SOFTWARE-VISIBLE DIFFERENCES BETWEEN THE R4650 AND THE R4600/R4700 ORION FAMILY MEMBERS

APPLICATION NOTE AN-135

By Ketan Deshpande

INTRODUCTION

The IDTR4650 is a low cost member of the IDTR4600 (Orion) family, targeted towards a variety of embedded applications. R4600 features not required in many embedded systems have been removed in the R4650 to lower device cost; others have been added to better suit the processor for its target applications. Given these changes in architecture, software designed to run on the Orion may need to be slightly modified to be able to take full advantage of the features of the R4650.

This Application note discusses the software visible changes integrated within the R4650; this information is required when porting existing low-level software (e.g. compilers, debuggers and other assembly language programs) from the R4600 to the R4650.

Architectural Differences

While a complete discussion of the architectural differences between the R4650 and the Orion is beyond the scope of this note, the relevant differences will be enumerated and software issues discussed. Some system control registers have been deleted, some new ones have been added, and some have been modified. Also, some exceptions are no longer generated, and some new exceptions can be generated.

1. Integer Execution Unit

The R4650 uses the same ALU as the Orion, with a few modifications:

a) Faster MULT/DMULT instructions.

As a result of the faster MULT / DMULT instructions, assemblers or assembly language programmers need not wait as many cycles as earlier to retrieve the result from the HI/LO registers.

For MULT instructions (32x32->64 bits) the R4650 detects the actual size of the operands; the execution time of the multiply is thus determined by the actual number of significant bits in the operands. For 16-bit operands, the time taken to perform a MULT instruction is 2 pipeline cycles (PCycles) and for 32-bit operands, the time is 3 PCycles.

The time to perform a DMULT operation(64x64->128 bits) is 5 cycles, irrespective of the size of the operands.

b) New instructions: MUL and MAD.

The MUL instruction can be used to multiply two CPU general purpose registers (GPRs) and store the result in another GPR (32x32->64-bits), bypassing the HI/LO pair, and eliminating the MFHI/MFLO instructions.

The MAD instruction multiplies two (32-bit) GPRs and adds the product to the contents of the HI/LO registers, storing the result in the HI/LO pair.

MUL and MAD are defined only for 32-bit numbers; there are no DMUL / DMAD instructions.

2. Control Processor 0 (CP0)

CP0 has been greatly changed from the original R4600 Orion. Only two modes: user and kernel are supported (selected by setting the UM bit in the STATUS register). All addresses (virtual and physical) are 32 bits. There is no 64-bit virtual address mode. All CP0 registers are now 32-bit, and the DMTC0/DMFC0 instructions are no longer valid. However, these instructions will not generate a trap.

a) PRId Register:

If the same software will be used to support the Orion and the R4650, CPU-specific code can be separated on the basis of the Implementation field of the PRId register in CP0, which is 0x22 for the R4650, and 0x20 for the Orion.

b)STATUS Register:

The STATUS register has a different format in the R4650.

i) It has a bit to lock set A of the I-Cache (the IL bit, bit 23), and one to lock set A of the Dcache (the DL bit, bit 24). Critical sections of the code / data may thus be locked into the cache for fast access. When locked, this set will not be chosen for line refill. However, a line in a locked set will still be chosen for refill if that line is invalid. Thus locked sets may be flushed without having to unlock them first. It takes 5 instructions after setting the IL bit for refills to be disabled, and 3 instructions after setting the DL bit.

ii) The FR bit (bit 26) can be set to select 16 or 32 32-bit floating point registers.

c) CAUSE Register:

The CAUSE register has a slightly different format. It has two new bits that denote whether the exception was due to IWatch or DWatch (bits 24 & 25 respectively, discussed below) and one bit (IV bit, bit 23) to force interrupts to use a different exception vector offset. On reset, Cause.IV is cleared; thus exceptions and interrupts use the same exception vector offset (0x180). When Cause.IV is set, interrupts use a new exception vector offset (0x200). This can be used for faster decoding of interrupts. This new exception vector did not exist in the R4600 Orion; thus, the use of a dedicated interrupt vector is an option, not a mandate, in the R4650. For systems whose performance is highly dependent on interrupts, additional software modifications may be desirable, since there may be code at that location that now needs to be moved, as well as moving the interrupt management code to that location.

