Guide for Adapting

Application Note

Intel® DX4 Design for

Revision Summary: This is the Initial release of this Application Note.

IBM 5x86C Microprocessors

Author D. K. Becker

Introduction

The IBM 5x86C microprocessor is a highly efficient, 32-bit scalar implementation microprocessor. The IBM 5x86C microprocessor embodies a collection of fifth generation features that deliver system performance equivalent to Intel** 75 MHz Pentium** microprocessor systems. These features include a 16 Kbyte write-back unified code and data cache, branch prediction, super pipelining, reduced instruction cycle counts, 2X and 3X clocking modes, and a Cyrix** designed Floating Point Unit. These and other features allow the IBM 5x86C microprocessor to deliver system performance equivalent to that of entry-level Pentium 75 MHz microprocessors in a cost effective 486 platform.

The IBM 5x86C microprocessor also sports power saving features that reduce overall power requirements by shutting down idle internal subsystems. The cache, Translation Lookaside Buffer, Floating Point Unit, and pipeline stages are not clocked when they are idle or stalled. This reduces the average operating power in typical system applications which, in turn, eases the burden on the system power supply, be it battery or plug-in, and on the cooling and packaging of the whole system. These power saving features are automatic, requiring no system board or software changes, and make for a more energy efficient system design. The IBM 5x86C microprocessor also provides a System Management Mode for managing both the processor and total system power consumption.

This guide to adaptation will outline the actions needed to successfully use the IBM 5x86C processor in place of the Intel DX4 microprocessors.

Hardware Differences

5V tolerant I/O

The IBM 5x86C microprocessor's I/O is 5 volt logic tolerant, enabling it to interface directly to 5 volt support chips. The Intel DX4 processor has a V_{CC5} pin, dedicated to programming the inputs and outputs.[1] On the Intel DX4 processor, the V_{CC5} pin is connected to 3.3 volts to program the I/O to 3.3 volt logic levels, and to 5 volts to program the I/O to 5 volt levels. The IBM 5x86C microprocessor does not need to be programmed, and thus has no I/O programming pin.

The two chips (in PGA form) also share a VOLTAGE DETECT pin which is driven low by the processor. This low level can be used by the system board to enable 3.3 volts to the CPU instead of 5 volts. This allows an automatic voltage selection to the processor if needed for an upgrade socket. The QFP versions do not have the VOLTAGE DETECT pin.

Pinout Differences (PGA package)

Pin	IBM 5x86C microprocessor	Intel DX4	Notes
A10	INV	INC	All INC pins on DX4 are internal no-connects and can be wired to.
A12	HITM#	INC	
A13	SUSPA#	INC	
B10	SMI#	SMI#	On 5x86C microprocessor, SMI# is bi-directional
B12	CACHE#	INC	
C10	WM_RST	SRESET	
C12	SMADS#	SMIACT#	
C13	TEST	NC	NC pins on DX4 should not be wired to anything.
G15	SUSP#	STPCLK#	
J1	NC	Vcc5	NC pins on IBM 5x86C microprocessor should not be wired to anything.

Pinout Differences (QFP package)

