Honeywell Interoffice Correspondence

Date: 761221

To: Architecture File

From: Lou Krasny LK

Location: LADC

section: 14.5
doc # : 0110A-0
page : 1

Subject: L66B Interrupt System

1.0 INTRODUCTION

The purpose of this document is to provide the CP-6 designer with an overview of the L66B interrupt system architecture. The L66B system is made up of a Level 66 CPU with the Virtual Memory and Security Option, the 4 Megaword System Controller, and the 6000B IOM with the IOM-B Paging Mechanism. In this document I have attempted to present the information in an orderly manner starting with a description of the individual flip-flops and storage registers involved in the interrupt system and then proceeding to a chronological description of the events that take place in the various system components. I have described the interrupt system as it will exist on the hardware that we expect to use for CP-6; the operation of earlier systems deviates significantly from that described here.

2.0 REFERENCES

The information in this document has been obtained from the following HIS documents:

EPS-1 6000 Processor, 43A219601
EPS-1 6000 EIS Processor, 43A232719
EPS-2 6000 EIS Central Processor, 43A240154
EPS-1 4 Megaword System Controller, 58001190
EPS-2 6000B System Controller, 43A177770
EPS-1 6000B IOM Central, 43A239854
EPS-2 IOM-B, 43A177745
EPS-1 IOM Peripheral Subsystem Interface Adapter, 43A177880
EPS-1 Direct Channel, 43A239853
Series 60 (Level 66)/6000 Macro Assembler Program (GMAP), DD08

3.0 DEFINITIONS

CPU - Central Processing Unit - the hardware module that fetches, interprets, and executes the instruction set of the L66B computer system.

SC - System Controller - the hardware module that provides intra-system communications between storage modules (main memory) and active modules (processors and input/output modules).

IOM - Input/Output Multiplexer - the hardware module that is the physical interface between the system and its peripheral devices. This hardware module contains the channels to which the peripherals connect and the channels used to transfer data directly associated with the operation of the system. The IOM is connected to a port on the system controller.

4.0 INTERRUPT CELLS

Each SC has 32 interrupt levels which are called interupt cells in the SC documentation. A cell is not a memory location; it is a flip-flop within the SC. For CP-6 we will be concerned only with the interrupt cells in the first SC even if there is more than one SC configured on a particular system. In the following discussion, all explicit and implicit references to the SC will be to the first SC.

In Sigma terminology, each individual interrupt cell is either in the armed state or the waiting state. An interrupt cell which is set to a "0" is in the armed state which means that it can accept and remember an interrupt request. An interrupt cell which is set to a "1" indicates that that interrupt level has been triggered and is in the waiting state. Interrupt cells in either the armed or waiting state can be individually enabled or disabled depending on the contents of an Interrupt Mask Register to be described later.

Under program control, a CPU operating in privileged master mode can read the state of all 32 interrupt cells by executing a specific variation of the Read System Controller Register (RSCR) instruction.

A CPU operating in privileged master mode can change the state of any of the 32 interrupt cells as follows:

1. By executing a specific variation of the Set System Controller Register (SSCR) instruction. This instruction unconditionally sets or resets all 32 interrupt cells according to the contents of the CPU Accumulator Quotient (AQ) register.
2. By executing a Set Memory Controller Interrupt Cells (SMIC) instruction. This instruction can set selected interrupt cells as specified by the contents of the CPU Accumulator (A) register. Interrupt cells cannot be reset by this instruction.

An IOM connected to a port on the SC can set 8 of the 32 interrupt cells. Each IOM connected to the SC is assigned a different group of 8 interrupt cells.

Table 4.1 shows the IOM related assignment of all 32 interrupt cells for indirect channels. Notice that the first four interrupt cells are not used by IOM indirect channels.

