THE INTEL 80860 RISC PROCESSOR

M. Lutfar Rahman and M. Alamgir Hossain

INTRODUCTION

The 80860 microprocessor manufactured by Intel Corporation of USA is the world's first 64-bit single chip microprocessor. It is also Intel's first reduced instruction set computer (RISC) microprocessor. This chip is likely to provide supercomputing power to the desktop personal computers.

The 80860 microprocessor (popularly known as i860) is designed for numerical and vector intensive applications. Many of the design principles used have been adopted for supercomputer technology enabling the i860 to deliver a peak arithmetic performance of 80 MFLOPS (million floating point operations per second) for single precision data and 60 MFLOPS for double precision data in conjunction with a peak integer performance of 40 MIPS (million instructions per second). In particular, its high throughput is achieved from a combination of RISC design technique, pipelined processing units, wide data paths and large on-chip caches.

Implemented on a single chip with over 1,000,000 transistors, the i860 supports a 64-bit architecture and is capable of executing upto three operations each clock cycle (25 ns @ 40 MHz).

MAIN FEATURES

On a single chip the processor supports the following facilities :

- inteer operations
- floating point operations
- graphics operations
- memory management support
- data cache and instruction cache
- high speed multiprocessing
- three-dimensional workstation support
- clock speed : 33 MHz, 40 MHz and 50 MHz
- as a co-processor to 80x86 processors
- 168-pin ceramic package
- CHMOS-IV semiconductor technology

Several i860 processors may be made to work in parallel to realise a minisupercomputer. The i860 can be employed to realise a high power graphics workstation and to bring mainframe power to the personal computer.

The ability to provide all these facilities, all on the same chip, enables hardware developers to create products which are less dependent on external components normally associated with sophisticated computer systems. Considering these points of views the microprocessor has some similarities with the transputers. The i860 is an ideal candidate for integration into highly parallel computer environments, such as, high computational performance, modularity, and real-state requirements.

ARCHITECTURE

The chip is a multiexecution system integrating several units on a single chip (Fig. 1). The main functional units are : the RISC integer processor unit, a 64-bit floaing point unit and a three-dimensional graphics processor unit. The other major units are : a paging unit, a data cache, an instruction cache, a bus and cache control unit, a 80x68 compatible memory management unit and three register files.

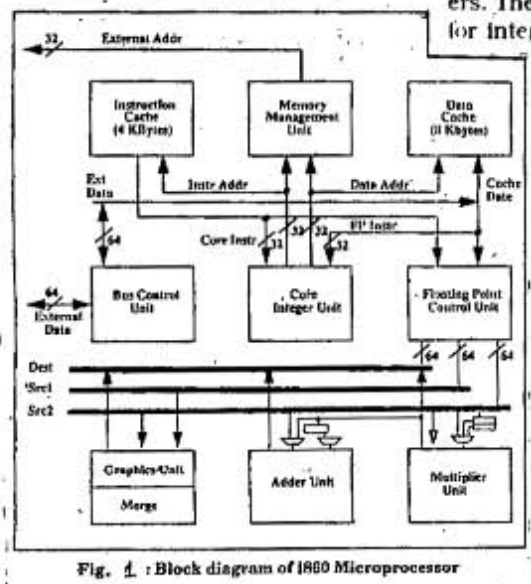

Fig. 1 : Block diagram of i860 Microprocessor

CORE EXECUTION UNIT

The i860 is centrally controlled by the (integer) core unit which is known as the administrator of the processor. It is responsible for fetching both integer and floating point instructions and decoding and executing integer, logical, control-transfer, load/store, exception handling and cache flushing instructions. Instructions are fetched into the core execution unit from the instruction cache. If any address location is not in the cache (a cache miss), the instruction is fed to core execution unit from external memory, while the corresponding instruction cache is simultaneously filled.