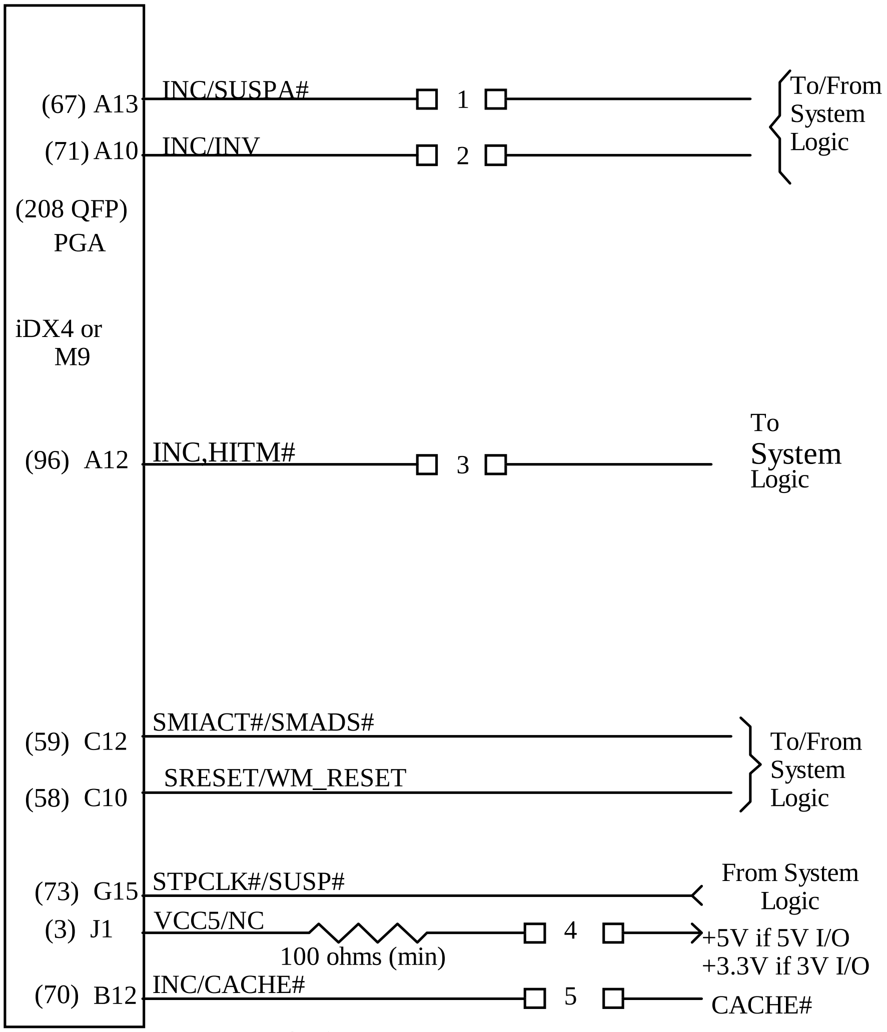
Pin	IBM 5x86C microprocessor	Intel DX4	Notes
3	NC	Vcc5	NC pins on IBM 5x86C microprocessor should not be wired to anything.
58	WM_RST	SRESET	
59	SMADS#	SMIACT#	
63	HITM#	NC	NC pins on DX4 should not be wired to anything.
65	SMI#	SMI#	On IBM 5x86C microprocessor, SMI# is bi-directional
67	SUSPA#	NC	
70	CACHE#	NC	
71	INV	NC	

[1] All data pertaining to the Intel DX4 processor is taken from the Intel document *Microprocessors: Volume II*

73	SUSP#	STPCLK#	
127	TEST	NC	

The Intel DX4 to IBM 5x86C microprocessor adapter schematic illustrates how to design a system to accommodate both an Intel DX4 and an IBM 5x86C microprocessor. The Common

(NOTE: Names on pins are Intel DX4 name / IBM 5x86C name)



Jumper Settings:

