at least the thread scheduling logic; and

implementing the gate-level design in the semi-custom IC.

10 20. The method according to claim 19, wherein the semi-custom IC comprises a structured application-specific IC (ASIC), which comprises pre-designed blocks and the user-configurable cells.

15 21. The method according to claim 19, wherein the semi-custom IC comprises a field-programmable gate array (FPGA), and wherein providing the device design comprises implementing the digital processing circuit using the user-configurable cells.

23. The method according to any of claims 19-21, wherein incorporating the registers and thread scheduling logic comprises adding a register replication circuit for holding context data of multiple threads processed by the digital processing circuit, the register replication circuit comprising a main storage element for holding and outputting to the digital processing circuit
20 the context data of one thread and at least one shadow storage element for holding the context data of at least one other thread.

24. The method according to claim 23, wherein incorporating the registers and thread scheduling logic comprises providing a clock signal for timing alternation of the multiple threads in the digital processing circuit, and coupling the register replication circuit to receive
25 the clock signal via a single clock input.

25. An electronic processing device, comprising a semi-custom integrated circuit (IC), which comprises predefined functional components including user-configurable cells,

wherein a device design comprising a multithreaded digital processing circuit is implemented in the semi-custom IC, and the design comprises sufficient registers and thread
30 scheduling logic to support multithreaded operation of the digital processing circuit, and

wherein the device design is ported to the functional components of the semi-custom IC so as to generate a gate-level design in which the user-configurable cells are configured to implement at least the thread scheduling logic, and the gate-level design is implemented in the semi-custom IC.

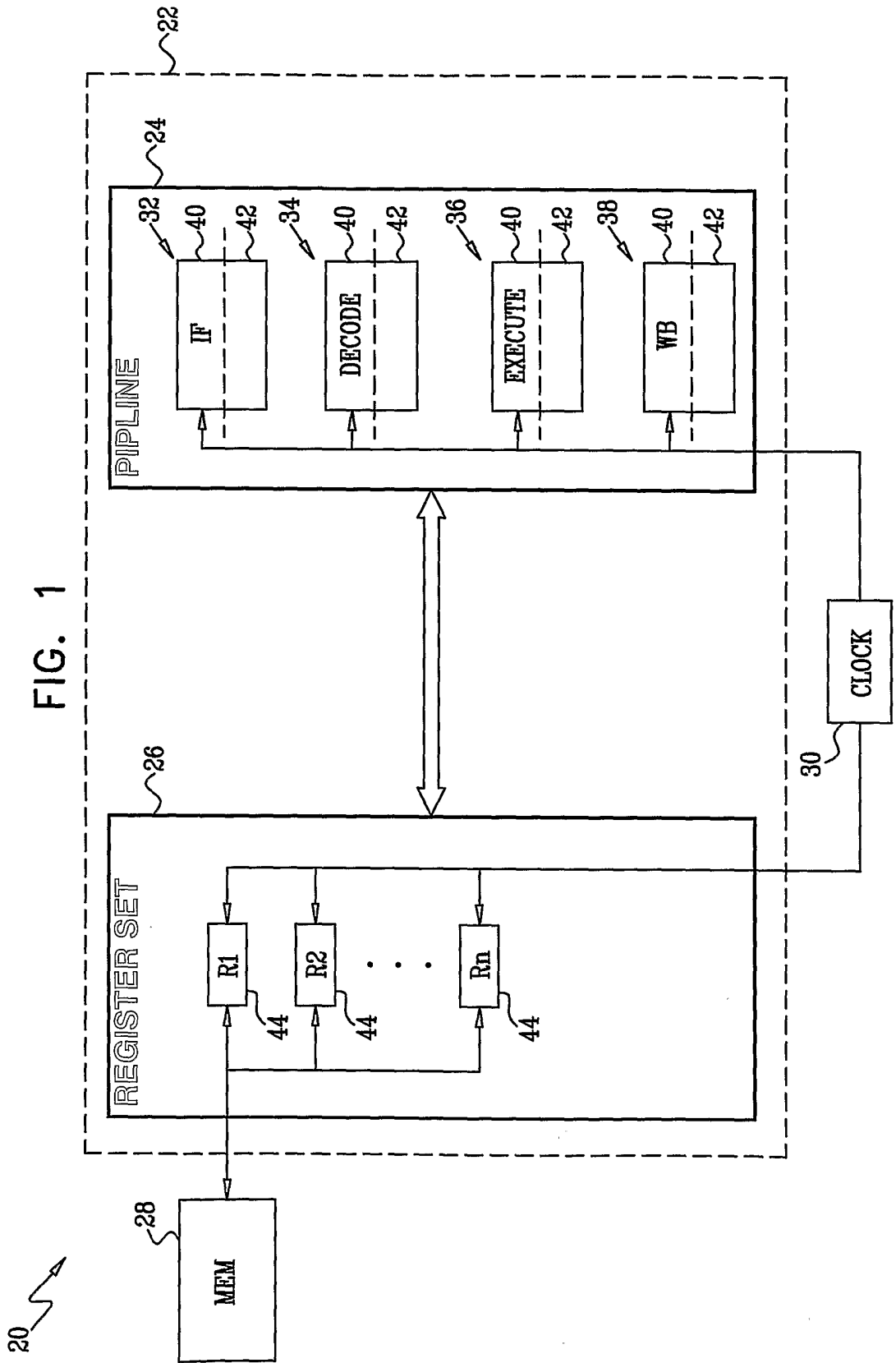
5 26. The device according to claim 25, wherein the semi-custom IC comprises a structured application-specific IC (ASIC), which comprises pre-designed blocks and the user-configurable cells.