The core unit uses a pipeline organisation. The four-stage pipeline operations are shown in Fig.2. When one instruction is fetched, the previous instruction is decoded, the one before that is executed and the results for its predecisor is stored. This processor has been designed according to RISC principles to maximise performance; instructions are purposefully simple and appear to operate in one clock cycle. Furthermore, the use of register bypassing and score-boarding techniques allow the load and store instructions to be executed at a sustained rate of one instruction every clock cycle, assuming that data and instructions are found in their respective caches. At this rate the integer core unit delivers 40 MIPS of integer type with a 40 MHz clock.

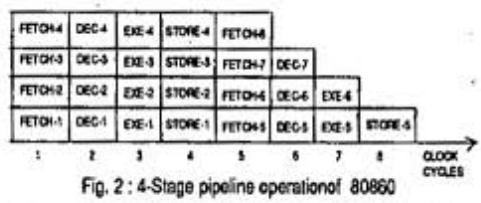

Fig. 2 : 4-Stage pipeline operationof 80860

FLOATING POINT UNIT

The floating point unit contains a control unit, an adder, and a multiplier. Operations can be executed in scalar or pipeline mode in the adder, and in pipeline mode inthe multiplier. In scalar mode, new operations are not started until

the previous ones are completed. In pipelined mode upto three instructions can be overlapped and executed concurrently at any time in the adder and two in the multiplier.

With the support of the instruction and data caches, the floating point unit is capable of executing two single precision floating point operations, one add and one multiply, every clock cycle; this is equivalent to 80 MFLOPS with a 40 MHz clock. An efficient implementation of multiply-accumulate operations makes the i860 well suited for a wide range of numerically intensive aplication areas including :

- matrix manipulations (e.g. solving linear equations)
- series calculations (e.g. expansion series)
- signal processing calculations (e.g. fast Fourier transformation)
- graphics (e.g. coordinate transformations)

Floating-point data types, floating point instructions, and exception handling support the IEE sandared for binary floating point arithmetic for both single and double precision data types. A complete set of traps includes tests for invalid source operands such as NaN (not a number), denormalised numbers, infinities as well as tests for errors in the result such as overflow and underflow.

OTHER UNITS

The graphics unit includes a special 64-bit integer logic module which supports three dimensional graphics algorithms and a special purpose MERGE register. Like 80386, the i860 can support 64 Terabytes virtual memory. The memory management unit is used to translate the logical address to physical address as and when required to access data and instructions in the memory.

The i860 supports a 64-bit (8 bytes) external data bus, a 128-bit internal data bus (two 64-bit paths between data cache and floating point controller) and a 64-bit internal instruction bus. Memory accesses for instructions and data take place through the caches. Each of the data cache and instruction chaches is a associative memory of 4 KB with 32-byte blocks. A cache controller uses pipelined structure to provide interface to the external world.

REGISTER SET

Programmes are developed using the user accesible registers (Fig. 3) of the processor. The i860 has the following user accessible registers.

- An integer register file
- A floating point register file
- Six control registers : psr (processor status register), epsr (extended psr), db (data breakpoint register), dirbase (directory base register) and fsr (floating point status register).
- Four special purpose registers : KR, KI, T and MEERGE.

The integer register file contains 32-bit wide 32 integer registers : r0-r32. The floating point register file consists of 32-bit wide 32 floaing point registers : f0-f31. The registers r0, f0 and f1 always return zero on read. The floating point registers can also be used for integer operations. The floatingpoint register file can be accessed as sixteen 64-bit registers or eight 128-bit registers. These registers support either 32-bit single precision or 64-bit double precision floating point operations.

Like 80386, the i860 supports standard data types which are signed and unsigned 32-bit integers and 32-bit floating points. Besides, the i860 supports a new data type known as the pixel which can be 8, 16 or 32 bits long.

