(12) UK Patent Application (19) GB (11) 2 195 038 (13) A

(43) Application published 23 Mar 1988

(21) Application No 8616467

(22) Date of filing 5 Jul 1986

(71) Applicant
Narayanaswamy D Jayaram,
36 Dalderse Avenue, Falkirk FK2 7EG, Scotland

(72) Inventor
Narayanaswamy D Jayaram

(74) Agent and/or Address for Service
Narayanaswamy D Jayaram,
36 Dalderse Avenue, Falkirk FK2 7EG, Scotland

(51) INT CL⁴
G06F 15/16 13/40

(52) Domestic classification (Edition J):
G4A MP

(56) Documents cited
GB A 2096369 WO A 84/01043 WO A 81/02210
EP A 0184657 WO A 83/01135 WO A 81/01066

(58) Field of search
G4A

(54) A multi-microprocessor system with confederate processors

(57) A multi-microprocessor architecture with hardware mechanisms and message paths that can effectively map the software mechanisms of synchronization and Interprocessor Communication primitives (shared variable and message-based), has been presented. It is claimed that this design could reduce the memory conflicts arising from busy-wait synchronization, and enables interrupts to be used as signals in the operations of synchronization and Interprocessor Communication primitives.

A confederate processor (COP) is operative for generating interprocessor interrups and sets up interface logic of computer modules for broadcasting messages. Each computer module has locations in all mailbox memories (MM) for messages, task state information and shared variables.