10 27. The device according to claim 25, wherein the semi-custom IC comprises a field-programmable gate array (FPGA), and wherein the digital processing circuit is implemented using the user-configurable cells of the FPGA.

15 28. The device according to any of claims 25-27, wherein the registers and thread scheduling logic comprise a register replication circuit for holding context data of multiple threads processed by the digital processing circuit, the register replication circuit comprising a main storage element for holding and outputting to the digital processing circuit the context data of one thread and at least one shadow storage element for holding the context data of at least one other thread.

20 29. The device according to claim 28, wherein the registers and thread scheduling logic are configured to provide a clock signal for timing alternation of the multiple threads in the digital processing circuit, and to couple the clock signal to the register replication circuit via a single clock input.

FIG. 1



2/6

FIG. 2

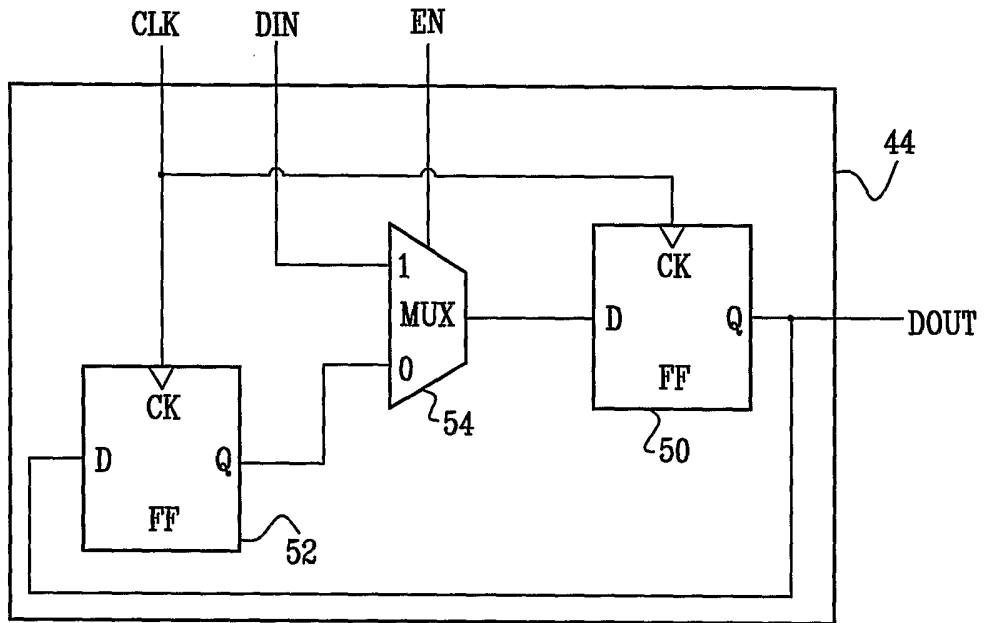
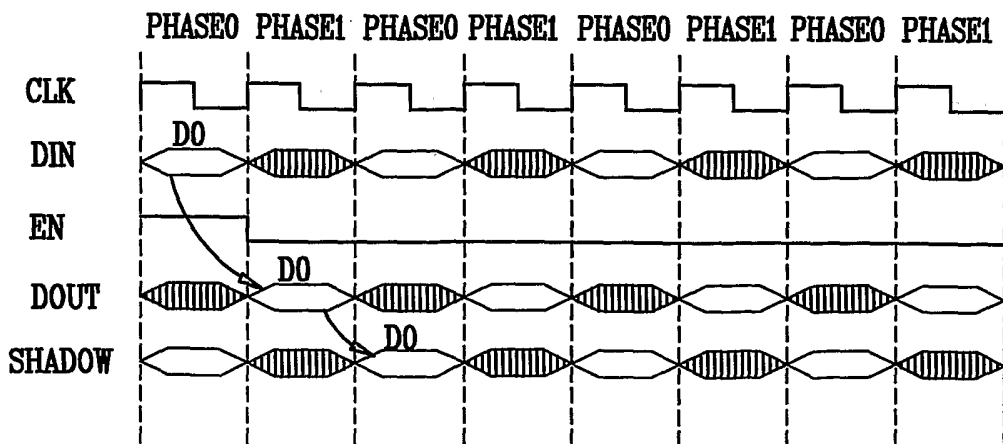