Table 4.1

ASSIGNMENT OF INTERRUPT CELLS FOR IOM INDIRECT CHANNELS

INTERRUPT CELL NO.	IOM NO.	ASSIGNMENT
0	0	Not Used
1	1	Not Used
2	2	Not Used
3	3	Not Used
4	0	Overhead Channel Interrupts
5	1	Overhead Channel Interrupts
6	2	Overhead Channel Interrupts
7	3	Overhead Channel Interrupts
8	0	Terminate Interrupts for Channels 32-63
9	1	Terminate Interrupts for Channels 32-63
10	2	Terminate Interrupts for Channels 32-63
11	3	Terminate Interrupts for Channels 32-63
12	0	Terminate Interrupts for Channels 8-31
13	1	Terminate Interrupts for Channels 8-31
14	2	Terminate Interrupts for Channels 8-31
15	3	Terminate Interrupts for Channels 8-31
16	0	Marker Interrupts for Channels 32-63
17	1	Marker Interrupts for Channels 32-63
18	2	Marker Interrupts for Channels 32-63
19	3	Marker Interrupts for Channels 32-63
20	0	Marker Interrupts for Channels 8-31
21	1	Marker Interrupts for Channels 8-31
22	2	Marker Interrupts for Channels 8-31
23	3	Marker Interrupts for Channels 8-31
24	0	Special Interrupts for Channels 32-63
25	1	Special Interrupts for Channels 32-63
26	2	Special Interrupts for Channels 32-63
27	3	Special Interrupts for Channels 32-63
28	0	Special Interrupts for Channels 8-31
29	1	Special Interrupts for Channels 8-31
30	2	Special Interrupts for Channels 8-31
31	3	Special Interrupts for Channels 8-31

An interrupt service request on a direct channel interface allows the external system to specify a 3-bit interrupt level number. The IOM in response to this direct channel interrupt service request sets the interrupt cell as indicated in Table 4.2.

Table 4.2

ASSIGNMENT OF INTERRUPT CELLS FOR IOM DIRECT CHANNELS

INTERRUPT CELL NO.	IOM NO.	INTERRUPT LEVEL
0	0	0
1	1	0
2	2	0
3	3	0
4	0	1
5	1	1
6	2	1
7	3	1
8	0	2
9	1	2
10	2	2
11	3	2
12	0	3
13	1	3
14	2	3
15	3	3
16	0	4
17	1	4
18	2	4
19	3	4
20	0	5
21	1	5
22	2	5
23	3	5
24	0	6
25	1	6
26	3	6
27	3	6
28	0	7
29	1	7
30	2	7
31	3	7

5.0 INTERRUPT MASK REGISTERS

Each SC contains four 32-bit Interrupt Mask Registers. Each individual bit of an Interrupt Mask Register is associated with one of the 32 interrupt cells. These mask registers perform the function of enabling and disabling individual interrupt cells within the SC. A specific bit of an Interrupt Mask Register set to a "0" indicates that the corresponding interrupt cell is in the disabled state. A specific bit of an Interrupt Mask Register set to a "1" indicates that the corresponding interrupt cell is in the enabled state.

Under control of switches on the SC configuration panel, each of the four Interrupt Mask Registers can be marked as unassigned or can be assigned to one of the eight ports which connect the SC to CPU's and IOM's. An Interrupt Mask Register is never assigned to a port connected to an IOM. An Interrupt Mask Register must be assigned to each port connected to a CPU that is to respond to interrupts. An Interrupt Mask Register cannot be assigned to more than one port at any time and more than one Interrupt Mask Register should not be assigned to the same port.

When one of the Interrupt Mask Registers is assigned to a port, each bit of that 32-bit Interrupt Mask Register acts as an enable control for the corresponding interrupt cell in determining whether an interrupt signal will be transmitted to the CPU connected to that port. When a bit in the Interrupt Mask Register is a "1" and the corresponding interrupt cell is set to a "1", an interrupt signal is sent to the CPU. When a bit in the Interrupt Mask Register is a "0", the corresponding interrupt cell can be set to a "1"; however, no interrupt signal will be sent to the CPU at that time as a result of that interrupt level. If at a later time the Interrupt Mask Register bit is set to a "1" and the corresponding interrupt cell is still set to "1", the interrupt signal will be sent to the CPU.