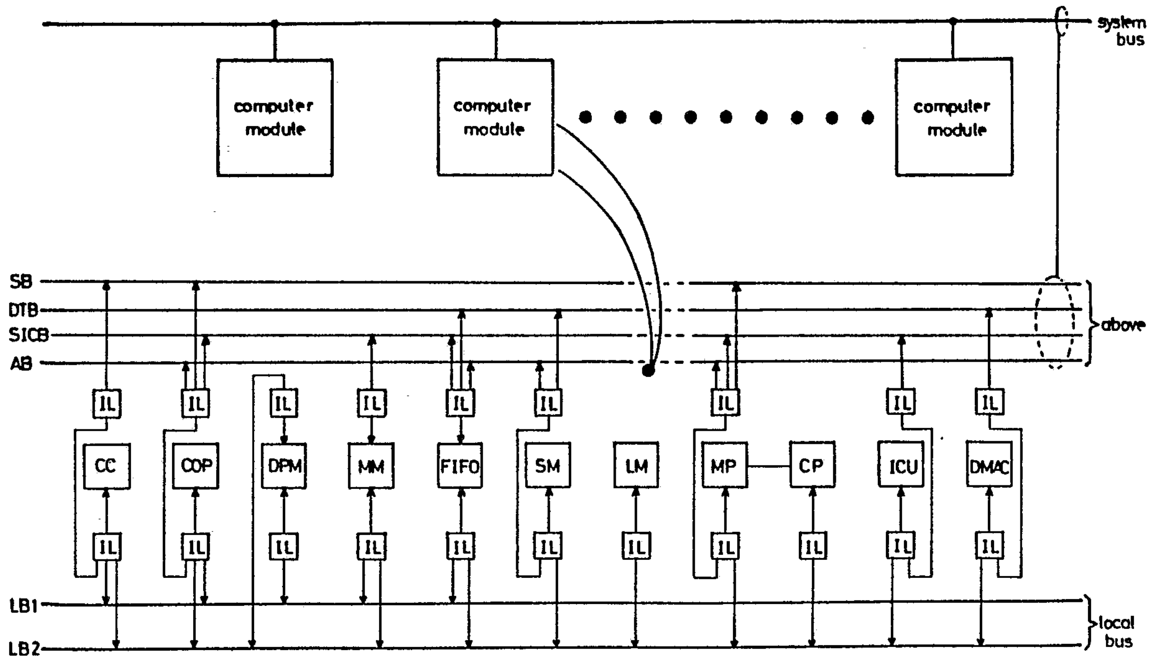


Figure 1

GB 2 195 038 A

-5 JUL 86- 16467