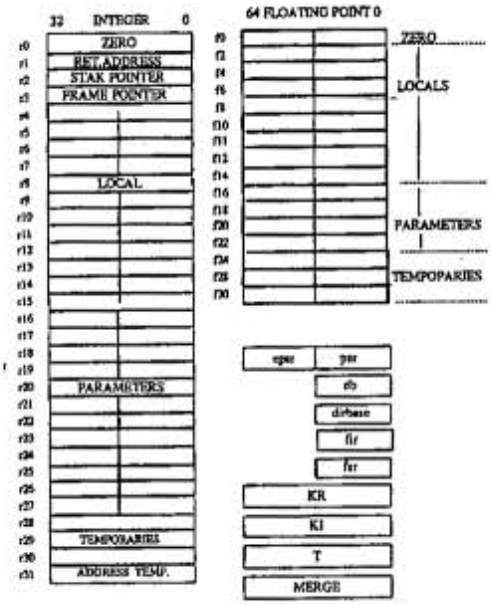

fig.3: Register set of 80860

The i860 uses different instruction sets for different types of operations, such as, the core unit instruction set, the floating point instruction set, the graphics instruction set, and the assembler pseudo

Table 1 : Comparison of 80860 with 80x86 processors :

Topics	80860	80x60
Introduced in	1989-90	1986-87 with 80386, 80486 and 80586 later
Pin count	168	132 for 80386
External data bus	64 bits	32 bits
Archtecture	RISC	CISC
Transistors	1,000,000	275,000 in 80386
Technology	CHMOS-IV	CHMOS-III
Graphics support	Yes	No
Clock (MHz)	33,40,50	12, 16 for 80386 25 for 80486 (higher for later versions
Operations	1 to 3 operations per cycle	On average 4.5 clock cycles per instruction for 80386, 80486 has twice the speed of i386, i586 has twice the speed of i486

operation and floating point pipelined operations. It is to be noted that the 80x86 processors do not use different instruction sets.

COMPARISON WITH OTHER PROCESSORS

Table 1 shows the comparison of i860 processor to the other advanced Intel processors. The external address bus of i860 is 32-bit wide, so like 80386 it can address 4 Gigabyte real addresses and 64 Terabyte virtual addresses. Considering memory organisation and integer and floating point data structure, i860 is compatible with 80x86 family. But i860 is far advanced than 80x86 family if technology speed and some other features are taken into account. At 40 MHz clock speed the i860 can operate at 20 MOPS; this is comparable to the performance of some supercomputers, such as, Cray. Table 2 shows the comparision of 80860 processor with some mainframe computers. Unlike 80x86 processors, the i860 is based on RISC design and parallel mode operation. As a result the i860 is not software compatiple to 80x86 processors.

The performance of i860 is better than some other RISC processor such as SPARC and R3000. The i860 based systems are likely to bring supercomputing power to the desktop computers in the years to come. ◦

Table 2 : Comparison of i860 and other mainframe computers.

	Intel i860	Texas Inst. ASC	Cray Resar. Cray-1	Control Data Star-100
Word (bits)	64 ext. 128 int.	64	64	64
Clock	33, 40, 50 (MHz)	About 6.6 (GHz)	About 0.8 (GHz)	20 (GHz)
Max. primary real memory support	2^{29} words each of 64 bits	130 million 64-bit words	523 million 64-bit words	264 million 64-bit words

INTEL TO BUILD XGA CHIPS FOR IBM

Intel Corp. has joined IBM in trying to establish XGA as the next standard for PC graphics.

Under a technology licensing agreement announced by the two firms last week, Intel will develop XGA chips offering increased graphics preformance, as well as other devices based on IBM's XGA technology.

The first produce will be an XGA chip that Intel will offer to PC manufacturers in early 1993, said Kenneth Fine, general manager of Intel's Multimedia and Supercomputing Components Group, in Phoenix. The chip is expected to offer higher resolutions and more coloer than IBM's own XGA implementation, Fine said.

IBM's XGA offering, available now on PS/2 Models 95 and 90 and on the Model M57 SLC multimedia system, generates 256 simultaneous colors at 1,024 by 768 pixel resolution. Super VGA graphics offer 256 coloers with a resotion of at least 800 by 600 pixels.