CPU	JUMPERS
IBM 5x86C	1, 2, 3, 5
Intel DX4	4

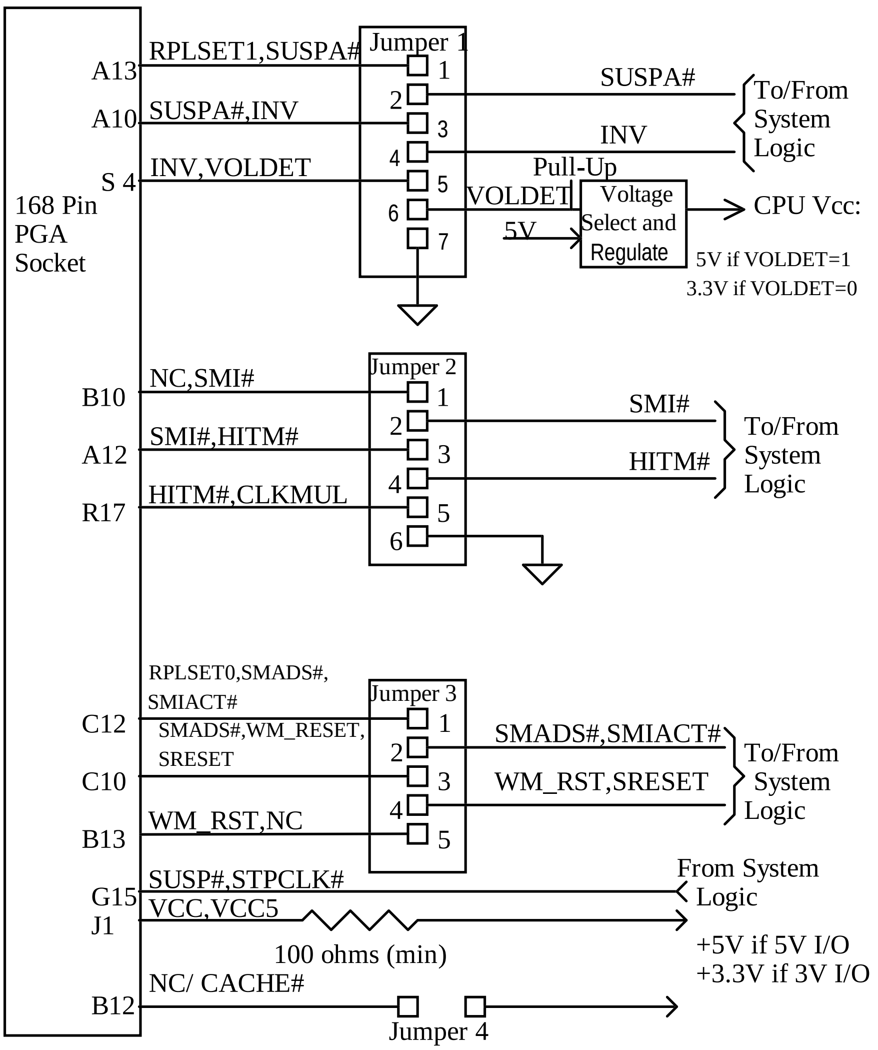
INC = Internal No Connect - These pins are not internally connected on the Intel DX4 and may be wired to. The corresponding pins for QFP DX4 are NC pins and should not be wired to.

NC = No Connect, Pin should not be connected

Jumpers 1,2,3 and 5 are to maintain DX4 compatibility.
Pin numbers on socket indicate pin numbers for QFP and PGA as (QFP) PGA.

Socket specification shows how a design can accommodate several different 486 bus microprocessors.

Intel DX4 to IBM 5x86C Adapter Schematic



Jumper Settings:

CPU	JUMPER 1	JUMPER 2	JUMPER 3
IBM 486DX2	2-3,4-5	2-3,4-5	2-3,4-5
IBM 486DX2-V	2-3,4-5,6-7	2-3,4-5	2-3,4-5
IBM 5x86C	1-2,3-4,5-6	1-2,3-4,5-6 (2X core)	1-2,3-4
I486DX/DX2	NONE	1-2	1-2,3-4
Intel DX4	5-6	5-6 for 2X core	1-2,3-4

Jumper 4 is to maintain DX4 compatibility. Connect it for IBM 5x86C use only.

The following Common Socket Schematic illustrates a method of using jumpers to enable a common PGA socket to service any of the listed processors.

Common Socket Specification Schematic for PGA Socket

JTAG Test Pins

These test pins are common on both the Intel DX4 and the IBM 5x86C microprocessors, and are used in chip test. The test patterns and functions on the pins are unique to the processors and should not be used for any common functions.

BLAST# Signal Differences

The signal BLAST# is handled differently on the IBM 5x86C microprocessor than on the Intel DX4 microprocessor. On the IBM 5x86C microprocessor, BLAST# is asserted on the last DWORD of the burst transfer, not on the last byte of the transfer. This is because the 1+4 method of bursting takes any odd alignment into account on the first burst cycle, instead of the last cycle. On the Intel DX4 microprocessor, BLAST# occurs on the very last transfer of the very last byte.

Considerations: Since any lone bytes or words transferred at the end of a burst sequence are normal read/write cycles, they are not really burst cycles. This allows the BLAST# signal to be used to indicate when the last burst access is really being done. The memory manager hardware chosen should account for this difference.

Clock Multiplier Selection

Prior Intel documentation indicated support of 2X, 2.5X, and 3X core to bus clock ratios. The 2.5X ratio support was dropped in 1995, so the IBM 5x86C microprocessor and Intel DX4 processors both support 2X and 3X clock modes identically. 2X mode is selected by holding CLKMUL low during RESET, and 3X mode is selected by floating or driving high the CLKMUL pin during RESET.