FIG. 3



3/6

FIG. 4

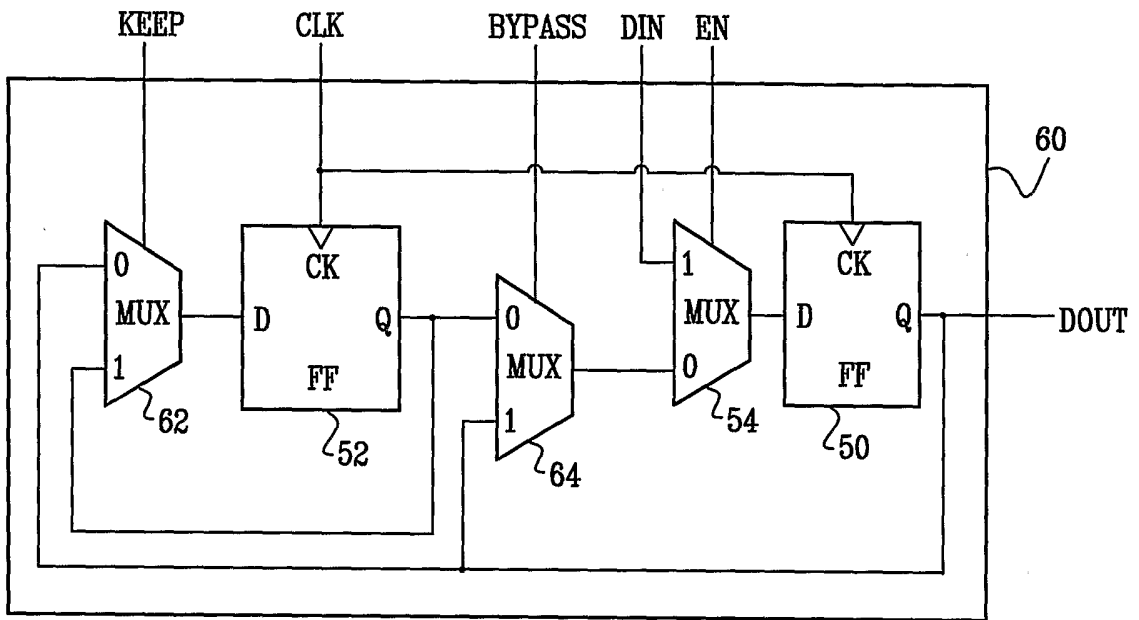


FIG. 5

