Upgrading System Designs from i960® Kx to the i960 Jx Processors

Technical Note

October 1998

1.0 Introduction

Intel designed the 80960Jx products to be an easy upgrade choice from the existing 80960KA/KB microprocessor and paid special attention to bus and control signal compatibility during the design process. If your present system or companion chip is based on the 80960KA/KB processor and you anticipate upgrading to the 80960JA/JF/JD/JS/JC/JT processors in the future, you can plan for that transition by observing suggestions in this technical note.

This document describes the hardware and software differences between the two processors that must be considered when upgrading. This document does not list in detail all of the differences between the 80960Jx and the 80960Kx, or the added features. For more detailed information on both processors, please refer to the following documents:

Electrical specifications for these products are found in the following documents:

- *80960 JA/JF/JD/JT Embedded 32-Bit Processor* datasheet
- *80960 JS/JC Embedded 32-Bit Processor* datasheet
- *80960KA Embedded 32-Bit Microprocessor* datasheet
- *80960KB Embedded 32-Bit Microprocessor with Integrated Floating-Point Unit* datasheet

Functional descriptions for these products are found in the following documents:

- *i960® Jx Microprocessor User's Manual*
- *i960® KA/KB Microprocessor Programmers Reference Manual*
- *i960® KB Hardware Designers Reference Manual*

2.0 Hardware Considerations

2.1 Bus Organization

Like the 80960KA/KB processor, the 80960Jx devices have a 32-bit, multiplexed address/data bus capable of high bandwidth burst transfers. Although the 80960Jx products occupy the same 132-lead PQFP and PGA packages and have similar control signals, they are not pin-compatible with the existing 80960KA/KB device.

Enhancements to the bus include de-multiplexed, incrementing address signals A[3:2] and programmable bus width. For 16-bit buses, the 80960Jx processor drives address signal A1 on the BE1# pin. For 8-bit buses, the processor also drives address signal A0 on the BE0# pin. This addressing convention corresponds to the 80960Cx processors.

2.2 Control Signals

The 80960Jx microprocessor uses three-state output buffers instead of open-drain buffers for all control signal outputs. This change could affect a few existing 80960Kx designs that have multiple processors sharing a memory subsystem.

The following list identifies control signals that are **directly** compatible to their 80960KA/KB counterparts:

- During the address cycle, pins AD[1:0] denote SIZE bits to indicate the length of a burst. The functionality of these pins is now extended to 8- and 16- bit bus widths, where the maximum number of transfers is also four.
- ALE# works identically. For new design work, the complementary ALE signal is available also.
- Byte enables BE[3:0]# on the 80960Kx processor are valid at least one clock before the corresponding data state of a burst. Byte enables may change states during a burst access to qualify unaligned data transfers. In other words, the byte enables are pipelined and toggling and external system logic must latch the signals. Microcode in the 80960Jx processors breaks up requests for unaligned data into smaller bus accesses that are always individually aligned. The 80960Jx processors drive appropriate BE[3:0]# pins active during T_a and inactive during T_r. The pins do not toggle within the burst, so there is no need to latch them. Existing system logic designed for pipelined, toggling byte enables automatically handle the new BE[3:0]# signals as a trivial case.
- The W/R#, DT/R# and DEN# pins function identically.

The following list identifies control signals that **may require minor changes** to an existing design based on the 80960KA/KB processor:

- The 80960Jx processor asserts ADS# only once per bus access, during the address cycle. The 80960Kx processor asserts this pin at other times during burst accesses, too, but most systems ignore all other assertions.
- The 80960Jx device extends the use of the READY# pin to control the number of recovery (T_r) states between a current bus transaction and the next bus transaction. If system ready logic is configured as Normally-Not-Ready, this new RDYRCV# pin automatically functions as before; i.e., there will be exactly one recovery state.

The following list identifies **new control signals** that are provided on the 80960Jx processors but are absent on the 80960Kx processors:

- WIDTH[1:0] pins denote the programmed bus width corresponding to each bus access.
- D/C# indicates data and code accesses, respectively.
- The Burst Last (BLAST#) pin, an important addition, signals the end of a bus access. This pin functions like BLAST# on the i960 Sx and i960 Cx microprocessors.

The following list enumerates 80960Kx control signals that are **not supported** by the new processor:

- CACHE
- BADAC#

2.3 Bus Arbitration

The HOLD/HOLDA protocol is enhanced so that the 80960Jx processor cannot pass directly from a T_d bus state to the T_h (HOLD) bus state without passing through the T_r state(s). This change is necessary to support the RDYRCV# pin described previously. Another difference is that 80960Jx processors respond to HOLD requests during reset (80960Kx processors cannot). The timing relationship of the HOLD and HOLDA pins is unchanged.

The 80960Kx processor has an open-drain LOCK#. The processor tests the state of this pin prior to asserting it. The 80960Jx processors have a one-way LOCK# output pin, implemented in three-state logic.

