Appendix C

DSP16/DSP16A Instruction Set Summary

APPENDIX C. DSP16/DSP16A INSTRUCTION SET SUMMARY

CONTENTS

C. DSP16/DSP16A INSTRUCTION SET SUMMARY

Tables 3-2 through 3-12 are repeated below for the convenience of the user, to provide quick access to the DSP16/DSP16A instruction set. See Section 3.9 for a more detailed desciption of the instruction set.

Table C-1. Conditional Mnemonics

Test	Meaning	Test	Meaning
pl	Result is nonnegative (sign bit is bit 35).	mi	Result is negative.
eq	Result is equal to zero.	ne	Result is not equal to zero.
gt	Result is greater than zero.	le	Result is less than or equal to zero.
lvs	Logical overflow set (36-bit overflow).	lvc	Logical overflow clear.
mvs	Mathematical overflow set (32-bit overflow).	mvc	Mathematical overflow clear.
c0ge	Counter 0 greater than or equal to zero.	c0lt	Counter 0 less than zero.
c1ge	Counter 1 greater than or equal to zero.	c1lt	Counter 1 less than zero.
heads	Pseudorandom sequence bit set.	tails	Pseudorandom sequence bit clear.
true	The condition is always satisfied in an **if** instruction.	false	The condition is never satisfied in an **if** instruction.

Note: Testing the state of c0 or c1 automatically increments the counter by 1.

Table C-2. Instruction Group Characteristics

Instruction Group	Flags Affected	Execute From Within Cache	Interruptible
Multiply/ALU	DAU	Yes	Yes
Special Function	DAU	Yes	Yes
Control	None	No	No
Data Move	I/O status only	Yes*	Yes
Cache	None	No	No

* Two-word data immediate instructions may not be executed from within the cache.

All multiply/ALU instructions require one word of memory.

Table C-3. Multiply/ALU Instructions

Function Statements	Transfer Statements	Transfer Cycles Out/In Cache
	p=x*y	y=Y x=X	2/1
aD=p	p=x*y	y=aT x=X	2/1
aD=aS+p	p=x*y	y[l]=Y	1/1
aD=aS-p	p=x*y	aT[l]=Y	1/1
aD=p		x=Y	1/1
aD=aS+p		Y	1/1
aD=aS-p		Y=y[l]	2/2
aD=y		Y=aT[l]	2/2
aD=aS+y		Z: y x=X	2/2
aD=aS-y		Z: y[l]	2/2
aD=aS&y		Z: aT[l]	2/2
aD=aS|y			
aD=aS^y			
aS-y			
aS&y			

Table C-4. Replacement Table for Multiply/ALU Instructions

Replace	Value	Meaning
aD, aS, aT	a0, a1	One of two DAU accumulators.
X	*pt++,*pt++i	ROM location pointed to by pt. pt is postmodified by +1 and i, respectively.
Y	*rM, *rM++, *rM−−, *rM++j	RAM location pointed to by rM. (M= 0, 1, 2, 3) rM is postmodified by 0,+1,−1, and j, respectively.
Z	*rMzp, *rMpz, *rMm2, *rMjk	Read/write compound addressing. M=0, 1, 2, 3. rM is used twice: first, postmodified by 0, +1, −1, and j, respectively; and second, postmodified by +1, 0, +2, and k respectively.

www.DataSheet4U.com

All special function instructions (conditional and unconditional) require 1 word of program memory and execute in 1 instruction cycle.

Table C-5. Special Function Instructions

Instruction	Description
aD=aS>>1 aD=aS>>4 aD=aS>>8 aD=aS>>16	Arithmetic right shift (sign preserved) of 36-bit accumulators.
aD=aS aD=−aS	—
aD=rnd(aS)	Round upper 20 bits of accumulator.
aDh=aSh+1	Increment high half of accumulator (lower half cleared).
aD=aS+1	Increment accumulator.
aD=y aD=p	—
aD=aS<<1 aD=aS<<4 aD=aS<<8 aD=aS<<16	Logical left shift (sign-extended from bit 31) of the least significant 32 bits of the 36-bit accumulators.

