MOTOROLA

TECHNICAL BULLETIN

DSP56L811 Digital Signal Processor, Mask 1H43A

Specification Changes and Enhancements

December 18, 1996

This short Technical Bulletin details important technical enhancements to the DSP56L811 Digital Signal Processor (DSP), mask 1H43A. Information on these enhancements is being incorporated into the *DSP56800 Family Manual (DSP56800FM/AD), DSP56L811 User's Manual (DSP56L811UM/AD),* and *DSP56L811 Technical Data (DSP56L811/D)* as detailed in this bulletin. Refer to these documents for complete information on these technical enhancements.

CORE/INSTRUCTION CHANGES

These design changes will be documented in the next revision of the *DSP56800 Family Manual (DSP56800FM/AD)* or the *DSP56L811 User's Manual (DSP56L811UM/AD),* as appropriate.

Change	*Description*
1. Instruction fetch from X memory	Instructions cannot be fetched from X memory. **Workaround**: Do not try to fetch instructions from X memory, and keep the X/P Memory (XP) bit in the Operating Mode Register (OMR) set to 0.
2. Data ALU instructions with accumulator as destination	When the Condition Code (CC) bit in the OMR is set, Data ALU instructions with an accumulator (A or B) as a destination check only the A1 or B1 portion of the accumulator to determine whether the Zero (Z) bit in the Status Register (SR) should be set. **Workaround**: When the CC bit in the OMR is set, separately test the individual portions of the accumulator (A1/A0 or B1/B0) when trying to determine whether the entire accumulator is 0.
3. MOVEs from an accumulator	When the CC bit in the OMR is set, a move from an accumulator (A or B) results in the data limiter checking the extension register (A2 of B2), even though the extension register should be ignored when the CC bit is set. A user should *not* move a 36-bit accumulator (A or B) unless it can be guaranteed that the extension (A2 or B2) is not in use for that accumulator. Instead, the user should move A1 or B1 directly. **Workaround**: When the CC bit in the OMR is set, read the accumulator out through its individual portions (A1 or B1).

4. Pipeline stalls in DO loops

An instruction sequence that automatically stalls the pipeline cannot be used at the end of a DO loop. The specific sequence to be avoided is a pipeline dependency resulting from a MOVE to the N register, followed by an Address Arithmetic Logic Unit (Address ALU) calculation involving the N register in the very next instruction. This instruction sequence is usually legal because the DSP core automatically stalls the pipeline one cycle. However, when this sequence is at the end of a DO loop, the loop is not executed the intended number of times, and should be avoided.

Workaround: Do not use an instruction sequence that stalls the pipeline at the end of a DO loop.

5. Saturation mode

Saturation mode is added to selectively limit overflow in Data ALU operations. A Saturation bit (SA) is implemented on Bit 4 of the OMR. This bit provides a Saturation mode for 16-bit algorithms that do not recognize or cannot take advantage of the extension accumulator.

When the SA bit is set, the saturation logic operates by checking three bits of the 36-bit result: two bits of the extension byte (exp[3] and exp[0]) and one bit of the Most Significant Portion (MSP) (msp[15]). The result obtained in the accumulator when SA = 1 is shown in the following table:

exp[3]	exp[0]	msp[5]	Result in Accumulator
0	0	0	Unchanged
0	0	1	$0 7FFF FFFF
0	1	0	$0 7FFF FFFF
0	1	1	$0 7FFF FFFF
1	0	0	$F 8000 0000
1	0	1	$F 8000 0000
1	1	0	$F 8000 0000
1	1	1	Unchanged

When SA = 0, no saturation of the Data ALU operation occurs.

6. LEA instruction enhancements

The following formats for the LEA instruction are added:

- LEA (R2 + xx) (1 word, 1 instruction cycle)
- LEA (SP – xx) (1 word, 1 instruction cycle)
- LEA (Rn + xxxx) (2 words, 2 instruction cycles)
- LEA (SP + xxxx) (2 words, 2 instruction cycles)

7. REP instruction restrictions

The following instructions are not allowed to follow a REP instruction:

- INCW x:<aa>
- DEC x:<aa>
- INCW x:(SP – xx)
- DEC x:(SP – xx)