2.4 Interrupt Control Inputs

The 80960Jx processor's interrupt control unit is identical to that of the i960 CA/CF processors. It has eight general purpose XINT# interrupt pins and one NMI# pin. Note that these are active-low inputs, but, unlike the i960 CA/CF device, the pins are sampled on the rising clock edge. For future compatibility, avoid using the 80960KA/KB processor's Interrupt/Interrupt Acknowledge protocol, as this model is not supported in the i960 Jx microprocessor family.

The 80960Jx processors do not support Inter-Agent Communication (IAC messaging), so there is no IAC# input pin.

2.5 Other Hardware Compatibility Notes

The active sense of the reset signal changed on the 80960Jx processors from high to low.

On the 80960Jx processor, the LOCK# pin is tested upon RESET# de-assertion. If it is low, the processor enters the ONCE three-state test mode. The 80960Kx processor does not support ONCE mode.

All versions of the 80960Jx processor are clocked at the bus frequency, even if the core clock is doubled or tripled (80960JD or 80960JT respectfully). The 80960Kx processor is clocked at 2*x* the bus frequency.

3.0 Software Considerations

3.1 IBR-Initialization Boot Record

The location of the Initialization Boot Record is different on the 80960Jx and the 80960Kx. The IBR on the 80960Kx is located at 0x00000000, whereas the IBR for the 80960Jx is at 0xFEFFFF30. On the 80960Jx, the area where the 80960Kx IBR is located is in internal data RAM. However on the 80960Kx the area where the 80960Jx boot code is located is reserved.

The formats of the IBRs for these processors are completely different. 80960Jx initialization is much more closely related to the CX processor.

3.2 Device ID

For the 80960Jx, a number characterizing both the microprocessor type and stepping is placed in register g0 upon reset. In addition, this device ID is also placed in a memory-mapped register on the 80960Jx and can be accessed directly. The 80960Kx does not have a device ID.

3.3 Initialization Code

The initialization code used on the 80960Jx is almost completely compatible with the Cx, but quite different from the 80960Kx. Some of the differences and similarities include the following:

- The PRCB of the 80960Jx does not need to be in RAM as on the 80960Kx; thus no moving of the PRCB and re-initialization is required. The PRCB is not looked at after reset on the 80960Jx. (The initialization is approximately 600 cycles faster without the re-initialization.)
- If the IMI is changed on the 80960Kx, a re-initialization is required afterward. To change the IMI on the 80960Jx, simply write to the corresponding MMR (memory mapped register).
- The 80960Jx must reinitialize after moving the interrupt table.
- Stacks and heaps must be in RAM on both the 80960Kx and the 80960Jx.
- The control table does not need to be in RAM on the 80960Jx.

Sample initialization code is available for both the 80960Jx and the 80960Kx in their respective manuals.

3.4 Register Values after Reset/Re-initialization

Not all registers on the 80960Jx contain the same initial values as their counterparts on the 80960Kx. The user should take care to initialize all registers on each part before using them.

3.5 Procedure Calls

Procedure calls and related table structures are compatible with the following stipulations:

- The FP and SP on the 80960Kx are always 64-byte aligned. The FP and the SP on the 80960Jx are always 16-byte aligned, eliminating unnecessary padding in the stack. Thus the processor always ignores the lower 4 bits of the FP on the 80960 Jx as opposed to the lower 6 bits on the 80960Kx.
- The 80960Jx does not allow writing to the RIP. An OPERATION.INVALID_OPERAND fault occurs if this is attempted. To update the RIP, execute a **flushreg** instruction and change the RIP out in memory. The new RIP will then be read in from memory.
- Reserved Return Status Field patterns result in an OPERATION.UNIMPLEMENTED fault on the 80960Jx.
- The resume bit in the Process Controls register is not implemented on the 80960Jx. This is due to the fact that the 80960Jx has no instruction suspension and resumption mechanism, and

does not use resumption records. (Resumption records are used on the 80960Kx for long floating-point instructions.)

- The internal state bits in the Process Controls register are unimplemented on the 80960Jx. The user should not write to these bits on either the 80960Kx or the 80960Jx.
- The 80960Jx does not have a stopped state.
- Register bypassing is implemented on the 80960Jx, and is not on the 80960Kx. The following is an example of register bypassing:

```
mov r6, fp The FP is written out on this step.

addo fp, 0, 55
```

This instruction does not read in the FP, but uses its value from the previous instruction. The actual FP could differ from the FP used in the addo instruction due to the fact that the processor ignores the lower 4 bits of the actual FP.

3.6 Interrupts

The 80960Jx uses the interrupt controller from the CX, which is different from the 80960Kx. For complete details of these processor's interrupt controllers, refer to their respective user's manuals.

3.7 Memory Control

While the 80960Kx has no built-in memory control, the 80960Jx has 8 programmable memory regions. It contains 8 PMCON registers for programming bus width for the physical memory regions. It also uses 2 types of logical templates each comprised of 2 logical memory control registers (LMMRs and LMADRs) to control data cache enabling for its specific region. The 80960Jx has a default logical memory configuration register (DLMCON) for accesses that do not fall inside one of the two logical memory templates. DLMCON also controls byte ordering for the entire memory map.