Intel is considering XGA designs that could offer capabilities such as 65,000 colors at 1,024 by 768 pixels, Fine said . "XGA can go consiberadly beyond where it is now, he said. "Our goal is to make XGA the standard on all PCs."

PC manufacturers could build Intel's XGA chips into PCs using local bus designs, he added.

Intel is also planning a singlengle chip that combines an XGA controller and its Digital Video Interactive technology, which allows PCs to display video, Fine said.

Analysts said such a chip would enable PCs to be used for high per ormance multimedia applications ."That would be a powerful component for displaying video," said Jon Peddie, publisher of the PC Graphics Report, an industry news letter in Oakland, Calif.

IBM currently manufactures XGA chips for its own use but could use Intel's XGA products in other PS/2s in 1993, said Paul Mugge, IBM's vice Personal Systems in Boca Raton, Fla.

The pact with Intel is the latest in a series of moves IBM has made to push XGA as a standard. In september, IBM licensed its XGA implementation to SGS Thomson Microelectronics Group, which offers XGA chips for use in Industry Standard Architecture PCs and graphics cards. Unlike the license with Intel, the SGS Thomson agreement does not give the compony the right to enhance the graphics capabilities, said IBM officials in White Plains, N.Y.

Neal Boudette

A REVIEW OF A TRANSACTION MODEL FOR

DISTRIBUTED OBJECT MANAGEMENT SYSTEM

Mustafa Rafiqul Islam

This article describes the transaction model of a Distributed Object Management (DOM) system. The DOM is a distributed active object-oriented environment in which autonomous heterogeneous systems can be integrated and new (non-traditional) applications can be developed. A DOM system appears to be homogeneous distributed object system, in which all objects are expressed in a common object model, even though some of the objects actually represent data and functionality of attached heterogeneous systems. Participating systems in the DOM retain a high degree of autonomy. The systems that participate in this federation may not be forced to change their known behavior, and they cannot be forced to give up local control. In order to support such varied and complex applications provided by the DOM the transaction model of such a system must be very powerful.

Previous research on transactions according to a working taxonomy characterizes a transaction mechanism according to its transaction model and its correctness criterion. Transaction model can be further characterized by transaction structure and object structure. The structure of the individual transaction allowed in the model is defined as transaction structure whereas the structure of objects on which the transactions can operate is defined as object structure. The correctness criterion implies the notion of correctness that is used to achieve a certain degree of concurrency transparency in the system.

DOM's transaction model combines closed and open nesting with contingency transactions and executes on complex and active objects. In DOM the closed nested transactions are called toptransactions which makes its result visible to entire system when it commits. Toptransactions can be combined into multitransactions that have some global semantics but permits results to be visible outside the multitransaction. The component transaction of multitransactions may either be vital or non-vital. More specialized subtransactions are Compensating transactions and Contingency transactions. Compensating transactions are defined to undo already committed partial results and Contingency transactions are executed when the primary transaction fails.

The main contribution of this article is to present a transaction model for DOM system. The DOM transaction model that is introduced here is the integration of solutions to individual requirements within a single uniform transaction model. The requirements addressed here are the following. Active capabilities are essential for timely response to events and updates in the environment. This new database model requires monitoring of events and the execution of system-triggered activities within running transactions. DOM also requires the support of long-running activities spanning hours, days or even weeks. Therefore, the transaction mechanism must support the sharing of partial results. For completeness, to avoid the failure of a partial task jeopardizing a long activity, it is necessary to differentiate between those that activities that are required for the completion of a transction and those that are not, and to provide for alternative actions in case the primary activity fails. DOM may not have any control over the heterogeneous and autonomous external systems with which DOM activities requires interaction. This requires the DOM to be flexible to cope with activities whose results may become visible and permanent (i.e. committed) before the DOM transaction that spawns them commits. The transaction mechanism must support the execution of compensating actions to undo the effects of committed subtasks. Since DOM is an object-oriented system, the transaction model must deal with abstract operations. It should improve concurrency by using semantic knowledge about the objects and their abstract operations.