2195038

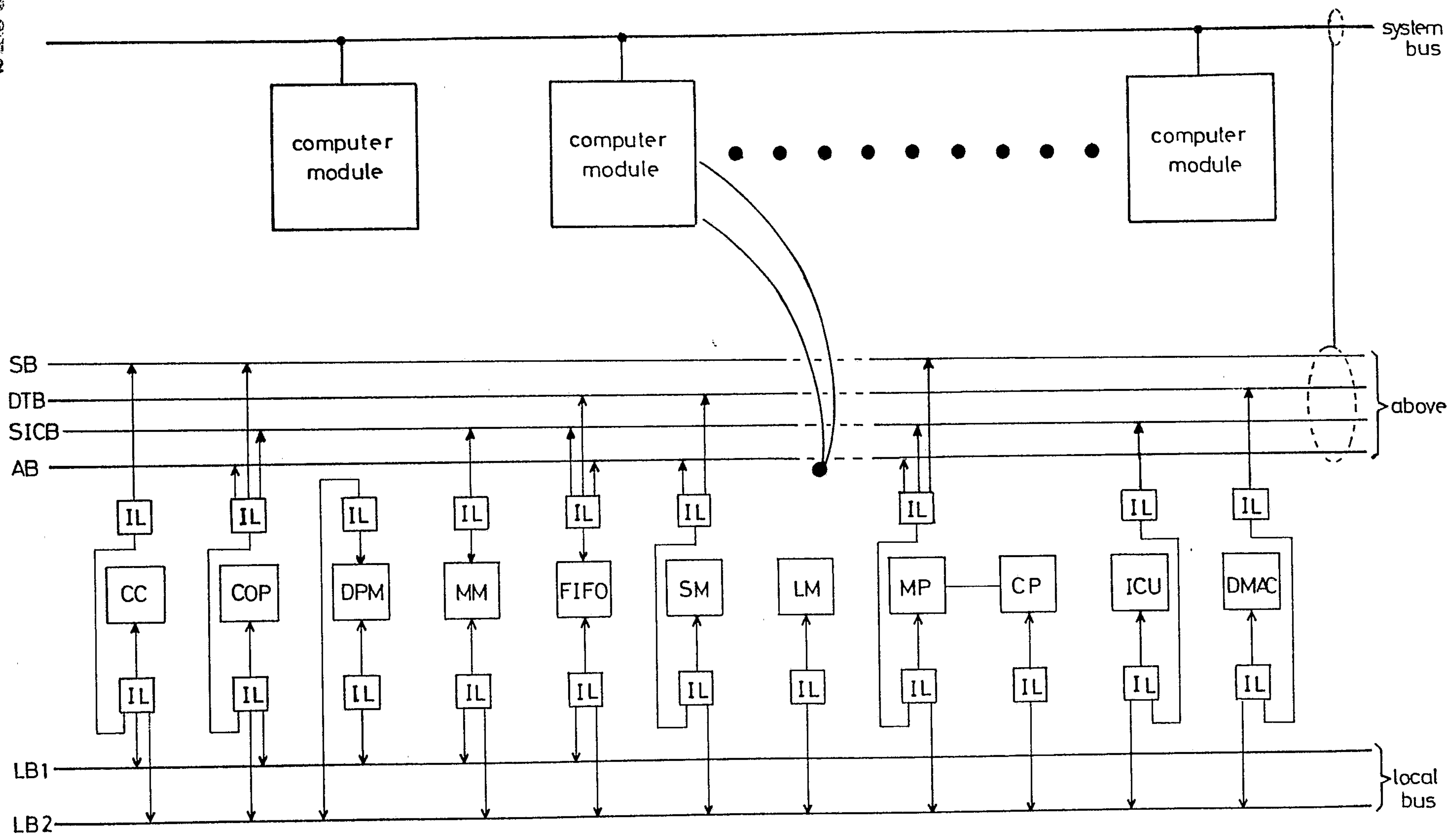
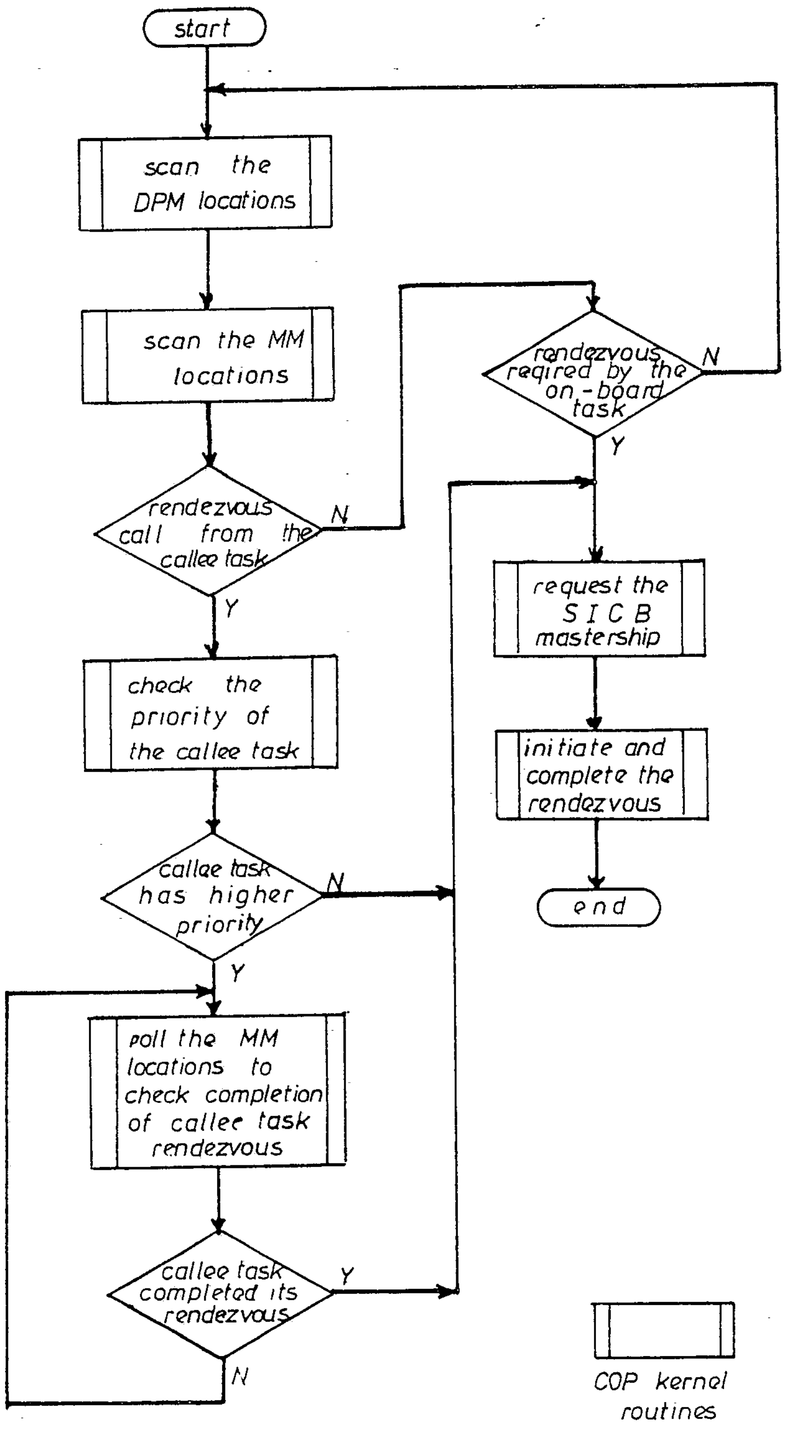