3.8 IACs

IAC (inter-agent communication) requests are not implemented on the 80960Jx. The following is a list of all of the 80960Kx IACs and a description of how their functions can be accomplished on the 80960Jx:

- Continue Init IAC is not needed on the 80960Jx.
- The functionality of a Freeze IAC can be accomplished on the 80960Jx by putting the processor into HALT mode.
- Software Interrupts are initiated with **sysctl** instruction on the 80960Jx.
- The functionality of the Purge Instruction Cache IAC is achieved on the 80960Jx using a **sysctl** or the new **icctl** instruction.
- The functionality of the Reinitialize Processor IAC is achieved on the 80960Jx using a **sysctl**.
- The Set Breakpoint IAC function is entirely different on the 80960Jx. The same functionality is accomplished on the 80960Jx by obtaining rights to the breakpoint registers using **sysctl**. Then, the registers can be written to directly as MMRs.

- The Store System Base IAC function is not needed on the 80960Jx because there is no System Address Table on the 80960Jx and the PRCB address is located in an MMR.
- The Test Pending Interrupts IAC function can be performed on the 80960Jx by posting a software interrupt with an invalid vector number in the range of 0-7. This causes pending interrupts to be rescanned.

3.9 Faults

The software handling of faults, including related table structures and procedures, is compatible between the two processors, with the following exceptions and additions:

- There are no floating-point faults on the 80960Jx.
- The 80960Jx never generates CONSTRAINT.PRIVILEGED faults, but generates TYPE.MISMATCH faults instead.
- The Trace Controls Breakpoint/Mark Mode only controls the mark instruction on the 80960Jx; on the 80960Kx, it also controls breakpoints. The Breakpoint/Mark Event Flag signals the event of a mark, fmark, or breakpoint fault on both processors.
- The 80960Jx has Parallel faults, Override faults, and system errors; the 80960Kx does not.
- The 80960Jx does not have a trace fault table.
- The system-call trace fault entries on the 80960Kx require the second word of the entry to be 0x27F. This 0x27F is also specified on the 80960Jx; however it is ignored.
- There is a new fault, OPERATION.UNALIGNED, on the 80960Jx that is not on the 80960Kx.
- There is a new fault resumption record added to the fault record to be used for Parallel and Override faults on the 80960Jx.

3.10 Tracing and Debugging

There are no major tracing and debugging differences between the 80960Jx and the 80960Kx. Several minor difference include:

- The 80960Kx and the 80960Jx have two instruction address breakpoints. In addition, the 80960Jx has two data address breakpoints.
- The breakpoint registers on the 80960Jx are located in MMRs. Application code on the 80960Jx must first request and acquire modification rights to the hardware breakpoint resources (BPCONs) before any attempt is made to modify them. This procedure is not required on the 80960Kx.
- Bits 27:24 of the trace controls describe Hardware Breakpoint Event flags on the 80960Jx. These bits are reserved on the 80960Kx.
- Bits 23:17 of the trace controls describe trace event occurrences on the 80960Kx. These bits are reserved on the 80960Jx.

3.11 Instructions

The 80960Kx and the 80960Jx contain the same basic instruction set with the following exceptions:

- New instructions on the 80960Jx (designated in the *i960® Jx Microprocessor User's Manual*) are not compatible with the 80960Kx.
- The 80960Jx has more supervisor instructions than the 80960Kx, including: **dcctl**, **icctl**, **intctl**, **indis**, **halt**, **sysctl**. The use of these instructions while not in supervisor mode results in a TYPE.MISMATCH fault.
- **daddc** and other decimal and floating-point instructions are not supported on the 80960Jx.
- **synmov** instructions were used for IACs and are not present on the 80960Jx.

3.12 Floating Point

The 80960KB contained a floating-point unit, whereas the 80960Jx does not; therefore, there are no floating-point registers on the 80960Jx. In addition, the floating-point bits in the Arithmetic Controls have no effect on the 80960Jx processor. In other words, no other function has been assigned to them.

3.13 Data RAM, Local Registers, and MMRs

The 80960Jx has 1 K of on-chip data RAM located at 0x00000000. The 80960Kx does not contain on-chip data RAM.

Another memory map conflict includes the reserved memory space on the 80960Kx (0xff000000 - 0xffffffff), which is where the Memory Mapped Registers on the 80960Jx are located.

The 80960Kx has four register sets whereas the 80960Jx has eight register sets. The 80960Jx can also reserve between 0 and 7 sets for high priority interrupts, and uses the Register Cache Configuration Word in the PRCB to program local register availability.

3.14 Caches

The instruction cache is 2 K on the 80960JA, 4 K on the 80960JF/JD, and 16 K on the 80960JS/JC/JT. The instruction cache on the 80960Kx is only 512 bytes. All 80960Jx instruction caches are two-way set associative. The 80960Jx also has the ability to load and lock one way of the instruction cache to minimize latency on program control transfers to key operations such as interrupt service routines.

The data cache is 1 K on the 80960JA, 2 K on the 80960JF/JD, and 4 K the 80960JS/JC/JT. All 80960Jx data caches are direct mapped. The 80960Kx does not have a data cache.

4.0 Other Compatibility Notes

The 80960Jx has no multiprocessing capability as the 80960Kx does.