d) TLB:
The R4650 does not include the R4600 Orion Memory Management Unit (MMU). The CP0 TLB registers 0-6, 10 and 20 have been removed. The instructions TLBR, TLBWI, TLBWR are no longer defined, but will not generate a trap. TLB exceptions like TLBMiss / XTLBMiss will never be generated. The exception vector offsets 0x000 and 0x080 are no longer used.
The R4650 performs virtual address translation based on Base/Bound register pairs. There are two sets of these pairs: One for Instruction and one for Data. In user mode, when an address is generated, it is compared with the base / bound register pair. If the address is "out of bounds", an exception is generated, with the appropriate ExcCode bits set in the Cause register (0x2 for Instruction, 0x3 for Data). An MTC0 instruction which changes any base / bound register must be done in unmapped space and mapped space cannot be entered for 5 instructions following a change to these registers. In kernel mode, kseg0 & kseg1 addresses undergo a fixed address translation; kuseg addresses follow the base/ bound translation.

e) Cache Algorithm Register:
The LLAddr register in the Orion has been replaced with the CAlg register, which defines the Cache Algorithm for each 512 MB region of the virtual address space. On reset, it gets initialized to 0x22233333, which is consistent with the Orion's interpretation of the K0 bits in its own CONFIG register. An MTC0 instruction should not change the field corresponding to the address space currently active. Doing so will cause undefined behavior.

f) Watch Registers:
Two new registers, IWatch and DWatch, greatly facilitate software debug. By setting the contents of the registers to the desired watch point and enabling the Watch Exception, an exception handler can be called every time the watch point is hit. The exception generated is at the general exception vector, with ExcCode = 0x23 in the Cause register. The IW/DW bits in the Cause register are set to denote whether the exception was caused by a Data Watch point or Instruction Watch point. The actual exception will be generated whenever both the ERL & EXL bits in the STATUS register are cleared. When DWatch is enabled, the two instructions immediately following may not be checked for match with the watch value. When IWatch is enabled, the 5 instructions following may not be checked for match with the watch value.

g) CONFIG Register:
The CONFIG register in the R4650 is read-only. The format has been modified: the IC & DC bits are both now 001, denoting the 8KB:8KB cache sizes. The K0 field has been deleted since this function has been expanded and is now performed by the CAlg register.

h) Other Registers:
The BadVAddr, EPC & ErrorEPC registers in the Orion were 64 bits; in the R4650 they are 32 bits wide.

3. Co-Processor 1 (CP1)

This is the Floating point coprocessor on board the R4650. The single biggest departure from the Orion is that the R4650 supports single precision operations only. The R4650 does not support double precision operations, which could be performed by an emulation library, if required. CP1 has a set of general purpose registers (FGRs) that are 32-bit wide, and can be accessed as a group of 16 or a group of 32 registers, by setting the FR bit in the CP0 STATUS register to 0 or 1, respectively. If STATUS.FR = 0, only even numbered FGRs can be accessed, and accessing an odd numbered register generates a trap. Any double precision operation in CP1 causes a trap to occur; thus a trap-based library could be written to emulate double-precision operations. DMFC1/ DMTC1 instructions will generate a trap.

There are two floating-point execution units in the R4650: one multiply unit and one unit for add/convert/divide/SQRT. As a result, multiplies and add/subtracts can be overlapped.

CONCLUSION

This Application Note discussed the issues involved in porting assembly code from the R4600 Orion to the R4650. Some relevant architectural differences were noted, with implications for software modification.

IMPORTANT NOTICE AND DISCLAIMER

RENESAS ELECTRONICS CORPORATION AND ITS SUBSIDIARIES (“RENESAS”) PROVIDES TECHNICAL SPECIFICATIONS AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for developers skilled in the art designing with Renesas products. You are solely responsible for (1) selecting the appropriate products for your application, (2) designing, validating, and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. Renesas grants you permission to use these resources only for development of an application that uses Renesas products. Other reproduction or use of these resources is strictly prohibited. No license is granted to any other Renesas intellectual property or to any third party intellectual property. Renesas disclaims responsibility for, and you will fully indemnify Renesas and its representatives against, any claims, damages, costs, losses, or liabilities arising out of your use of these resources. Renesas' products are provided only subject to Renesas' Terms and Conditions of Sale or other applicable terms agreed to in writing. No use of any Renesas resources expands or otherwise alters any applicable warranties or warranty disclaimers for these products.

(Rev.1.0 Mar 2020)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.