System Management Mode differences

The IBM 5x86C processor supports both the Cyrix and the Intel SL-Enhanced** SMI hardware protocol. The SL-Enhanced protocol is selected by setting the SMM-mode bit in CCR3 to '1.'

In Cyrix SMM mode the SMI# pin is bi-directional. During normal operation it functions as an input signal. At the falling edge of SMI#, the 5x86C processor enters system management mode (SMM). The CPU signals acknowledgement of the SMI# request by holding the SMI# pin low until SMM is exited. While in SMM mode, the SMADS# signal serves as the address strobe rather then the ADS# signal.

On the Intel DX4, the SMI# and SMACT# pins are enabled on the IBM 5x86C by setting USE_SMI in CCR1.

In Intel SL-Enhanced mode the SMI# is still used to signal the processor to enter SMM mode; however, the SMADS# pin now serves as the SMIACT# pin. In this mode a second SMI# can be latched while the CPU is in SMM mode.

On the IBM 5x86C microprocessor, system management mode accesses are not cached, whereas on the Intel DX4 microprocessor, SMM accesses are cached. This does not really present any hardware impact, but there is a beneficial software difference.

Since IBM 5x86C microprocessor doesn't cache SMM accesses, and they are infrequent compared to application accesses, there is no need to flush and refill the cache after an SMM session. On the Intel DX4 microprocessor, the cache must be flushed on SMM entry and exit, even though the SMM mode might be used infrequently. This can lead to reduced performance.

Burst orders

The byte orders used in memory bursts are different between the two processors. The IBM 5x86C microprocessor uses the 1+4 mode of bursting, and the Intel processor uses a toggle burst order. The 1+4 mode of bursting performs any needed alignment accesses first, then performs quad DWORD burst for the remainder of the burst. The toggle burst order implemented by the Intel DX4 is used to keep two way interleaved memory busy, where the IBM 5x86C microprocessor's method keeps DWORD orders in line to optimize the paging mode of memory, and keep the burst times short.

The IBM 5x86C microprocessor can be configured to use the linear burst order through configuration control register 3, bit 2 (LINBRST).

LINEAR BURST ORDER	**1+4 MODE**	**TOGGLE BURST ORDER** (Intel)
0-4-8-C	0-4-8-C	0-4-8-C
4-8-C-0	4 + 8-C-0-4	4-0-C-8
8-C-0-4	8-C-0-4	8-C-0-4
C-0-4-8	C + 8-C-0-4	C-8-4-0
ALL BURSTS ARE 4 DWORDS	**ALL BURSTS ARE 4 DWORDS**	**2 DWORD BURST MAY OCCUR**

Considerations: In order to use the IBM 5x86C microprocessor in 1+4 burst order, the memory controller must be able to support the 1+4 burst sequence.

Suspend Mode and Stop Clock

The IBM 5x86C microprocessor and the Intel 486 DX4 microprocessor both support stopping the clock to reduce idle power to the processor, but the protocol is different.

On the DX4 microprocessor, the signal STPCLK# is used to request that the processor empty its pipelines and write buffers, and then generate a Stop Grant acknowledge bus cycle. Once the DX4 microprocessor has acknowledged STPCLK#, the system can then alter or completely stop the input clock to the processor.