The above special function instructions can be conditionally executed:

if CON *instruction*

and with an event counter:

ifc CON *instruction*

which means:

```
if CON is true then
        c1 = c1 + 1
        instruction
        c2 = c1
else
        c1 = c1 + 1
```

Table C-6. Replacement Table for Special Function Instructions

Replace	Value	Meaning
aD,aS	a0,a1	One of two DAU accumulators.
CON	mi, pl, eq, ne, gt, le, lvs, mvs, mvc, c0ge, c0lt, c1ge, c1lt, heads, tails, true, false	See Table C-1 for definitions of processor flags.

All unconditional control instructions (except icall) execute in 2 instruction cycles and require 1 word of program memory; conditional control instructions execute in 3 instruction cycles and require 2 words of program memory. **icall** requires 1 word of program memory and executes in 3 instruction cycles.

Table C-7. Control Instructions

goto *JA*	icall
goto pt	return (goto pr)
call *JA*	ireturn (goto pi)
call pt	

The control instructions, with the exception of ireturn and icall can be conditionally executed as follows:

if CON *instruction*

Table C-8. Replacement Table for Control Function Instructions

Replace	Value	Meaning
CON	mi, pl, eq, ne, gt, le, lvs, mvs, mvc, c0ge, c0lt, c1ge, c1lt, heads, tails, true, false	See Table 3-12 for definitions of processor flags.
JA	12-bit value	Least significant 12 bits of an absolute address within the same 4 Kword memory section.

Data move instructions execute in 2 instruction cycles. Immediate data move instructions require two words of program memory; all other data move instructions require only 1 word. The only exception is a special case immediate load (short immediate) instruction. If a YAAU register is loaded with a 9-bit, or smaller, value, the instruction requires only 1 word of memory and executes in 1 instruction cycle.

Table C-9. Data Move Instructions

R = N	R = M
R = Y	Y = R
aT = R	R = aS
Z : R	

www.DataSheet4U.com

Table C-10. Replacement Table for Data Move Instructions		
Replace	Value	Meaning
R	x	DAU register – signed, 16 bits.
	y	DAU register – signed, 16 bits.[1]
	yl	DAU register – unsigned, 16 bits.
	auc	DAU control register – unsigned, 7 bits.
	c0	DAU counter 0 – signed, 8 bits.
	c1	DAU counter 1 – signed, 8 bits.
	c2	DAU counter 2 – signed, 8 bits.
	r0	YAAU pointer reg. – unsigned, 9 bits (16 bits in DSP16A).
	r1	YAAU pointer reg. – unsigned, 9 bits (16 bits in DSP16A).
	r2	YAAU pointer reg. – unsigned, 9 bits (16 bits in DSP16A).
	r3	YAAU pointer reg. – unsigned, 9 bits (16 bits in DSP16A).
	rb	YAAU modulo addr. reg. – unsigned, 9 bits (16 bits in DSP16A).
	re	YAAU modulo addr. reg. – unsigned, 9 bits (16 bits in DSP16A).
	j	YAAU incr. register – signed, 9 bits (16 bits in DSP16A).
	k	YAAU incr. register – signed, 9 bits (16 bits in DSP16A).
	pt	XAAU pointer register – unsigned, 16 bits.
	pr	XAAU program return register – unsigned, 16 bits.
	pi	XAAU program interrupt register – unsigned, 16 bits.[2]
	i	XAAU incrmient register — signed, 12 bits.
	psw	Processor status word.
	sioc	Serial I/O control register.[3]
	sdx	Serial I/O data register.
	tdms	Serial I/O tdms control register.[3]
	srta	Serial receive/transmit address.[3]
	pioc	Parallel I/O control register.
	pdx0	Parallel I/O data register with PSEL = 0 (pin 72).
	pdx1	Parallel I/O data register with PSEL = 1 (pin 72).
aD, aS	a0, a1	High half of accumulator.[1]
Y	*rM,*rM++, *rM−−,*rM++j	Same as in multiply/ALU instructions.
Z	*rMzp,*rMpz, *rMm2,*rMjk	Same as in multiply/ALU instructions.
N	16-bit value	Immediate data.
M	9-bit value	Immediate data for YAAU registers.

Notes:
When reading signed registers less than 16 bits wide, their contents are sign-extended to 16 bits. When reading unsigned registers less than 16 bits wide, their contents are zero-extended to 16 bits. When short immediate addressing is used to write to YAAU registers in the DSP16A, unsigned registers are zero-extended from 9 to 16 bits. Signed registers (j,k) are sign-extended from 9 to 16 bits.