8.	**Pipeline dependency on Stack Pointer**	Due to a pipeline dependency, an immediate move to the Stack Pointer (SP) should not be followed directly with a JSR instruction. **Workaround**: Insert a NOP between a MOVE to the SP followed by a JSR command.
9.	**Hardware stack overflow enhancement**	When the Hardware Stack (HWS) limit is exceeded, the oldest loop information (top-of-loop address and LF bit) is not updated; therefore, the loop information is not lost. This differs from the description in the *DSP56800 Family Manual (DSP56800FM/AD).* Workaround: None needed. This is the preferred implementation.
10.	**Carry bit update**	For an operation that involves an accumulator (A or B) and a 16-bit destination in the data memory (ADD F, x:<> for example), the Carry bit (CC) is updated out of Bit 35 of the result, not Bit 15 of the result as described in the *DSP56800 Family Manual (DSP56800FM/AD).* Workaround: To ensure correct 16-bit operation, the accumulator (A or B) should be sign-extended to the extension byte (A2/B2).
11.	**P-Fetch breakpoint limitation**	When using the On-Chip Emulation (OnCE™) functionality, if a P-Fetch breakpoint is set on the last instruction in a DO Loop, the core does not halt at that exact location. Instead, it halts at the first instruction in the loop instead. This is because the breakpoint is placed on the Program Address Bus during the DO instruction processing since it must be placed in the Loop Address Register. **Workaround**: Set the breakpoint outside the DO loop, or at some place inside the DO loop that is not the last instruction.

ELECTRICAL CHANGES

These design changes are documented in the *DSP56L811 Technical Data (DSP56L811/D), Rev. 2.*

Change		*Description*
1.	**Clock oscillator amplifier enhanced**	The on-chip clock oscillator amplifier circuit was redesigned to improve the robustness in supporting the specified range of input clock frequencies (32 kHz to 40 MHz). The functionality of PB15 has been modified. In addition to its General Purpose Input/Output functionality, it also functions as a control pin for the clock oscillator circuit. The pin name is changed to PB15/$\overline{\text{XCOLF}}$. During reset, this pin is tied to an on-chip pull-up transistor. $\overline{\text{XCOLF}}$ is read and internally latched in the DSP when the processor exits the Reset state. $\overline{\text{XCOLF}}$ selects the Crystal Oscillator amplifier mode of operation. A few cycles after reset is deasserted, this pull-up transistor is disabled. Full details of these enhancements are provided in *DSP56L811 Technical Data (DSP56L811/D).*

2. Asserting $\overline{\text{TRST}}$ with $\overline{\text{RESET}}$

To ensure complete hardware reset during normal operation, $\overline{\text{TRST}}$ should be asserted whenever $\overline{\text{RESET}}$ is asserted. The only exception is in a debug environment when $\overline{\text{RESET}}$ is asserted without requiring a JTAG port reset, which results from the assertion of $\overline{\text{TRST}}$.

3. External oscillator start-up time

External oscillator start-up time is no longer specified, as this specification depends on the external part and not on internal DSP circuitry.

4. PLL circuitry value changes

When using the on-chip oscillator in conjunction with an external crystal to generate the DSP clock, the following specifications apply.

Multiplication Factor	C_L	R	C
1024	5 nF	15 KΩ	15 nF
512	2.7 nF	15 KΩ	15 nF
256	2.7 nF	15 KΩ	15 nF
128	2.7 nF	15 KΩ	15 nF
100	2.7 nF	15 KΩ	15 nF
80	2.7 nF	15 KΩ	15 nF
40	2.7 nF	15 KΩ	15 nF
10	750 pF	2 KΩ	10 nF
4	750 pF	2 KΩ	10 nF
2	750 pF	2 KΩ	10 nF

Note: 1. Because of the high number of Multiplication Factors available, these are the only Multiplication Factors evaluated.

PERIPHERAL CHANGES

These design changes will be documented in the next revision of the *DSP56L811 User's Manual (DSP56L811UM/AD).*

	Change	*Description*
1.	**SSI write/read requires instruction delay**	When writing to the SSI Receive Control Register (SCRRX), an immediate access (any read or write) to the SSI Transmit Control Register (SCRTX) should not be performed, because unpredictable results may occur. **Workaround**: Insert a NOP or other instruction between read and write instructions for the SSI.
2.	**TDBE and TDE flags do not change state**	In the SSI, the Transmit Data Buffer Empty (TDBE) and the Transmit Data Empty (TDE) flag bits do not change state when the SSI is disabled by clearing the SSI Enable bit (SSIEN). **Workaround**: None available. The user needs to understand that these bits are not set when the SSIEN bit is cleared. It is recommended to disable the SSI only after the TDBE and TDE flags have been set, but this is not a requirement.
3.	**TDE flag cleared incorrectly**	In the SSI Network mode, the Transmit Data Empty (TDE) flag bit is set before the first Transmit Frame Sync is asserted in the first transmission after DSP reset. **Workaround**: Initialize the status circuitry by transmitting a dummy word in the first Network mode transmission after DSP reset. The TDE bit is set as previously specified thereafter.
4.	**TDBE flag cleared incorrectly**	In the SSI External Gated Clock mode, the Transmit Data Buffer Empty (TDBE) flag is cleared incorrectly when the Receive Data Register is read while the Transmit Data Empty (TDE) flag is cleared. **Workaround**: Do not poll the TDBE flag and use it to determine whether the Transmit Data Buffer is not in use.
5.	**New reset value for PCR1 register**	The PLL Control Register 1 (PCR1) is set to $0200 on reset, not cleared as described in the *DSP56L811 User's Manual* . This modification provides a division ratio of 16 for the prescaler clock, which ensures that all timers work reliably on restart. **Workaround**: None needed. Ensure that each timer is correctly initialized before using it.