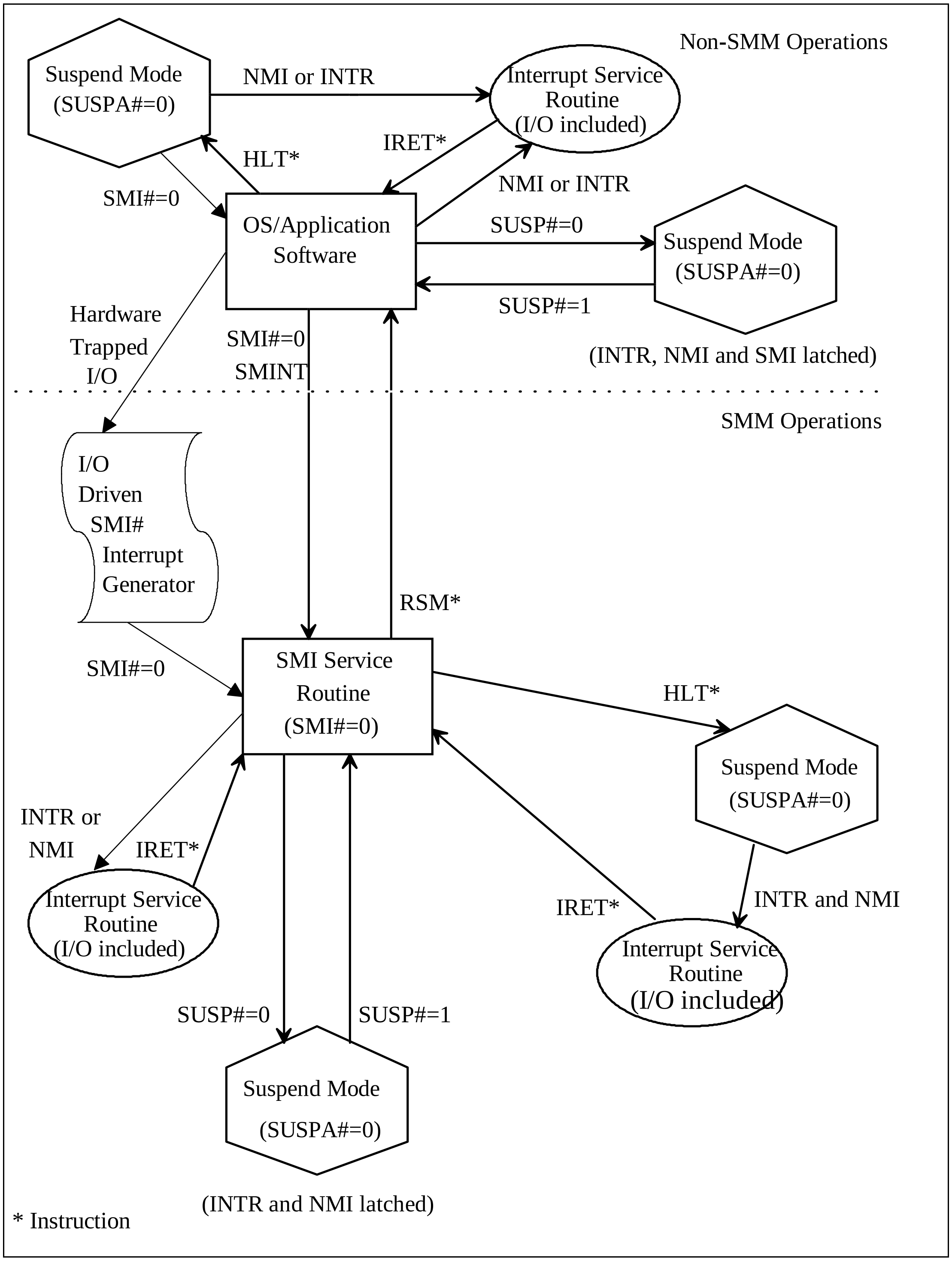
On the IBM 5x86C microprocessor, the analogous control signal is SUSP#. If the USE_SUSP bit in CCR2 is set, and the SUSP# pin is asserted low on the IBM 5x86C microprocessor, the CPU will complete its current instruction, any pending instructions and posted write operations, and then assert the SUSPA# (suspend acknowledge) signal to indicate that the processor has entered a low power state. The clock to the 5x86C microprocessor may be stopped at this time. To resume operation, the clock input must be restarted, followed by the negation of the SUSP# pin. The processor then resumes instruction fetch and execution where it left off. The following state diagram illustrates the state flow associated with going into and out of System Management Mode and Suspend Mode. Also illustrated is the use of an external hardware I/O trap mechanism that allows I/O operations to force an interrupt into System Management Mode. This allows power management for I/O devices to be handled in the SMM service routine.

Misaligned Accesses

Misaligned accesses are handled slightly differently on the IBM 5x86C microprocessor than on the Intel DX4. On the IBM 5x86C microprocessor, low bytes are read first on unlocked reads, whereas on the Intel DX4, high bytes are always accessed first. All other cycles are handled in the same order.

