

# I INDEX

## A

Absolute Value 3-26

Add/Add with Carry 3-5

Addressing branch targets 8-6

Adobe Acrobat 1-8

Alternate registers ( *see Secondary registers* )

ALU carry (AC) bit 2-5, 2-12, 2-16, 3-4, 3-6, 3-9, 3-13, 3-16, 3-19, 3-21, 3-24, 3-27, 3-30, 3-33, 3-37, 3-45, 4-9, 4-12, 4-15, 4-18, 4-20, 4-21, 4-25, 5-7, 5-9, 5-11, 5-13, 5-15, 5-18, 5-21, 5-24, 9-12

ALU data registers 2-3

ALU feedback (AF) register 3-2, 3-6, 3-9, 3-12, 3-13, 3-16, 3-19, 3-21, 3-23, 3-24, 3-27, 3-29, 3-30, 3-32, 3-33, 3-36, 3-44, 6-11, 6-15, 6-19, 9-19

ALU Instructions 2-19

ALU mode control 3-4

ALU negative (AN) bit 2-5, 3-4, 3-6, 3-9, 3-13, 3-16, 3-19, 3-21, 3-24, 3-27, 3-30, 3-33, 3-37, 3-45, 4-9, 4-12, 4-15, 4-18, 4-20, 4-21, 4-25, 5-7, 5-9, 5-11, 5-13, 5-15, 5-18, 5-21, 5-24

ALU or multiplier function (AMF) 9-1  
Codes 9-8

ALU overflow (AV) bit 2-5, 2-12, 2-15, 3-4, 3-6, 3-9, 3-13, 3-16, 3-19, 3-21, 3-24, 3-27, 3-30, 3-33, 3-37, 3-45, 4-9, 4-12, 4-15, 4-18, 4-20, 4-21, 4-25, 5-4, 5-7, 5-9, 5-11, 5-13, 5-15, 5-18, 5-21, 5-24, 9-11

ALU overflow latch enable

(AV\_LATCH) bit 2-12, 7-10, 8-69

ALU quotient (AQ) bit 2-6, 3-4, 3-6, 3-9, 3-13, 3-16, 3-19, 3-21, 3-24, 3-27, 3-30, 3-33, 3-37, 3-45, 4-9, 4-12, 4-15, 4-18, 4-20, 4-21, 4-25, 5-7, 5-9, 5-11, 5-13, 5-15, 5-18, 5-21, 5-24

ALU result (AR) register 2-3, 2-12, 3-2, 3-5, 3-6, 3-7, 3-8, 3-9, 3-10, 3-12, 3-13, 3-14, 3-15, 3-16, 3-18, 3-20, 3-21, 3-23, 3-24, 3-26, 3-27, 3-29, 3-30, 3-32, 3-33, 3-36, 3-44, 4-2, 4-9, 4-12, 4-15, 4-24, 5-3, 5-6, 5-8, 5-10, 5-12, 5-15, 5-17, 5-20, 5-23, 6-4, 6-11, 6-15, 6-19, 7-3, 7-10, 7-24, 7-28, 7-49, 7-52, 7-59, 7-61, 9-15, 9-19