There are significant differences between the transaction models of the current databases and the DOM. There have been many advanced transaction model proposed in current studies from which two dimensions are classified as the transaction structure and object structure. Along with the object structure dimension, simple objects (e.g., files, pages, records), object as instances of abstract data types (ADTs), complex objects, and active objects in increasing complexity are identified. Many of the current processing systems operates on simple objects, mostly on physical pages. The improtant feature of this class is that the operations on simple objects do not take into account the semantics of the objects. For example, an update of a page is considered a write on the page, without considering what logical objects is stored on the page. Whereas the DOM transaction model operates on the active objects, which are capable of responding to events by triggering the execution of actions. Since events may be detected while executing a transaction on that object, the execution of the corresponding rule may be spawned as a nested transaction. Because of the capabilities of additional rules firing within a rule execution, nesting of arbitrary depth are also possible in the DOM transaction model.

Along the transaction structure dimension, flat transactions, closed transactions, open nested transactions such as sagas and combination of these forms, in increasing complexity are introduced. Most of the transaction management model in databases that operate on simple objects has concentrated on flat transactions. There are also some transaction model which operates on closed nesting (i.e. Hipace) and open nesting (i.e. Sagas). The DOM's transac-

tion model can behave as a conventional flat transaction model, it can allows for closed nesting and the execution of triggered processes, or it can be utilized in its most powerful and flexible form by incorporating both open and closed nestings.

The transaction model of the DOM is presented under a few assumptions. Transactions within a DOM multitransaction are assumed to be executable in parallel, unless specified through a precedence constraint. Also while defining the precedence constraints the commit precedence is assumed to be the default mode with the begin precedence specifiable through rules.

The main advantages of the DOM transaction model are as following.

It is a complete, distributed, object-oriented transaction model that combines in a single model of capability of handling open nesting, closed nesting, both explicitly and as a result of handling active objects, and Contingency transactions. It is tailorable and can provide as much flexibility as it is required by the applications. It can use the Local Application Interface (LAI) objects as concurrency control placeholders for external repositories.

The disavantages of this transaction model are as following. As the model is very powerful and flexible implementation of such a system would be very complex. It would be difficult to incorporate ultimate measure of performance. Even though the model is formulated in terms of ACTA formalization, a correctness theory should also be developed. The transaction model does not deal with many of the correctness criterion. Temporal dependencies among transactions are not captured by the DOM transaction model. In order to satisfy such dependencies, new correctness criteria and mechanisms for enforcing them have to be developed. Also whether the framework of the taxonomy is completed and minimal is not determined. The notion of consistency will also have to be revised to consider such issues as the locality of consistency (i.e. for which (sub)systems consistency must be enforced). the level of consistency supported by the (sub)systems and the timeliness of enforcement. These issues needed to be addressed in the DOM transaction model.

This article would be of major importance for initial studies for transaction model of a Distributed Object Management system. The ideas presented here will be very useful to develop a integrated environment that can promote the interoperation of a large variety of software systems in many application domain. The concept of this transaction model can also be useful for designing heterogeneous and multidatabase system. But as mentioned earlier, there are many issues of the DOM transaction model that requires further investigation. For the full heterogeneous system, new notions of consistency must be defined. Relaxations along the lines of locality or timeliness of consistency enforcement must be fromalized.

Mustafa Rafiqul Islam B.S.,M.S.(USA) is Software Consultant of Flora Ltd., Dhaka. Tel. : 246950, 230491, 231751, 241107, 861416.

ISDN LEADING THE NEW WAVE OF COMPUTING

ISDN is a technology that has the potential to revolutionise telecommunications the way the chip revolutionised computing. But what exactly is ISDN and what can users get out of it? This article discusses some of the exciting applications.