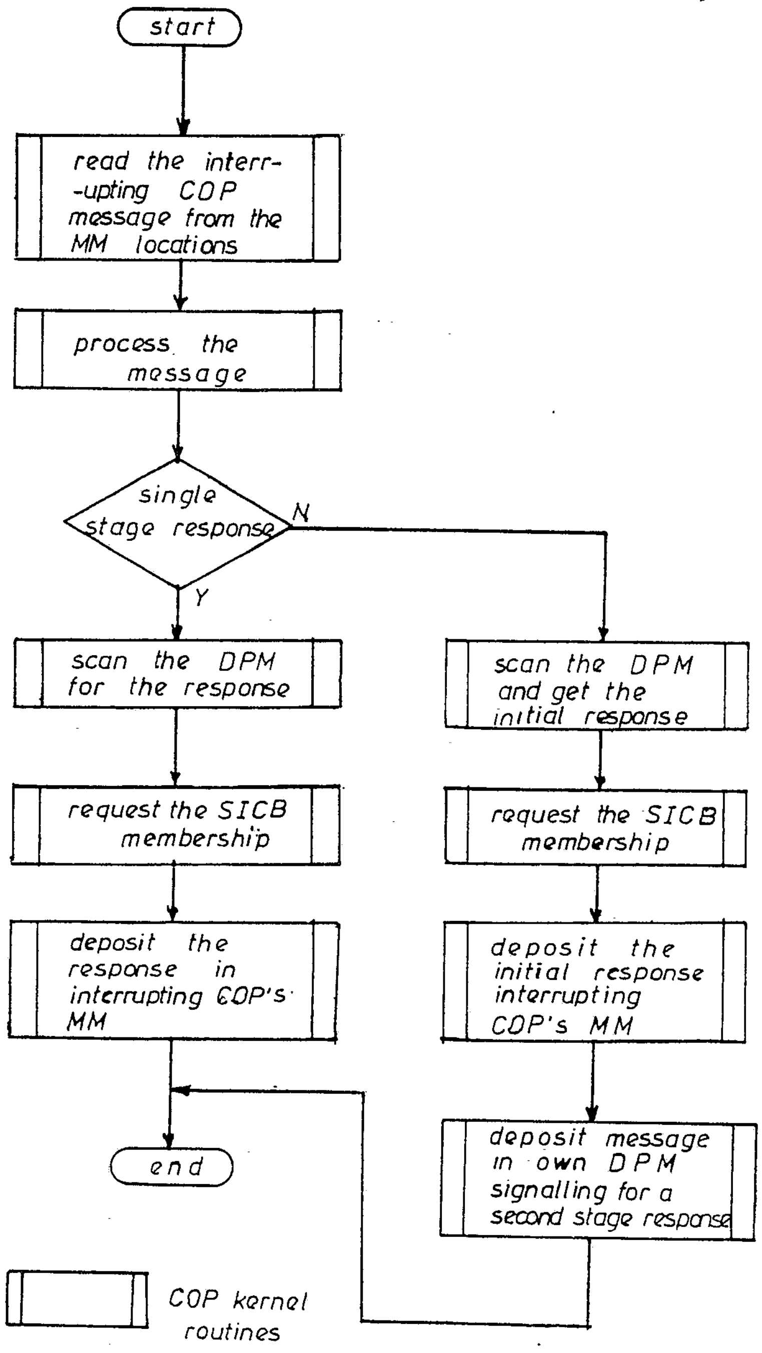
Figure 1

-5 JUL 86- 16467


start
scan the DPM locations
scan the MM locations
rendezvous call from the callee task
N
Y
check the priority of the callee task
callee task has higher priority
N
Y
poll the MM locations to check completion of callee task rendezvous
callee task completed its rendezvous
Y
N
rendezvous reqired by the on-board task
N
Y
request the SICB mastership
initiate and complete the rendezvous
end
COP kernel routines

Flow chart for the COP monitoring of the mutual rendezvous Polling situation

Figure 2


start
read the interr--upting COP message from the MM locations
process the message
single stage response
N
Y
scan the DPM for the response
request the SICB membership
deposit the response in interrupting COP's MM
scan the DPM and get the initial response
request the SICB membership
deposit the initial response interrupting COP's MM
deposit message in own DPM signalling for a second stage response
end
COP kernel routines