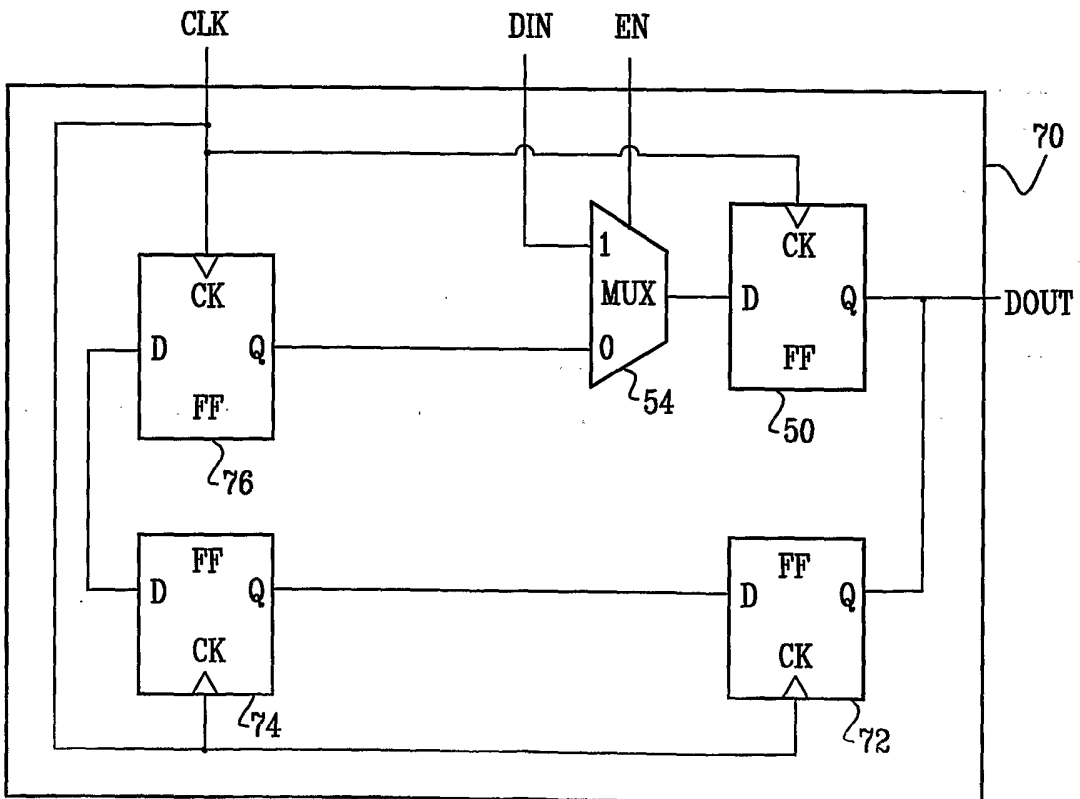


FIG. 6

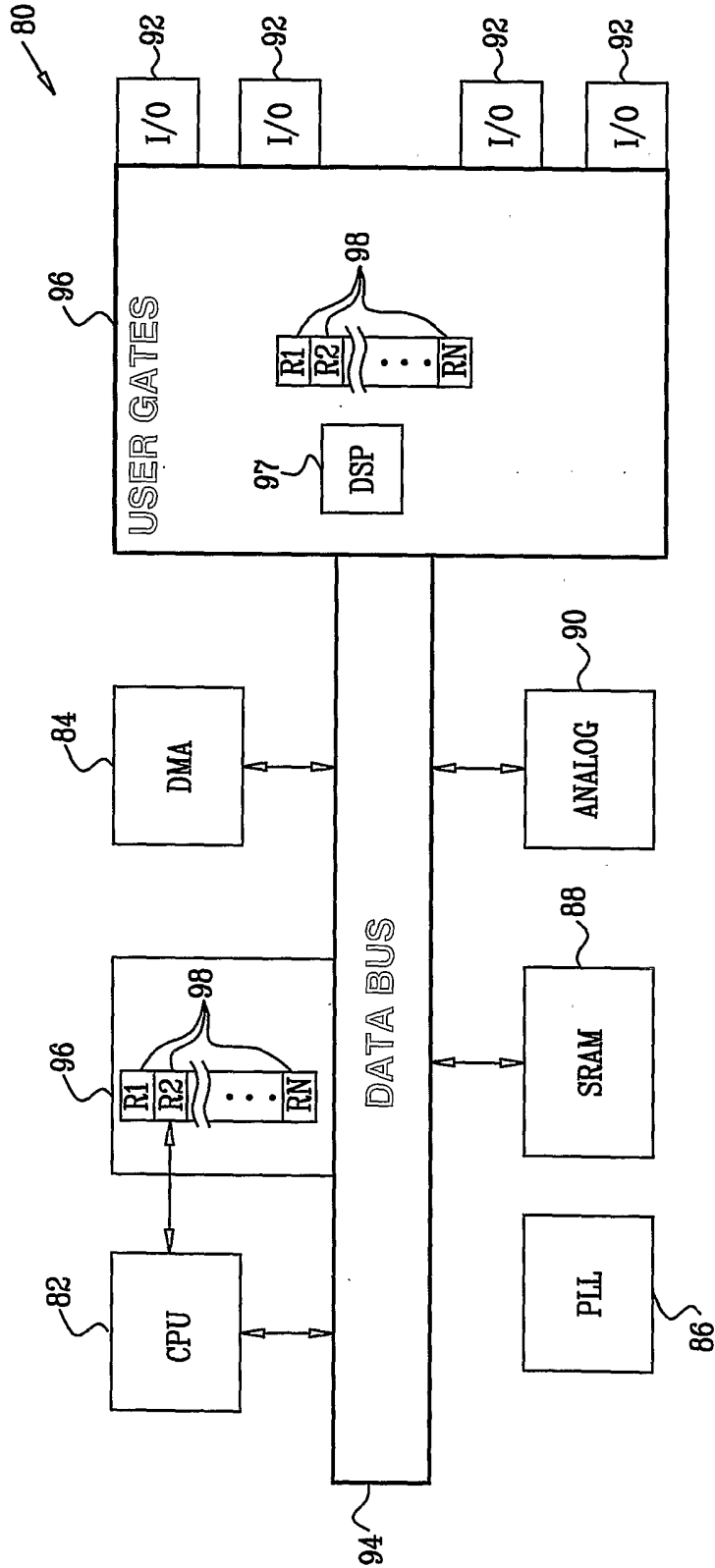


FIG. 7