# INDEX

- ALU saturation mode enable
  - (AR\_SAT) bit [2-12](#), [3-4](#), [7-10](#), [8-69](#)
- ALU signed (AS) bit [2-5](#), [3-4](#), [3-6](#), [3-9](#), [3-13](#), [3-16](#), [3-19](#), [3-21](#), [3-24](#), [3-27](#), [3-30](#), [3-33](#), [3-37](#), [3-45](#), [4-9](#), [4-12](#), [4-15](#), [4-18](#), [4-20](#), [4-21](#), [4-25](#), [5-7](#), [5-9](#), [5-11](#), [5-13](#), [5-15](#), [5-18](#), [5-21](#), [5-24](#), [8-22](#), [9-1](#), [9-40](#)
- ALU status flags [3-4](#)
- ALU x- and y-input (AX AY) registers [1-9](#), [2-3](#), [3-5](#), [3-6](#), [3-8](#), [3-9](#), [3-12](#), [3-15](#), [3-16](#), [3-18](#), [3-20](#), [3-21](#), [3-23](#), [3-26](#), [3-29](#), [3-32](#), [3-36](#), [3-44](#), [4-9](#), [4-12](#), [4-15](#), [5-6](#), [5-8](#), [5-10](#), [5-12](#), [5-15](#), [5-17](#), [5-20](#), [5-23](#), [6-11](#), [6-15](#), [6-19](#), [7-3](#), [7-24](#), [7-28](#), [7-49](#), [7-52](#), [7-59](#), [7-61](#), [9-3](#), [9-5](#), [9-14](#), [9-19](#)
- ALU zero (AZ) bit [2-5](#), [3-4](#), [3-6](#), [3-9](#), [3-13](#), [3-16](#), [3-19](#), [3-21](#), [3-24](#), [3-27](#), [3-30](#), [3-33](#), [3-37](#), [3-45](#), [4-9](#), [4-12](#), [4-15](#), [4-18](#), [4-20](#), [4-21](#), [4-25](#), [5-7](#), [5-9](#), [5-11](#), [5-13](#), [5-15](#), [5-18](#), [5-21](#), [5-24](#)
- Analog Devices product information [1-6](#)
- AND, OR, XOR [3-15](#)
- Arithmetic shift [5-6](#)
- Arithmetic shift (ASHIFT) instruction [5-3](#), [5-7](#), [5-9](#), [5-26](#)
- Arithmetic shift immediate [5-8](#)
- Arithmetic status (ASTAT) register
  - [2-4](#), [2-5](#), [2-12](#), [3-4](#), [3-6](#), [3-9](#), [3-13](#), [3-16](#), [3-19](#), [3-20](#), [3-21](#), [3-24](#), [3-27](#), [3-30](#), [3-33](#), [3-37](#), [3-45](#), [4-7](#), [4-9](#), [4-12](#), [4-15](#), [4-18](#), [4-20](#), [4-21](#), [4-25](#), [5-5](#), [5-7](#), [5-9](#), [5-11](#), [5-13](#), [5-15](#), [5-18](#), [5-21](#), [5-24](#), [7-2](#), [7-3](#), [7-10](#), [7-11](#), [7-22](#), [7-28](#), [7-32](#), [7-35](#), [7-39](#), [7-43](#), [7-47](#), [7-50](#), [7-53](#), [7-56](#), [7-58](#), [7-60](#), [7-62](#), [7-64](#), [7-65](#), [8-8](#), [8-9](#), [8-13](#), [8-16](#), [8-49](#), [8-56](#), [9-14](#)
- Arithmetic status bits [2-5](#), [2-6](#)
- Assembler [1-4](#)
- Audience (intended) [1-1](#)
- B**
- Background registers ( *see Secondary registers* )
- Base (B0-7) registers [1-9](#), [2-3](#), [7-6](#), [7-8](#), [7-15](#), [7-61](#), [7-62](#)
- Bias rounding enable (BIASRND) bit [2-8](#), [4-5](#)
- Bit-reversed addressing [7-17](#)
- Bit-reversed addressing enable (BIT\_REV) bit [2-11](#), [7-10](#), [7-17](#), [8-69](#)
- Branch options [8-5](#)
- Branching, preparation [8-31](#)
- Bus exchange, program memory (PX) register [2-4](#)

## C

Cable for information 1-6  
 Cache control (CACTL) register  
     7-61  
 CALL 8-5  
 CALL (PC relative) 8-30  
 CALL instruction 8-5, 8-6, 8-7, 8-8,  
     8-19, 8-20, 8-25, 8-28, 8-30,  
     8-31, 8-38, 8-42, 8-43, 8-46  
 Carry, ALU (AC) bit 2-5  
 Circular buffer addressing 6-8, 6-16  
     Restrictions 7-17  
 Circular data buffers 7-15  
 Clear bit (CLRBIT) instruction  
     3-18  
 Clear Interrupt 8-64  
 Clear interrupt (CLRINT)  
     instruction 9-60  
 Clear multiplier instruction 4-17  
 Compiler 1-4  
 Compute with dual memory read  
     6-3  
 Compute with memory read 6-10  
 Compute with memory write 6-14  
 Compute with register to register  
     move 6-18  
 Condition code (CCODE) register  
     2-4, 2-6, 2-7, 7-3, 7-10, 7-11,  
     7-22, 7-28, 7-32, 7-35, 7-39,  
     7-43, 7-47, 7-50, 7-53, 7-56,  
     7-58, 7-60, 7-62, 7-64, 7-65,  
     8-4, 9-12, 9-14  
 Condition codes 2-15, 9-11