Flow chart for the COP assessment of a "signalling" interrupt

Figure 3

SPECIFICATION

A multi-microprocessor system with confederate processors

1 *Background Summary*

1.1 *Introduction*

A multi-microprocessor system compared with a uniprocessor system, offers attractive performance characteristics like enhanced throughput, improved functional reliability, and faster real-time response. However, in the past, the technological limitations of the 8-bit microprocessors presented architectural restrictions on multi-microprocessor system design and led to a somewhat limited use of the multiprocessing technique in real-time applications in industry. The advent of a new generation of high performance microprocessors like M68000/ M68020, iAPX 286/iAPX 386 and NS 16032/NS 32332 and the increasing availability of the state of the art single board computers with these processors have dramatically changed this situation. Today, a multi-microprocessor system can be implemented from off-the-shelf plug-on single board computer modules and backplane buses. But despite the advancement in processor and single board computer technology, some residual problems need to be resolved in the key areas like synchronization and interprocessor communication (SIC) before the potential of the technique is fully realized. Investigations have shown that (MUHL 80) these problems manifest often as undesirable side effects when the SIC primitives operate in concurrent task execution environment. These side effects inhibit parallelism and would prevent the system achieving the linear improvement in performance expected from the system. Performance degradation is inevitable since parallelism in the system architecture and task structure is greatly undermined by the constraints imposed by the operations of SIC primitives.

The author believes that the solution lies in the design of a single board computer with emphasis on hardware mechanisms and message paths that can effectively map the semantics of the SIC primitives on to the system at run-time. Thus in the multi-microprocessor architecture design presented here, an attempt is made to achieve synergy between the architecture and the software mechanisms of the SIC primitives to maximise the performance.

1.2 *The synchronization and interprocessor communication mechanisms*

The issues of synchronization and interprocessor communication result from the operation of the SIC primitives in a concurrent processing environment. This environment is jointly created and sustained by the underlying architecture and the software (the user as well as the embedded kinds). For the purpose of considering the issues raised by the operation of the SIC primitives, let us assume that the semantics of the primitives are implemented on a direct shared memory MIMD organisation (AND 75). Since most of the single board computers referred to earlier employ the shared memory approach, discussions centred on this configuration seem appropriate.

In a real-time task-sharing situation, the rate of execution of a task in a single board computer module and the instant at which this task is ready to rendezvous with a cooperating task (in another computer module) cannot be estimated in advance. The SIC operations like the rendezvous mechanism are thus laced with a high degree of non-determinism. Processors hence resort to two kinds of continual polling such as rendezvous polling and information polling to determine the occurrence of events related to SIC operations (GEH 84). To carry out these polling operations, the processors busy wait on the shared memory and the global bus, thus increasing the traffic on the system bus (MUHL 80, GEH 84). This in turn introduces delays in shared data accesses from other processors and the task communication delays that ensue would weaken concurrency and degrade the potential of the system.

In the following discussions no distinction is made between the terms task, processor, or a processor module, for the purposes of SIC operations. The terms sender and receiver refer to a sender task or a source processor and a receiver task or a destination processor according to the context. Synchronization between two tasks in initiated and carried through by the SIC primitives of message passing or shared variable kind. In the blocking form of message passing primitives the sender waits until the receiver is ready to receive the message (and data) or the receiver waits until the sender is ready to send the message and data. A sender or a receiver could hence spend a considerable time in a wait state, doing no useful processing thus compromising parallelism. The rendezvous mechanism of ADA and the input and output commands of CSP (OCCAM) are examples of this kind of primitives. In the case of non-blocking primitives it is essential that the sender-to-the receiver message exchange is completed before the sender changes its state, for the information contained in the message to be valid. But delays in the message acquisition could prevent this to happen, since the messages at the receiver may be a queue or the receiver is not looking for the message. That is, if the message transaction is initiated from the sender at a time when the receiver is busy otherwise, the arrival of a message may not be recognised and considerable delays in processing the messages would result. Interrupts used as signals in such message transactions pre-empt the task to which they are directed, thus changing the state of the task. Further, a processor operating in a critical section of a task is considered non-interruptible and hence all but the highest priority interrupt are masked. Interrupts introduce an additional degree of non-determinism in a concurrent processing environment that is already operating with a high degree of non-deterministic SIC primitives and the potential programming difficulties discourage the use of interrupts as signals in the domain of the SIC primitives.

