



Keeping Embedded Drivers Portable between AMCC / IBM PowerPC 4xx Processors

Background

Ideally for embedded software engineers, IBM would have kept things identical to the PowerPC 405 core when introducing the higher performance PowerPC 440 core. However, as design likely necessitated, there are a great number of differences that need to be considered when developing software that is meant to be portable between the two cores. Applied Micro Circuits Corporation (AMCC) has since acquired the 400 series of embedded PowerPC products from IBM and is providing support for them.

This document will specifically talk about the differences when porting from the PPC 405GP to the PPC 440GP, and subsequently to the PPC 440GX. AMCC has released documents relating to the software considerations when porting between these processors, but these tend to over-simplify the work that is required. This document does not discuss all the new features of each processor, but rather the features requiring consideration for the upgrade.



The primary differences between the PPC 405GP and PPC 440GP are:

Subsystem	PowerPC 405GP	PowerPC 440GP
SPR & DCR Addresses	Many different addresses between the processors.	
OPB/PLB	32 bit addressing	36 bit addressing
Exceptions	Fixed exception routine offsets located by EVPR	User defined exception routine offsets set via IVOR0-IVOR15 and located by IVPR.
Interrupts	All interrupts mapped to UIC0.	Interrupts mapped to UIC0 and UIC1.
MMU - General	“Real” mode enabled upon startup allowing direct 1-1 mapping to memory before TLB entries are written.	No real mode. Only has a shadow TLB entry enabled on startup allowing access to addresses 0xFFFFF000-0xFFFFFFFF. Code must be located in this space which sets up valid TLB entries.
MMU - TLB entries	TLBHI and TLBLO used to map between virtual (32 bit) and real (32 bit) addresses.	TLB0, TLB1, and TLB2 used to map between virtual (32 bit) and real (36 bit) addresses.
PCI	PCI controller located as device zero on PCI bus. Some register read/writes require configuration cycles.	PCI registers memory mapped.
Ethernet	One Ethernet controller	Two Ethernet Controllers.



The Subsequent differences between the PPC 440GP and PPC 440GX are:

Subsystem	PowerPC 440GP	PowerPC 440GX
Interrupts	Interrupts mapped to UIC0 and UIC1.	Interrupts mapped to UIC0, UIC1, and UIC2. Also requires initialization of UICB0.
Machine Check Exception	Uses same SRR registers as other exceptions.	New machine check specific SRR registers, as well as new capability to recover based on parity errors.
Ethernet	Two Ethernet Controllers.	Four Ethernet Controllers and some different register bits.



PowerPC 405 GP to PowerPC 440 GP

SPR & DCR Addresses

Perhaps one of the most troublesome points of trying to keep software portable is that many SPR and DCR addresses were changed, removed, and added. In addition, the functionality of bits within some of the SPRs was modified. While IBM suggests that by using the same assembly mnemonics the compiler will pick up the new addresses, this method only works if your compiler has an option specifically for the 405GP and the 440GP. The most portable solution is to use the generic “mtspr”, “mfspr”, “mtdcr”, and “mfocr” mnemonics with a definition of the SPR/DCR number. This way, you can include a file specific to the processor which defines the specific register numbers, and a 405GP compiler will create code that works equally well on the 440GP.

For example, instead of :

```
mfsrr0          %r28          /* Save/Restore Register 0 */  
mfsrr1          %r29          /* Save/Restore Register 1 */
```

use:

```
mfspr           %r28, SRR0_SPRN /* Save/Restore Register 0 */  
mfspr           %r29, SRR1_SPRN /* Save/Restore Register 1 */
```

OPB/PLB

While the processor still only as a 32-bit virtual address space, the PLB and OPB were changed to use 36-bit addressing to create a 64 GB real address space. This change primarily affects the MMU as well as the DMA, UIC, and PCI peripheral functions. Extra care is needed when writing the drivers for these peripherals in order to accommodate the additional 4 address bits. The MMU packs these address bits into an existing register, however the peripherals universally use a “high” and “low” 32-bit register scheme for extended addressing. Presumably, this allows an easy migration path all the way up to 64-bit real address space.

If your application can make due with a 4 GB real address space, the modifications are trivial. If the application requires a larger address space, the MMU allows plenty of options for dynamic changes in TLB entries (via TLB instruction and data miss exceptions) so that the application can use the extended address space with only a few low-level exception routine modifications.