Considerations: If you have memory mapped I/O that clears certain bits on reads, and the byte that does the resetting is in the low byte, then the BIOS should handle the access so that the status is read first with a byte read, then perform the read/clear operation.

SMM and Suspend Mode State Flow Diagram



Software Differences

Write Through and Write Back Cache

The IBM 5x86C microprocessor supports write through and write back cache while the Intel DX4 microprocessor only handles the write through cache. This means that snoops in an IBM 5x86C microprocessor based system will keep the cache updated and will minimize the overhead of flushing and updating the cache after snoops. This improves the performance of the memory subsystem. There is no software impact to Intel DX4 microprocessor software, but IBM 5x86C microprocessor software that is written to use this feature would be potentially incompatible with Intel.

Cache Coherency Management

The IBM 5x86C microprocessor part manages cache coherency (accuracy) using two mechanisms: 1) a valid bit per line (4 DW) and, 2) a dirty bit per Double word. The Intel DX4 microprocessor only uses the 1 valid bit per line to indicate validity. This allows the IBM 5x86C microprocessor part to track cache validity to a finer granularity than the Intel part, and allows more cache hits to occur, with potentially fewer bus cycles needed for cache update.

Considerations: There are no hardware or software changes needed to use this feature in the IBM 5x86C microprocessor. The performance increase will be automatic.

Cache features programmable

The cache operating modes have enhanced control through the configuration control registers. These registers are accessible through I/O space but must be accessed in a specific order and only after enabling their access through the MAPEN control bits. This prevents conflicts between similarly addressed external I/O ports.

Intel DX4 and IBM 5x86C microprocessor Common Control Bits
CR0.CD 1 = CACHE DISABLE
CR0.NW 0 = (WRITE/INVAL) ENABLE
PWT 1 = PERMIT WRITE THROUGH
0 = PERMIT WRITE BACK
PCD = PAGE CACHE DISABLE

EXTRA IBM 5x86C microprocessor cache control bits.
CCR2.4 = WT1 = WRITE THROUGH IN 640K-1MB RANGE ENABLE
CCR2.1 = WBAK = ENBL WRITE BACK PINS CACHE#,INVAL,WM_RST,
CCR2.1 = WBAK = ENBL WRITE BACK PINS CACHE#,INVAL,WM_RST,
CCR3.0 = LOCK CERTAIN SMI REG BITS FROM GENERAL MODIFICATION. SETTABLE ONLY, RST CLR.

Implications: Having additional control on the cache subsystem hardware enhances the system stability and robustness. Program security is also enhanced via the SMI lock bit. BIOS should take these extra control bits into account. Since these control pins affect external hardware pins, the system design should use the pins according to the way BIOS is going to program them.

Branch Prediction

The IBM 5x86C microprocessor has a rudimentary branch prediction scheme that essentially is a weighting algorithm for the probability of a future branch. Since the branch target buffer can store the anticipated branch to address, if a branch prediction is successful, then there is no bus time lost in fetching the new branch-to-code and the branch occurs in one CPU cycle. The algorithm stores the pseudo probability of a branch in a 4 state level sensor. It really does a running 4-way average of "branch taken" conditions. As a branch is taken, and predicted correctly, the weight is increased by 1 to a maximum of 3, where 0 is the low end of branch prediction success and 3 is the highest level of success. If a branch is incorrectly predicted the weight is decreased down to a minimum of 0. Since most real code has loops in it which loop significantly more than twice, the branch prediction algorithm does a fair job of correctly predicting branches.

Software Implications: Faster branch execution on the average with no code changes will improve performance; however, any software timing loops will have their timings significantly changed. The best approach to getting around this is to use internal timers for delaying by a repeatable amount of time.

Register Differences

The internal operational registers of the IBM 5x86C microprocessor are identical to those of the Intel DX4 microprocessor. The IBM 5x86C microprocessor has additional configuration registers that allow the BIOS to set up the controls for some of the new features in the IBM 5x86C microprocessor processor. These registers and their access are described below.

The configuration registers are available through I/O ports 22 hex and 23 hex through an indexing scheme. The index must be written to port 22h, and then the access to the configuration register must follow to/from port 23h. Port 22h reads are always directed off chip.