Conditional instruction (COND)  
     codes 1-8, 2-15  
 Conditional Instructions 1-3  
 Conditions 8-2  
 Conditions, loop termination 8-22  
 Constant Codes 9-12  
 Constants 3-3  
 Contact information 1-6  
 Context switching 8-16  
 Conventions 1-8  
 Core Register Codes 9-13  
 Core registers 2-3, 7-2  
 Counter (CNTR) register 7-10,  
     7-28, 8-3, 8-7, 8-10, 8-11,  
     8-23, 8-55, 9-15  
     Operation 8-24  
 Counter expired (CE) condition  
     2-16, 8-3, 8-10, 8-12, 8-22,  
     8-23, 8-58, 9-7, 9-11, 9-12  
 Counter-based conditions 8-3  
 Customer support 1-6

## D

DAG1 registers (G1reg) 7-4  
 DAG2 registers (G2reg) 7-4  
 Data Address Generator (DAG)  
     Register latency 7-25  
     Register restrictions 7-14  
     Registers 7-6  
 Data addressing methods 7-12  
 Data format options 4-3  
 Data formats 4-4

# INDEX

- Data memory page (DMPGx)
  - registers 2-4, 7-3, 7-7, 7-25, 7-28, 7-32, 7-35, 7-39, 7-42, 7-47, 7-50, 7-53, 7-56, 7-58, 9-15
- Data move instructions 2-22
- Data registers (Dreg) 1-9, 3-5, 3-8, 3-12, 3-15, 3-20, 3-23, 3-26, 3-29, 3-32, 4-2, 4-9, 4-12, 4-15, 4-16, 5-6, 5-8, 5-10, 5-12, 5-15, 5-17, 5-20, 5-23, 6-7, 6-11, 6-15, 6-19, 7-4
  - X-input only (XOP) 1-9
  - Y-input only (YOP) 1-9
- Decrement 3-32
- Delayed branch (DB) option 8-5, 8-21, 8-28
  - Latency 8-32, 8-44
  - Restrictions 8-5, 8-31, 8-43, 8-46
- Denormalization 5-26
- Development tools 1-3
- Direct addressing 7-12
- Direct JUMP (PC relative) 8-27
- Direct memory read/write—
  - immediate address 7-24
- Direct register load 7-27
- Disable (DIS) instruction 8-69, 8-70
- Divide 3-35
- Divide primitives (DIVS DIVQ) 6-3, 6-10, 6-14, 6-18
- Divide, bit-wise quotient (DIVQ) instruction 6-3, 6-10, 6-14, 6-18
- Divide, sign of (DIVS) instruction 6-3, 6-10, 6-14, 6-18
- Division
  - Applications 3-43
  - Code example 3-43
  - Exceptions 3-41
  - Integer 3-41
  - Signed 3-39
  - Theory 3-38
  - Unsigned 3-40
- DMI, DMM, PMI, and PMM
  - Codes 9-18
- DO UNTIL instruction 7-10, 8-3, 8-5, 8-7, 8-10, 8-22, 8-23, 8-28, 8-31, 8-43, 8-46
- Dual memory read 6-7
- E
  - E-mail for information 1-6
  - Emulator cycle counter interrupt
    - enable (EMUCNTE) bit 2-9
  - Emulator interrupt mask (EMU) bit 2-10
  - ENA/DIS mode 7-9
  - Enable (ENA) instruction 8-69, 8-70
  - Ending loops, termination 8-22
  - Equal to zero (EQ) condition 2-15, 9-11
  - Execution order, multifunction 6-2
  - Exponent (block) adjust 5-23
  - Exponent adjust (EXPADJ) instruction 5-3, 5-23, 5-24
  - Exponent derive 5-20

Exponent derive (EXP) instruction  
[5-3](#), [5-14](#), [5-15](#), [5-21](#), [5-26](#)  
 Exponent, shifter (SE) register [2-4](#)  
 Exponent, shifter block (SB) register  
[2-4](#)  
 External IO port read/write [7-59](#)

## F

Far absolute branches [8-6](#)  
 FAX for information [1-6](#)  
 File Transfer Protocol (FTP) site  
[1-6](#)  
 Finite loops [8-10](#)  
 FLUSH CACHE [8-61](#)  
 FOREVER condition [8-12](#), [8-22](#),  
[8-23](#), [8-24](#), [8-58](#)  
 Fractional mode ( *see Multiplier  
 results mode* ) [2-13](#)

## G

Generate ALU Status Only [3-44](#),  
[4-24](#)  
 Global interrupt enable [8-14](#)  
 Global interrupt enable (GIE) bit  
[2-8](#)  
 Greater than (GT) condition [2-15](#),  
[9-11](#)  
 Greater than or equal to zero (GE)  
 condition [2-15](#), [9-11](#)  
 GREG Codes [9-18](#)