ISDN, or Integrated Services Digital Networking, has been hailed as the new type of information communications network.

ISDN allows the digital transmission of data, voice and video at high rates of speed over the same line. It is a totally new communications infrastructure that gives its users everything, from telephone calling to data transmission to image comm unications. ISDN can provide highquality data, voice, image and video services because it has the high-speed capacity to do so.

It uses three bands—two 64-kbps information bands called Base or B bands, and a single 19.2-kbps Data or band. Suffice it to say that ISDN provides communications speeds many times faster than the rate at which most people transmit faxes nowadays. But you might ask: "So what? Why do I need ISDN?"

One obvious benefit is that users of ISDN only need one network to do everything rather than having multiple networks for different purposes.

Second, ISDN allows users to transfer high volumes of data faster than is now possible with conventional phone lines. For example, transferring the contents of a 1.4Mb floppy over a conventional long-distance line can take 20 to 40 minutes. With ISDN, the same data can be transferred in about 1.5 minutes.

Faster and better fax transmissions are possible. The new generation of Group IV faxes will be able to send and receive faxes six times faster, with better print quality, than conventional machines. However, be prepared to pay a premium for these machines—they cost around US$10,000 each.

Accessible

Accessibility is another plus. Today, to connect computers via a phone line, users need modems to convert the digital signals of the computer to the analogue environment of conventional phone lines.

Also, the phone lines must be of sufficient capacity and quality to sustain computer communications. With ISDN, these will not be problems. People can "plug" their computers into any existing ISDN line. Modems will not be needed because a standard for digital transmission will exist.

Finally, high-quality voice communications and new services, such as the ability to display the number of a caller, are possible with ISDN. However, the really exciting benefits of ISDN are the new possibilities it provides in computer services.

ISDN will make it easier for users to combine video, voice and data on one machine, in one application, over remote facilities.

Multimedia applications have many practical uses. For example, engineers designing a component can prepare a written description of that part and addimages to clarify any potential areas of confusion, that is, "what does the logo look like?", and use video facilities to illustrate prototypes in motion.

Videoconference

Another advantage is that ISDN makes computer desktop videoconferencing possible.

I define computer desktop videoconferencing as the ability to use workstations currently used for computer applications to conduct face to face conversations remotely.

As prices for ISDN services decrease, ISDN becomes a cost-effective transport mechanism which will in turn make other services economically viable.

Electronic data interchange or EDI has been widely talked about. Acceptance of EDI has been relatively slow, in part because present transmission technology does not make EDI economically practical.

Sending large paper files electronically today can be very time consuming and costs money. ISDN, with its ability to rapidly send large volumes of data, will make EDI economically viable.

Furthermore, you could use your computer to do what you have to do manually. For example, with X.500, your computer in Hong Kong could ask the computer in New York to find Jones in the New York directory, retrieve the relevant entries and return them to you.

However, this potentially involves looking through and transmitting volumes of data. Since you want this done as rapidly as possible, transmission time must be minimised.

ISDN provides the communications backbone that allows you to send your inquiry and receive the response data quickly and economically. ISDN is becoming a reality and for once, Asia is not lagging but actually leading the world in this technology.

KDD in Japan has not one but several ISDN networks installed and running. ISDN products, such as videoconferencing systems suitable for home use, are now available in Japan.

In Singapore, commercial ISDN service is available. In Korea, the Korean Telecommunication Authority (KTA), will commence ISDN trials early next year. In other areas, France, the UK and Germany are well advanced in ISDN technology.

The US, however, is lagging behind. There are few American ISDN users and sadly, very few American ISDN products although ISDN services are available from AT&T, GTE, Contel, among others.

Every generation or so, a technology comes along that causes a revolution. Not by what it does but by what it makes possible. ISDN is an exciting technology that makes a host of revolutionary computer services possible and the early starters will reap the benefits.

Timothy Regan