A CPU operating in privileged master mode can determine which port each of the four Interrupt Mask Registers is assigned to by executing a specific variation of the Read System Controller Register (RSCR) instruction.

A CPU operating in privileged master mode can read the Interrupt Mask Register assigned to it by executing a Read Memory Controller Mask Register (RMCM) instruction. A CPU operating in privileged master mode can set or reset all 32 bits of the Interrupt Mask Register assigned to it by executing a Set Memory Controller Mask Register (SMCM) instruction.

A CPU operating in privileged master mode can read the Interrupt Mask Register assigned to any port by executing a specific variation of the Read System Controller Register (RSCR) instruction. If no Interrupt Mask Register is assigned to the port specified in the RSCR instruction, zeros are placed in those bit positions that would contain the contents of an assigned Interrupt Mask Register. A CPU operating in

privileged master mode can set or reset all 32 bits of the Interrupt Mask Register assigned to any port by executing a specific variation of the Set System Controller Register (SSCR) instruction.

Both of the instructions which allow a CPU to modify the contents of one of the Interrupt Mask Registers cause all 32 bits of that Interrupt Mask Register to be set or reset at the same time. There is no single instruction that allows a CPU to modify some of the bits of an Interrupt Mask Register while leaving the other bits unmodified.

6.0 INTERRUPT MULTIPLEX WORDS

The Interrupt Multiplex Words are a set of 32 contiguous words located in main memory that are used to communicate the channel number of an interrupting channel from the IOM hardware to the CPU software. The contents of an Interrupt Multiplex Word are modified by the IOM hardware prior to setting an interrupt cell. Interrupt Multiplex Words have no effect on the hardware portion of an interrupt operation in either the SC or the CPU. In response to an interrupt, CPU software can interrogate the contents of one or more Interrupt Multiplex Words.

Interrupt Multiplex Words are located in the lower 256K of real memory on a 32 word boundary as specified by 13 switches on the SC configuration panel. The Interrupt Multiplex Word address is always interpreted as a real address. There is a one to one correspondence between each Interrupt Multiplex Word and each interrupt cell; e.g., the 12th Interrupt Multiplex Word is associated with interrupt cell number 12.

Each of the first 32 bits of each Interrupt Multiplex Word is associated with a specific channel within the IOM. The bit number within an Interrupt Multiplex Word that is associated with a channel is determined by the low order 5 bits of the channel number.

7.0 IOM INTERRUPT SERVICE

The following are the steps executed by the IOM hardware in performing an interrupt service:

1. Using a read and clear operation, the IOM accesses the Interrupt Multiplex Word associated with the interrupt cell that the IOM wishes to set. The read and clear operation is executed by the memory system as a single operation, thereby providing a reliable "multiprocessor gating" function.

2. The IOM ORs a "1" into the bit position of the Interrupt Multiplex Word specified by the low order 5 bits of the channel number. No other bits in the Interrupt Multiplex Word are affected.

3. The IOM stores the modified Interrupt Multiplex Word back into memory.

4. The IOM requests that the SC set the appropriate interrupt cell to a "1".

8.0 INTERRUPT SIGNAL TO CPU

There is an interrupt signal line on each port of the SC. This interrupt signal will always be in the false state for every port which has no Interrupt Mask Register assigned to it. For a port which has an Interrupt Mask Register assigned to it, the interrupt signal will be in the true state whenever there is at least one interrupt level for which a bit in the Interrupt Mask Register assigned to that port and the corresponding interrupt cell are both set to the "1" state.