Exceptions

While the 405GP allowed the exception table to be located anywhere in memory via the EVPR, the actual offsets from the base address were fixed. The 440GP allows the same base address option via the IVPR, but additionally allows the specific vectors to be at any offset within a 64k address space from the base address, though they must be 16-byte aligned.

This option allows for more detailed and efficient exception processing routines directly at the vector location. For compatibility however, it is recommended that you use the same offsets as defined by the 405GP processor.



Interrupts

The 405GP only used one bank of registers for the Universal Interrupt Controller. With its extended interrupts capabilities, the 440GP requires two banks of registers to accommodate all possible interrupt sources. In addition, the UIC1 bank is chained into the UIC0 bank. In this way, when an interrupt happens on the UIC1, it creates a UIC0 interrupt indicating that UIC1 needs to be checked.

This configuration likely requires a new low-level interrupt processing routine. However, by having these low-level routines feed back into a table driven mechanism linking generic vector numbers with specific interrupt masks, the application need not change. The application can then process interrupts using these generic vector numbers.

MMU – General

Upon startup with the 405GP, the entire memory map was available in real mode. The 440GP has no real mode, and only defines a 4 KB “shadow” TLB entry at 0xFFFFF000 upon startup. Startup code must then be located at this address which creates enough TLB entries for the main application to function before a full virtual address space is defined.

MMU - TLB entries

Each TLB entry on the 405GP packed all options into the TLBHI and TLBLO registers. With the additional four (4) addressing bits as well as new storage attributes for user and supervisor mode on the 440GP, each TLB entry uses three (3) registers deemed TLB0, TLB1, and TLB2.

While this will likely require a new routine to set up the TLB entries, the basic operation is the same. One major change however is the addition of a “readable” bit on the 440GP. While the 405GP did not include this bit in the storage attributes, without the bit set on the 440GP, the TLB entry is essentially useless even if the entry is marked executable.

PCI

On the 405GP the PCI controller is actually identified as device zero on the PCI bus, and therefore requires configuration cycles in order to configure some of the registers. The setup is actually simpler on the 440GP where the registers are directly memory mapped so that they can accept standard writes.

One important consideration for both processors, is that configuration is much easier if the TLB for this space is setup using little endian mode. All PCI registers use little endian mode, so having this TLB entry simplifies the setup steps.

Ethernet

The only difference between the 405GP and 440GP EMAC controllers is the number of controllers. The 405GP has one MAC controller built in while the 440GP has two. There are no functional differences, so by defining macros to automatically calculate register offsets for a particular controller, support for *N* EMAC controllers is easily possible.



PowerPC 440 GP to PowerPC 440 GX

Interrupts

The 440GX took a different approach to accommodating even more interrupts. The 440GX uses three banks of registers for the UIC. Rather than daisy chaining them similar to the 440GP, all three feed into the UICB0 register. In this way, all interrupts are registered in either UIC0, UIC1, or UIC2, and then that specific controller registers an interrupt in the UICB0 bank. Any time an interrupt is generated, it needs to be cleared in both sources to keep from instantly recurring.

Similar to the 405GP to 440GP conversion, this likely requires a new low-level processing routine. However, with a careful application design no changes should be required above the low level interrupt handlers.

Machine Check Exception

The machine check exception on the 440GX changed significantly from the 440GP, and this is detailed very specifically in AMCC's application note "Migrating to PPC440GX from PPC440GP: Software Considerations" (AN2001).

There is one key to keep software backwards compatible however. The new Machine Check Status Register (a SPR) can be cleared without causing any errors on the 405GP and 440GP even though it does not exist on those controllers.

Ethernet

The main difference on the 440GX is that it adds two additional EMAC controllers over the 440GP. However, as discussed in the previous section, macros can be used to access these new with no functional changes in the driver.

The 440GX does define some EMAC register bits differently than both the 405GP and 4405GP, but again, macros are sufficient to isolate these few bit masks and keep the EMAC driver consistent between processors.

Contact Information:

Kozio, Inc.
Longmont, Colorado
Phone: (303) 776-1356
sales@kozio.com
www.kozio.com

kDiagnostics™, kManufacturing™, kPOST™ and Flash-N-Run™ are registered trademarks of Kozio, all other trademarks are property of their respective owners.