[1]Data moves to y, a0, or a1 load the high half (bits 31—16) of the register. If clearing of the destination is enabled (according to the CLR field of the auc register), the low half of the destination register is cleared (0) when the high half is loaded.

[2]The pi register acts as a "shadow" of the pc register. Each time the pc changes, its value is also loaded into pi. "Shadowing" is disabled when executing an interrupt service routine, therefore, pi contains the contents of pc prior to the interrupt. Writes to pi do not alter its contents, except during interrupt service routines.

[3]sioc, tdms, and srta registers are not readable.

The **do** and **redo** instructions require 1 word of program memory. The **do** instruction executes in 1 instruction cycle, and the **redo** instruction executes in 2 instruction cycles.

Table C-11. Cache Instructions

do *K* {
instruction1
instruction2
instructionNI
}

redo *K*

Table C-12. Replacement Table for Cache Instructions		
Replace	Value	Meaning
K	2 ≤ K ≤ 127	Number of times the instructions are to be executed.
NI	1 ≤ NI ≤ 15	1 to 15 instructions may be included.

When the cache is used to repeat a block of NI instructions, the cycle timings of the instructions are as follows:

1. The "first pass" does not affect cycle timing except for the last instruction in the block of NI instructions. This instruction executes in 2 cycles.
2. During pass 2 through pass K+1, each instruction is executed "in the cache" (see Table C-3).
3. During the last (Kth) pass, the block of instructions executes "inside the cache" except for the last instruction which executes outside the cache.

The instructions remain in the cache memory and may be re-executed using the **redo** command without the need to reload the cache.

Appendix D

Programmable Registers

APPENDIX D. PROGRAMMABLE REGISTERS

CONTENTS

D. PROGRAMMABLE REGISTERS

This reference section shows the six programmable control registers of the DSP16/DSP16A device:

- Processor status word (psw)
- Arithmetic unit control (auc)
- Parallel I/O control (pioc)
- Serial I/O control (sioc)
- Serial receive/transmit address (srta)
- Time-division multiplexed slot (tdms)

All six registers are described in detail in other chapters in this manual. This section is provided only as a quick reference for the programmable registers.

Table D-1. Arithmetic Unit Control (auc) Register

Bit	6	5	4	3	2	1	0
Field	CLR			SAT		ALIGN	

Field	Value	Result/Description
CLR	1xx	Clearing yl is disabled (enabled when 0).
	x1x	Clearing a1l is disabled (enabled when 0).
	xx1	Clearing a0l is disabled (enabled when 0).
SAT	1x	a1 saturation on overflow is disabled (enabled when 0).
	x1	a0 saturation on overflow is disabled (enabled when 0).
ALIGN	00	p ← (x×y).
	01	p ← (x×y) ÷ 4.
	10	p ← (x×y) × 4.
	11	Reserved.

Table D-2. Processor Status Word (psw) Register

Bit	15—12	11	10	9	8—5	4	3—0
Field	DAU Flags	X	X	a1[V]	a1[35–32]	a0[V]	a0[35–32]

Field	Value	Result/Description
DAU Flags	Wxxx	lMI – logical minus when set.
	xWxx	LEQ – logical equal when set.
	xxWx	LLV – logical overflow when set.
	xxxW	LMV – mathematical overflow when set.
a1[V]	W	Accumulator 1 (a1) overflow when set.
a1[35–32]	Wxxx	Accumulator 1 (a1) bit 35.
	xWxx	Accumulator 1 (a1) bit 34.
	xxWx	Accumulator 1 (a1) bit 33.
	xxxW	Accumulator 1 (a1) bit 32.
a0[V]	W	Accumulator 0 (a0) overflow when set.
a0[35–32]	Wxxx	Accumulator 0 (a0) bit 35.
	xWxx	Accumulator 0 (a0) bit 34.
	xxWx	Accumulator 0 (a0) bit 33.
	xxxW	Accumulator 0 (a0) bit 32.