Mfax and OnCE are trademarks of Motorola, Inc.

How to reach us:

USA/Europe/Locations Not Listed:
Motorola Literature Distribution
P.O. Box 5405
Denver, Colorado 80217
1 (800) 441-2447
1 (303) 675-2140

Asia/Pacific:
Motorola Semiconductors H.K. Ltd.
8B Tai Ping Industrial Park
51 Ting Kok Road
Tai Po, N.T., Hong Kong
852-2662928

Technical Resource Center:
1 (800) 521-6274

Japan:
Nippon Motorola Ltd.
Tatsumi-SPD-JLDC
6F Seibu-Butsuryu-Center
3-14-2 Tatsumi Koto-Ku
Tokyo 135, Japan
81-3-3521-8315

Mfax™:
RMFAX0@email.sps.mot.com
TOUCHTONE (602) 244-6609

DSP Helpline
dsphelp@dsp.sps.mot.com

Internet:
http://www.motorola-dsp.com

MOTOROLA

TECHNICAL BULLETIN ADDENDUM

DSP56L811 Digital Signal Processor, Mask 1H43A

Additional Specification Changes and Enhancements

February 18, 1997

CHANGES TO INSTRUCTION FUNCTIONS IN DEBUG MODE

This Technical Bulletin Addendum describes DSP56L811 Digital Signal Processor (DSP), mask 1H43A specification changes. These design changes will be documented in the next revision of the *DSP56800 Family Manual (DSP56800FM/AD)* or the *DSP56L811 User's Manual (DSP56L811UM/AD),* as appropriate.

Change	*Description*
1. Interrupt processing in Debug mode	If the chip is in Debug mode without having interrupts masked (i.e., Bits 8 and 9 of SR do not equal 11) and an interrupt is pending, interrupt processing begins when instructions are executed from Debug mode. This corrupts SR and SP and leads to at least one incorrect instruction execution in Debug mode. A user may encounter this feature when debugging code that uses general purpose timer interrupts, for example. If the chip is in Debug mode when the peripheral timer sends the interrupt, the interrupt request processing begins at the same time that the OnCE software executes instructions from Debug mode, (e.g., to display results or move memory contents). **Workaround**: Currently under development.

2. Single stepping of ADD a,x:<aa> and ADD a,x:(sp–xx) instructions	When single-stepping through an instruction sequence, if the sequence includes a 2-word opcode ADD a,x:<aa> or ADD a,x:(sp–xx) instruction, execution halts improperly in the middle of the instruction.
	For example, the opcode sequence for ADD a,x:$10 is $4010 $E042. Because execution halts in the middle of the instruction when single-stepping, the ADD a,x:$10 is always executed as ADD x:$10,a. The accumulator is improperly modified and the condition codes can be corrupted. Restoring the program flow when returning from Debug mode is also affected. If the program flow is not being changed from Debug mode, the value from OPDBR is written back twice, the second time with GO = EX = 1. But, in this case, $E042 is restored, which is not a valid opcode.
	The following instructions (opcode follows the ;) are affected:
	`ADD a,x:<aa>  ; 0100 0000 00aa aaaa $E042`
	`ADD b,x:<aa>  ; 0100 0000 10aa aaaa $E042`
	`ADD x0,x:<aa>  ; 0100 0010 00aa aaaa $E042`
	`ADD y0,x:<aa>  ; 0100 0011 00aa aaaa $E042`
	`ADD y1,x:<aa>  ; 0100 0011 10aa aaaa $E042`
	`ADD a,x:<sp – xx>  ; 0100 0000 01aa aaaa $E042`
	`ADD b,x:<sp – xx>  ; 0100 0000 11aa aaaa $E042`
	`ADD x0,x:<sp – xx>  ; 0100 0010 01aa aaaa $E042`
	`ADD y0,x:<sp – xx>  ; 0100 0011 01aa aaaa $E042`
	`ADD y1,x:<sp – xx>  ; 0100 0011 11aa aaaa $E042`
	where, aa = first 64 locations of X memory; xx = 1–63
	Workaround: Currently under development.