## H

High half output shift (HI) option  
[5-3](#), [5-4](#), [5-7](#), [5-9](#), [5-11](#), [5-13](#),  
[5-15](#), [5-18](#), [5-20](#), [5-21](#), [5-24](#)  
 High half output shift unless  
 overflow (HIX) option [5-3](#), [5-4](#),  
[5-20](#), [5-21](#)

## I

I and M Codes [9-17](#)  
 I/O memory page (IOPG) register  
[2-4](#), [7-3](#), [7-11](#), [7-28](#), [7-59](#),  
[7-60](#), [7-64](#), [7-65](#), [9-15](#)  
 Idle [8-67](#)  
 IDLE instruction [8-67](#)  
 Increment [3-29](#)  
 Index (IO-7) registers [2-3](#), [6-4](#), [6-7](#),  
[6-11](#), [6-15](#), [7-3](#), [7-6](#), [7-7](#), [7-15](#),  
[7-17](#), [7-24](#), [7-28](#), [7-30](#), [7-32](#),  
[7-34](#), [7-35](#), [7-37](#), [7-39](#), [7-41](#),  
[7-42](#), [7-46](#), [7-47](#), [7-50](#), [7-53](#),  
[7-55](#), [7-56](#), [7-57](#), [7-58](#), [7-63](#),  
[7-65](#), [8-7](#), [8-42](#), [8-45](#), [9-3](#), [9-4](#),  
[9-5](#), [9-13](#), [9-14](#), [9-17](#), [9-18](#),  
[9-19](#)  
 Indirect 16-bit memory  
 read/write—postmodify [7-30](#)  
 Indirect 16-bit memory  
 read/write—premodify [7-34](#)  
 Indirect 16-bit memory write—  
 immediate data [7-55](#)  
 Indirect 24-bit memory  
 read/write—postmodify [7-37](#)

## INDEX

- Indirect 24-bit memory
  - read/write—premodify [7-41](#)
- Indirect 24-bit memory write—
  - immediate data [7-57](#)
- Indirect addressing [1-9](#), [7-12](#)
- Indirect branches [8-7](#)
- Indirect CALL [8-42](#)
- Indirect DAG register write
  - (premodify or postmodify),
    - with DAG register move [7-45](#)
- Indirect JUMP [8-45](#)
- Indirect jump memory page (IJPG)
  - register [2-4](#), [7-3](#), [7-11](#), [7-28](#),  
[7-32](#), [7-35](#), [7-39](#), [7-43](#), [7-47](#),  
[7-50](#), [7-53](#), [7-56](#), [7-58](#), [7-60](#),  
[7-62](#), [7-64](#), [7-65](#), [8-42](#), [8-45](#),  
[8-46](#), [9-15](#)
- Indirect memory read/write—
  - immediate postmodify [7-49](#)
- Indirect memory read/write—
  - immediate premodify [7-52](#)
- Infinite loops [8-11](#), [8-24](#), [8-58](#)
- Input registers [3-2](#)
- Input registers, multiplier [4-2](#)