Table D-3. Parallel I/O Control (pioc) Register

Bit	15	14	13	12	11	10	9—5	4—0
Field	IBF	STROBE		PODS	PIDS	S/C	INTERRUPTS	STATUS

Field	Value	Result/Description
IBF	R	IBF interrupt status bit (same as bit 4).
STROBE	00 01 10 11	Strobe width of PODS PIDS T* T 2T 2T 3T 3T 4T 4T
PODS	0	PODS is an input (passive mode).
	1	PODS is an output (active mode).
PIDS	0	PIDS is an input (passive mode).
	1	PIDS is an output (active mode).
S/C	0	Not S/C mode.
	1	S/C mode.
INTERRUPTS	Wxxxx	IBF interrupt enabled when set.
	xWxxx	OBE interrupt enabled when set.
	xxWxx	PIDS interrupt enabled when set.
	xxxWx	PODS interrupt enabled when set.
	xxxxW	INT interrupt enabled when set.
STATUS	Rxxxx	IBF status bit.
	xRxxx	OBE status bit.
	xxRxx	PIDS status bit.
	xxxRx	PODS status bit.
	xxxxR	INT status bit.

* T = 2×tCKIHCKIH.

Table D-4. Serial I/O Control (sioc) Register

Bit	9	8	7	6	5	4	3	2	1	0
Field	LD	CLK		MSB	OLD	ILD	OCK	ICK	OLEN	ILEN

Field	Value	Result/Description
LD	0	Active ILD/OLD = ICK÷16, Active SYNC = ICK÷128/256†
	1	Active ILD/OLD = OCK÷16,‡ Active SYNC = OCK÷128/256.†,‡
CLK	0 0	Active clock = CKI÷4
	0 1	Active clock = CKI÷12
	1 0	Active clock = CKI÷16
	1 1	Active clock = CKI÷20
MSB	0	LSB first
	1	MSB first
OLD	0	OLD is an input (passive mode).
	1	OLD is an output (active mode).
ILD	0	ILD is an input (passive mode).
	1	ILD is an output (active mode).
OCK	0	OCK is an input (passive mode).
	1	OCK is an output (active mode).
ICK	0	ICK is an input (passive mode).
	1	ICK is an output (active mode).
OLEN	0	16-bit output
	1	8-bit output
ILEN	0	16-bit input
	1	8-bit input

† Either 128 or 256 – see tdms register SYNC field.
‡ Select this mode when using SADD (not necessary if ICK = OCK).

Table D-5. Serial Receive/Transmit Address (srta) Register

Bit	15 — 8	7 — 0
Field	RECEIVE ADDRESS	TRANSMIT ADDRESS

Field	Value	Result/Description
RECEIVE ADDRESS	1xxxxxxx	Receive address 7
	x1xxxxxx	Receive address 6
	xx1xxxxx	Receive address 5
	xxx1xxxx	Receive address 4
	xxxx1xxx	Receive address 3
	xxxxx1xx	Receive address 2
	xxxxxx1x	Receive address 1
	xxxxxxx1	Receive address 0
TRANSMIT ADDRESS	1xxxxxxx	Transmit address 7
	x1xxxxxx	Transmit address 6
	xx1xxxxx	Transmit address 5
	xxx1xxxx	Transmit address 4
	xxxx1xxx	Transmit address 3
	xxxxx1xx	Transmit address 2
	xxxxxx1x	Transmit address 1
	xxxxxxx1	Transmit address 0

Table D-6. Time-Division Multiplex Slot (tdms) Register

Bit	9	8	7 — 1	0
Field	SYNCSP	MODE	TRANSMIT SLOT	SYNC

Field	Value	Result/Description
SYNCSP	0	SYNC = ICK/OCK† ÷ 128 ‡
	1	SYNC = ICK/OCK† ÷ 256
MODE	0	Multiprocessor mode off. DOEN is an input (passive mode).
	1	Multiprocessor mode on. DOEN is an output (active mode).
TRANSMIT SLOT	1xxxxxxx	Transmit slot 7.
	x1xxxxxx	Transmit slot 6.
	xx1xxxxx	Transmit slot 5.
	xxx1xxxx	Transmit slot 4.
	xxxx1xxx	Transmit slot 3.
	xxxxx1xx	Transmit slot 2.
	xxxxxx1x	Transmit slot 1.
SYNC	xxxxxxx1	Transmit slot 0. SYNC is an output (active mode).
	xxxxxxx0	SYNC is an input (passive mode).

† See sioc register, LD field in Table D-4.

‡ Select this mode when in multiprocessor mode.

www.DataSheet4U.com