It is seen from the foregoing that for effective operation of SIC primitives a task should have the information about the state of its communicating partner/s *before* the start of the message transaction. This is feasible only if a mechanism exists in each computer module to monitor the states of the tasks in execution in the system. The responsibility of disseminating task state information can be entrusted to a second processor in each computer module. This second processor, a microcontroller, henceforth referred to as a confederate processor, handles chores like polling and interrupt handling. A second confederate processor can be added to deal exclusively with interrupt handling operations, if interrupts as signals are found to be useful in mapping the semantics of the SIC primitives efficiently on to the multi-microprocessor architecture. A dual-port RAM addressable from the local bus and the system bus sides in each computer module can be configured as a mailbox memory, with specified locations serving as repository for messages, information and signals. The confederate-dual-port RAM combination thus forms a versatile hardware mechanism that can implement effectively all kinds of software mechanisms of SIC primitives.

2 *Description of the Design*

2.1 *The multi-microprocessor architecture*

The proposed multi-microprocessor system shown in Fig. 1 is organised with a number of single board computer modules connected through a conglomerate system bus structure. The Access Bus (AB), the Synchronization and Interprocessor Communication Bus (SICB), the Data Transfer Bus (DTB) and the Serial Bus (SB) form the constituents of this bus structure. Each computer module shown in Fig. 1 is designed with a mainprocessor (MP) such as M 68010 or NS 16032 or Intel iAPX 286, a coprocessor (CP) such as Intel 80287 or M 68881, a Confederate Processor (COP) such as MK 68200, a mailbox memory (MM) such as TMS 9650, a DMA contrller (DMAC) such as HD 68450, a communication controller (CC) like SCN 68652, a FIFO unit (FIFO) like Z8038 and an interrupt control unit (ICU) like NS 16202. Each computer module is assigned a code that is formed from the four most significant bits of its SM address space. Access to a bus is controlled by an arbiter unit that grants bus mastership to the appropriate processor in each module on a round-robin basis. A counter in each arbiter unit continuously scans the codes of all modules and the bus requests are serviced in accordance with the rotating priority. The interface logic units (IL) associated with the COP and MP contain the arbitration logic and the arbitration signal lines form part of the bus structure. A COP, by sending appropriate signals can set up the interface logic (of the computer modules) to broadcast a message, through a single attempt, to more than one computer module. The local memory (LM) is used for storing programs and private data and the shared memory (SM), addressable from the local bus and system bus sides stores the shared data. The local memory accesses from the mainprocessor, which are normally local program and data accesses are carried out through the local bus LB1. The local bus LB2 is used for memory accesses from the confederate processor/s. These memory accesses are primarily busy-wait and interrupt handling operations, and constitute activities like monitoring messages, information and data in the MM and FIFO units. The SICB bus is used exclusively by the confederate processors to carry out the SIC operations like message deposition, task state communication, shared variable tests and interrupt communication. The AB bus is used as a communication path for activities like shared data accesses by the processors MP and COP. These memory accesses are SM-related operations and hence do not *overlap with* SIC operations which are MM-based and FIFO-based operations. Further the mailbox memory facility transforms the traditional system bus activity like busy-waiting into a local bus (LB2) operation. Each computer module in the system has locations designated in all MMs for messages, task state information, shared variables etc associated with the task running in that module. The confederate processor in a computer module need only poll its MM locations through its local bus LB2 to monitor messages and task states mentioned above. An algorithm can be designed and implemented by the COPs to recognise a pre-assigned priority of a module and to allow the task with the higher priority module to initiate its rendezvous first (Fig. 2). A confederate processor busy-waiting on a MM location thus does not interfere with the shared data accesses by the mainprocessors. It is hence possible to handle potential deadlock situations arising from mutual rendezvous polling when two or more tasks set out to initiate the rendezvous as callers (GEH 84). While the AB bus is used for shared data accesses and single byte data transfers, multi-byte data transfers are carried out by DMA controllers using the DTB bus, thus avoiding potential data transfer conflicts. The SB bus can be used for protocol-directed serial communication between computer modules and from the computer module to the external world.