3. Shift-type instruction operation

If program execution is halted to enter Debug mode immediately before execution of a shift-type instruction with accumulator destinations, the accumulator is improperly written when instructions are executed from Debug mode.

For example if execution is halted and the next instructions is

```
LSSR b1,y1,b
```

the b register is updated by the shift operation of the LSSR instruction when an instruction is executed from Debug mode. The SR is not updated. Then when instruction flow is restored upon leaving Debug mode, the LSSR executes normally, causing a double-shift to occur. This affects all LSRR, ASRR, ASLL, and LMPY instructions using accumulator destinations.

The following instructions (opcode follows the ;) are affected:

```
LSRR y0,y0,a        ;752C
LSRR y0,y0,b        ;75AC
LSRR y1,y0,a        ;757C
LSRR y1,y0,b        ;75FC
LSRR a1,y0,a        ;753C
LSRR a1,y0,b        ;75BC
LSRR b1,y1,a        ;751C
LSRR b1,y1,b        ;759C
LSRR y1,x0,a        ;755C
LSRR y1,x0,b        ;75DC
LSRR y0,x0,a        ;754C
LSRR y0,x0,b        ;75CC

ASRR y0,y0,a        ;652C
ASRR y0,y0,b        ;65AC
ASRR y1,y0,a        ;657C
ASRR y1,y0,b        ;65FC
ASRR a1,y0,a        ;653C
ASRR a1,y0,b        ;65BC
ASRR b1,y1,a        ;651C
ASRR b1,y1,b        ;659C
ASRR y1,x0,a        ;655C
ASRR y1,x0,b        ;65DC
ASRR y0,x0,a        ;654C
ASRR y0,x0,b        ;65CC

ASLL y0,y0,a        ;672C
ASLL y0,y0,b        ;67AC
ASLL y1,y0,a        ;677C
ASLL y1,y0,b        ;67FC
ASLL a1,y0,a        ;673C
ASLL a1,y0,b        ;67BC
ASLL b1,y1,a        ;671C
ASLL b1,y1,b        ;679C
ASLL y1,x0,a        ;675C
ASLL y1,x0,b        ;67DC
ASLL y0,x0,a        ;674C
ASLL y0,x0,b        ;67CC
```

```
LMPY y0,y0,a        ;662C
LMPY y0,y0,b        ;66AC
LMPY y1,y0,a        ;667C
LMPY y1,y0,b        ;66FC
LMPY a1,y0,a        ;663C
LMPY a1,y0,b        ;66BC
LMPY b1,y1,a        ;661C
LMPY b1,y1,b        ;669C
LMPY y1,x0,a        ;665C
LMPY y1,x0,b        ;66DC
LMPY y0,x0,a        ;664C
LMPY y0,x0,b        ;66CC
```

Workaround: Currently under development.

Software supporting the DSP56L811 Evaluation Module (DSP56L811EVM) and the Motorola Application Development System (ADS) is currently under development to support these specification changes. When the revisions are completed, immediate distribution of the new software will be through the World-Wide Web. In cases where access to the Internet is restricted or unavailable, contact your local Motorola representative or distributor for additional information.

To obtain the Domain Technologies Debugger software that supports the DSP56L811EVM, refer to the following website:

ftp://ftp.domaintec.com/domtech

Select the latest file starting with e811_. The next three characters are the file revision number, followed by .zip to indicate that this is a compressed file. Download the file by double-clicking on the filename. The file must be de-compressed after downloading. Contact the Technical Resource Center by telephone or the DSP Helpline by email to determine the availability and the name of the file that supports the described specification changes.

Changes in the Motorola ADS software will be distributed via the Motorola DSP web site:

http://www.mot.com/SPS/DSP/software/index.html

Contact the Technical Resource Center by telephone or the DSP Helpline by email to determine the availability and the name of the file that supports the described specification changes.

Mfax and OnCE are trademarks of Motorola, Inc.

How to reach us:

USA/Europe/Locations Not Listed:
Motorola Literature Distribution
P.O. Box 5405
Denver, Colorado 80217
1 (800) 441-2447
1 (303) 675-2140

Asia/Pacific:
Motorola Semiconductors H.K. Ltd.
8B Tai Ping Industrial Park
51 Ting Kok Road
Tai Po, N.T., Hong Kong
852-26629298

Japan:
Nippon Motorola Ltd.
Tatsumi-SPD-JLDC
6F Seibu-Butsuryu-Center
3-14-2 Tatsumi Koto-Ku
Tokyo 135, Japan
81-3-3521-8315

Technical Resource Center:
1 (800) 521-6274

Mfax™:
RMFAX0@email.sps.mot.com
TOUCHTONE (602) 244-6609

DSP Helpline
dsphelp@dsp.sps.mot.com

Internet:
http://www.motorola-dsp.com