Debug12 Interrupt Usage Errata
Axiom Mfg. 05/01/00

Motorola documentation for the use of Interrupts under the Debug12 program is incorrect in most cases. This includes both the HC12A4 and HC12B32 versions of Debug12. Effected documents are the Motorola AN-1280 Using and Extending Debug12 Routines and the MC68HC912B32EVB Users Manual. Axiom had not previously documented interrupt usage under Debug12.

To correctly install interrupt vectors under Debug12, the user should place the interrupt vector service routine address into the Debug12 pseudo vector table located in the HC12 internal ram address at 0x0B00 – 0xB3F hex. Each Debug12 pseudo vector address is 2 bytes and is organized in the same order as the standard vector table. Following is the interrupt vector cross reference table:

Interrupt Source	Hardware Vector	Debug12 Vector
Reserved	0xFFC0 – FFCD	0x0B00 – 0X0B0D
A4 WAKE-UP H	0xFFCE – FFCF	0x0B0E – 0B0F
A4 WAKE-UP J /B32 BDLC	0xFFD0 – FFD1	0x0B10 – 0B11
ANALOG / DIGITAL CONV.	0xFFD2 – FFD3	0x0B12 – 0B13
A4 SCI1	0xFFD4 – FFD5	0x0B14 – 0B15
SCI0	0xFFD6 – FFD7	0x0B16 – 0B17
SPI	0xFFD8 – FFD9	0x0B18 – 0B19
PULSE ACC IN EDGE	0xFFDA – FFDB	0x0B1A – 0B1B
PULSE ACC OVERFLOW	0xFFDC – FFDD	0x0B1C – 0B1D
TIMER OVERFLOW	0xFFDE – FFDF	0x0B1E – 0B1F
TIMER CH 7	0xFFE0 – FFE1	0x0B20 – 0B21
TIMER CH 6	0xFFE2 – FFE3	0x0B22 – 0B23
TIMER CH 5	0xFFE4 – FFE5	0x0B24 – 0B25
TIMER CH 4	0xFFE6 – FFE7	0x0B26 – 0B27
TIMER CH 3	0xFFE8 – FFE9	0x0B28 – 0B29
TIMER CH 2	0xFFEA – FFEB	0x0B2A – 0B2B
TIMER CH 1	0xFFEC – FFED	0x0B2C – 0B2D
TIMER CH 0	0xFFEE – FFEF	0x0B2E – 0B2F
REAL TIME INT	0xFFF0 – FFF1	0x0B30 – 0B31
IRQ*	0xFFF2 – FFF3	0x0B32 – 0B33
XIRQ*	0xFFF4 – FFF5	0x0B34 – 0B35
SWI	0xFFF6 – FFF7	0x0B36 – 0B37
ILLEGAL OP CODE	0xFFF8 – FFF9	0x0B38 – 0B39
COP FAIL RESET	0xFFFA – FFFB	NOT PSEUDO VECTORED
COP MONITOR RESET	0xFFFC – FFFD	NOT PSUEDO VECTORED
RESET*	0xFFFE – FFFF	NOT PSEUDO VECTORED

Debug12 will test the pseudo vector address on the occurrence of an interrupt. If the pseudo vector is empty (0x0000), then the interrupt will be trapped and program execution will stop. Otherwise, program execution will continue at the user interrupt vector. Pseudo vector entry requires approximately 24 cycles additional delay over standard vector service.