The COPs generate and process interprocessor interrupts. An interrupt arriving via the SIC bus to a computer module is first directed to the COP. The COP determines the class of response required during the interrupt assessment phase (Fig. 3). The response is classified as single stage and multistage and is related to whether the expected information can be extracted from the COP alone or the joint effort of the COP and the MP is required to produce the information. In the second case, the COP sends a preliminary response to be followed by the required response from the MP. The preliminary response keeps the source of the interrupt informed of the state of the interrupt processing. The COP ensures the preservation of the uninterruptible state of the MP and facilitates the use of interrupts as signals in the domain of interprocessor communication. The bus conglomeration and confederate processors minimise operational conflicts and provides a high degree of fault-tolerance and fail-safe mechanism against inter-module communication failures.

3 *Applications*

The multi-microprocessor system presented has applications as an embedded system, in such areas with inherent parallelism as real-time monitoring in process control operations and processing signals in industrial and military environments.

References

AND 75 G A Anderson and E D Jensen, "Computer interconnection structures; taxonomy, characteristic and examples", *ACM Computing Surveys,* Vol 7, No 4, pp 197–214, December 1975

GEH 84 N H Gehani and T A Cargill, "Concurrent programming in the Ada Language: the polling bias", *Software Practice and Experience,* Vol 14, No 5, pp 413–427, May 1984

MÜHL 80 K Mühlemann, "Method for reducing memory and conflicts caused by busy waiting in multiple-processor synchronisation", *IEE Proc,* Vol 127, pt E, No 3, pp 85–87, May 1980

CLAIMS

I claim that in the multi-microprocessor system design presented:

1. the organization of the confederate processor, the two separate local bus and the four independent system bus structures, the mailbox memory and first-in-first out unit will significantly reduce the memory access conflicts arising from polling operations associated with the operation of synchronization and interprocess communication primitives,

2. the organization referred to in 1 provides fault-tolerant interprocessor communication and system operations through processor and bus redundancies in computer modules,

3. the organization referred to in 1 facilitates interrupts to be used as signals in all synchronization and interprocessor communication operations, with the confederate processor functioning as an intelligent interrupt handler in each computer module.

Amendment to the claims have been filed, and have the following effect:—

Claims 1 to 3 above have been deleted or textually amended.

New or textually amended claims have been filed as follows:—

I claim that in the multi-microprocessor system presented:

1. The mainprocessor (MP), the confederate processor (COP) and the dual-port mailbox memory (MM) set up in a processor board, is presented as a novel organization with the mainprocessor looking after the computation-related activities, and the confederate processor a single-chip microcomputer (a microcontroller), attending to the synchronization and interprocessor-related activities via the dual-port mailbox memory in a normal operational environment. The confederate processor can also serve albeit in a limited capacity, as a stand by mainprocessor in a processor board.

2. The quadruple-bus system bus structure has in it two new components, two separate buses, the SIC bus for carrying the synchronization and interprocessor communication-related signals directed towards the dual-port mailbox memories from the confederate processors, and the AB bus for carrying shared resource arbitration-related signals.

3. The dual-bus local bus structure is also considered novel in that it has two separate local buses, LB1 and LB2, to facilitate independent local accesses to devices and memories from the confederate processor and the mainprocessor respectively in a processor board. This helps to implement the operations described in (1) above.

Published 1988 at The Patent Office, State House, 66/71 High Holborn, London WC1R 4TP. Further copies may be obtained from The Patent Office, Sales Branch, St Mary Cray, Orpington, Kent BR5 3RD. Printed by Burgess & Son (Abingdon) Ltd. Con. 1/87.