9.0 CPU RESPONSE TO INTERRUPT SIGNAL

When an interrupt signal is received by the CPU, the CPU will carry out the interrupt procedure as soon as an instruction from an odd memory location has been executed that:

1. Did not have its interrupt inhibit (bit 28) set to a 1.

2. Did not cause an actual transfer of control. (A transfer of control is effected if the instruction is an unconditional transfer, a conditional transfer with the condition satisfied, or an instruction generated fault.

3. Was not an Execute or Execute Double (XEC or XED) instruction. (An XEC or XED instruction and the one or two instructions carried out under instruction control are regarded as a single instruction execution.)

4. Was not accessed while in the repeat mode or was not the instruction following the termination of the repeat mode.

5. Was not a CLIMB instruction.

10.0 CPU INTERRUPT PROCEDURE

When the above conditions are satisfied, the CPU executes the following interrupt procedure:

1. The CPU enters the master mode. (The master mode indicator is not affected.)

2. The CPU issues a Read Program Interrupt Cells command to the SC.

3. In response to the Read Program Interrupt Cells command, the SC returns to the CPU a 5-bit code containing the cell number of the lowest numbered interrupt cell that is currently set and whose corresponding bit in the Interrupt Mask Register is also currently set. The SC then resets the interrupt cell whose number was returned to the CPU.

4. The CPU executes a "wired in" CLIMB instruction which obtains an Entry Descriptor from real memory locations 30 and 31 (octal). The hardware makes the following assumptions concerning the second word of this "wired in" CLIMB:

   Bit 0 = 0 - No parameters are passed.

   Bit 18 = 0 - Do not change X0.

   Bit 19 is ignored and the Master Mode bit of the Indicator Register is set to a "1".

   Bits 22-23=00 - CALL type of CLIMB: context is saved in Safe Store Stack.

   Bits 24-35 are ignored and an Entry Descriptor is obtained from real memory locations 30 and 31 (octal).

   The 5-bit interrupt cell number is saved in bits 13-17 of word 5 of the safe store frame and bit 12 of the same word is set to a "1". The CLIMB instruction is not interruptable.

There is an alternate action for step 3 which can occur in a multi-CPU case or in the single CPU case where the CPU resets its Interrupt Mask Register after the interrupt signal is received by the CPU but before the CPU has initiated the interrupt procedure. If in responding to the Read Program Interrupt Cells command from a port, the SC finds no enabled waiting interrupts for that port, a unique response is returned which causes the CPU to return to the interrupted program without executing a CLIMB instruction and without involving the software in any way.

It should be noted that this is not a true priority interrupt system since the hardware retains no record of the interrupt level currently being processed by the software. As part of the process of causing an interrupt, the interrupt cell is returned to the "0" (armed) state. If the software which is processing an interrupt takes no special action, it can itself be interrupted by any interrupt cell (including the interrupt cell that it is currently processing). The software can simulate a priority interrupt system by resetting some or all of the bits in its Interrupt Mask Register early in the interrupt processing procedure.

In processing an interrupt from an IOM, the following steps may be executed by the system software to determine which channel caused the interrupt:

1. The Interrupt Multiplex Word associated with the interrupt level received can be accessed by the CPU using either a Load A and Clear (LDAC) instruction or a Load Q and Clear (LDQC) instruction. These are the only two CPU instructions that perform a single read and clear operation at memory.

2. If the Interrupt Multiplex Word is non-zero, each one bit identifies a channel on the IOM that requested an interrupt service.

3. If the Interrupt Multiplex Word is zero, one of three possible conditions has occurred: 1) The IOM is in the middle of setting another bit in the same Interrupt Multiplex Word; 2) In responding to an earlier interrupt, the CPU read the Interrupt Multiplex Word after the IOM completed the write back to memory but before the CPU was interrupted for the second interrupt; 3) In a multi-CPU environment where more than one CPU can receive interrupts, if more than one channel associated with the same interrupt cell is requesting an interrupt, an all zero Interrupt Multiplex Word can indicate that another CPU has already accessed that Interrupt Multiplex Word. The sorting out of these conditions in the software is left as an exercise for the reader.