After reset, configuration registers with indexes C0h to CFh and FEh and FFh are accessible. The remaining registers (index D0h to FDh) are accessible only if the MAPEN(3...0) bits in Configuration control register 3 are set to ones. With the MAPEN bits set to ones, all indexed accesses are held inside the chip and produce no external I/O cycles. If the index number in port 22h is outside the C0h-CFh, FEh, FFh range, or if MAPEN(3..0) are set to 0, external I/O cycles will occur. The following table details the names and accessibility requirements of the configuration registers. The indexes are listed in hexadecimal.

INDEX
(port 22 contents)
	REGISTER NAME	ACRONYM	BIT WIDTH	MAPEN(3..0)
				for access

00 - 1F	Reserved	--	--	--
20	Performance Control	PCR0	--	0001
21 - C0	Reserved	--	--	--
C1	Config 1	CCR1	8	xx
C2	Config 2	CCR2	8	xx
C3	Config 3	CCR3	8	xx
C4 - CC	Reserved	--	--	--
CD - CF	SMM Addr Region	SMAR	24	xx
D0 - E7	Reserved	--	--	--
E8	Config 4	CCR4	8	0001
E9 - EF	Reserved	--	--	--
F0	Power Mgmt	PMR	8	0001
F1 - FD	Reserved	--	--	--
FE	Device ID0	DIR0	8	xx
FF	Device ID1	DIR1	8	xx

For a more detailed description of the configuration control registers, see the IBM 5x86C microprocessor BIOS Writer's Guide, for more information.

Instructions

Some of the instructions in the IBM 5x86C microprocessor have been architected to execute in fewer (in most cases) clock cycles than in the Intel DX4 microprocessor. The only effect of these instructions on software will be to speed up a timing loop, or more positively put, speed up execution of code that uses these instructions (this list may be incomplete).

Assumptions for all: same core frequency, on cache hit, 5x86C microprocessor correct branch prediction.:

Instruction	**Intel DX4 microprocessor cycles**	**IBM 5x86C microprocessor cycles**
ARITH mem,reg	3	1
ARITH reg,mem	2	1
CALL indirect reg	5	3
CALL near	3	3
DIV word	24	29
Jcc near	3,1	1
JMP indirect mem	5	3
LOOPcc	9/6	2
MOVSX	3	1
MUL word	5,6	5
POP mem	5	4
REP CMPS	7+7x	11+4x
REP MOVS	12+3x	12+2x
REP STOS	7+4x	9+2x
RET near	5	3
SETcc mem	3,4	1

Burst Write Control

On the IBM 5x86C microprocessor processor, burst writes can be separately enabled through the BWRT control bit in the configuration registers. This provides extra control over burst writes that the Intel DX4 does not have.

Considerations: The BIOS would be responsible for maintaining the control bit BWRT. In addition, any designs requiring the burst write function should use a chipset that supports the burst write. Otherwise, performance will be impacted.

Programmable I/O Recovery Clock Count

This feature in the IBM 5x86C microprocessor, not available in the Intel DX4 microprocessor, allows software to program the minimum number of clocks to wait between I/O operations. Some I/O operations (particularly those involving mechanical equipment) need time between I/O accesses to allow the mechanics to finish up a prior operation or to prepare for the next. Some software applications have coded-in delays to provide this recovery time. The IBM 5x86C microprocessor has a hardware mechanism to optionally introduce this minimum inter-I/O clock count. This can be used in new software to free up the CPU during any required delays and improve CPU throughput. The delay may be set to 0 clocks to provide compatibility with old software. The default setting is 32 clock cycles.

Considerations: This will not require any hardware changes. Since you can set it to 0 clock delays between I/Os, the BIOS should provide a mechanism for doing so, if complete compatibility is desired.

The information contained in this document is subject to change without notice. The products described in this document are NOT intended for use in implantation or other life support applications where malfunction may result in injury or death to persons. The information contained in this document does not effect or change IBM's product specifications or warranties. Nothing in this document shall operate as an express or implied license or indemnity under the intellectual property rights of IBM or third parties. All the information contained in this document was obtained in specific environments, and is presented as an illustration. The results obtained in other operating environments may vary.

THE INFORMATION CONTAINED IN THIS DOCUMENT IS PROVIDED ON AN "AS IS" BASIS. In no event will IBM be liable for any damages arising directly or indirectly from any use of the information contained in this document.