- Instruction
- (Type 1) Compute | DregX«...DM | DregY«...PM 9-21
  - (Type 3) Dreg/Greg «...»DM/PM 9-22
  - (Type 4) Compute | Dreg «...»DM/PM 9-23
  - (Type 6) Dreg «...Data16 9-24
  - (Type 7) Reg1/2 «...Data16 9-25
  - (Type 8) Compute | Dreg1 «...Dreg2 9-26
  - (Type 9) Compute 9-27, 9-30
  - (Type 10) Direct Jump/Call 9-32, 9-33
  - (Type 11) Do ... Until 9-34
  - (Type 12) Shift | Dreg «...» DM/PM 9-35
  - (Type 14) Shift | Dreg1 «... Dreg2 9-36
  - (Type 15) Shift Data8 9-37
  - (Type 16) Shift Reg0 9-38
  - (Type 17) Dreg1 «... Dreg2 9-39
  - (Type 18) Mode Change 9-40
  - (Type 19) Indirect Jump/Call 9-41
  - (Type 20) Return 9-42
  - (Type 21) Modify Dag1 9-43
  - (Type 22) DM/PM «... Data16/24 9-45, 9-46
  - (Type 23) Divide primitive, DIVQ 9-47
  - (Type 24) Divide primitive, DIVs 9-48
  - (Type 25) Saturate 9-49
  - (Type 26) Push/Pop/Cache 9-50
  - (Type 29) Dreg «...» DM/PM 9-51
  - (Type 30) NOP 9-52
  - (Type 31) Idle 9-53
  - (Type 32) Reg «...» DM 9-54, 9-55
  - (Type 33) Reg «... Data12 9-56
  - (Type 34) Dreg «...» IO 9-57
  - (Type 35) Dreg «...» Reg 9-58
  - (Type 36) Long Jump/Call 9-59
  - (Type 37) Interrupt 9-60
- Instruction pipeline 8-32, 8-44
- Instruction Set Notation 2-17
- Instructions
- Summary 2-17
- Integer division 3-41
- Integer mode ( *see Multiplier results mode* ) 2-13
- Interrupt control (ICNTL) register 2-4, 2-8, 4-5, 7-3, 7-10, 7-11, 7-28, 8-70, 9-15
- Interrupt latch (IRPTL) register 2-4, 2-10, 7-3, 7-28, 8-4, 8-14, 8-15, 8-62, 8-64, 9-15
- Interrupt mask (IMASK) register 2-4, 2-10, 7-3, 7-10, 7-28, 8-4, 8-14, 8-15, 8-62, 9-15
- Interrupt nesting enable (INE) bit 2-8
- Interrupt service routine (ISR) 8-9, 8-13, 8-48

# INDEX

- Interrupts 8-13
  - Enabling 8-14
  - Nesting 8-16
- Interrupts enable (INT) bit 8-69
- Interrupts, global enable (GIE) bit 2-8
  
- J**
- JUMP (PC relative) 8-34
- JUMP instruction 8-5, 8-6, 8-25, 8-27, 8-28, 8-31, 8-34, 8-37, 8-40, 8-43, 8-45, 8-46
  - Long jumps 8-20
  
- L**
- Latency 8-21
  - Delayed branch 8-32
  - ENA/DIS modes 8-70
  - Registers 7-9, 7-10, 7-25
- LCALL instruction 8-20, 8-32, 8-38
- Length (L0-7) registers 2-3, 6-8, 6-12, 6-16, 7-3, 7-6, 7-8, 7-15, 7-28, 9-15
- Less than or equal to zero (LE) condition 2-15, 9-11
- Less than zero (LT) condition 2-15, 9-11
- Level (stack) interrupt 8-13
- Linear addressing 6-16
- Linker 1-5
- LJUMP instruction 8-20, 8-28, 8-41
- Loader 1-5
  
- Logical shift 5-10
- Logical shift (LSHIFT) instruction 5-3, 5-7, 5-9, 5-11, 5-13, 5-26
- Logical shift immediate 5-12
- Long CALL 8-37
- Long calls 8-20
- Long JUMP 8-40
- Loop PC stack address (LPCSTACKA) register 2-4, 7-3, 7-28, 9-15
- Loop PC stack interrupt enable (LPCSTKE) bit 2-9
- Loop PC stack pointer (LCPSTACKP) register 2-4
- Loop PC STACK pointer (LPCSTACKP) register 7-3
- Loop PC stack pointer (LPCSTACKP) register 7-3, 7-28
- Loop stack address (LPSTACKA) register 8-11, 8-55
- Loop stack empty status (LPSTKEMPTY) bit 2-14, 8-22, 8-31, 8-38, 8-43, 8-49, 8-53, 8-56, 8-63, 8-65
- Loop stack full status (LPSTKFULL) bit 2-14, 8-22, 8-31, 8-38, 8-43, 8-49, 8-53, 8-56, 8-63, 8-65
- Loop stack pointer (LPSTACKP) register 8-11, 8-24, 8-55, 9-14
- Loop stacks Operation 8-10
- Loop termination 8-58
- Loop termination conditions 8-22

- Loops
  - Exiting 8-17
  - Restrictions 8-24, 8-52
- Loops, finite 8-10
- Loops, infinite 8-11
- Loops, PUSH/POP 8-11
- Low half output shift (LO) option
  - 5-3, 5-4, 5-7, 5-9, 5-11, 5-13, 5-15, 5-18, 5-20, 5-21
  
- M
- MAC output registers 4-2
- Mailing address for information 1-7
- Mode Control 8-69
- Mode status (MSTAT) register 2-4, 2-11, 3-4, 4-3, 4-6, 7-2, 7-3, 7-8, 7-10, 7-11, 7-17, 7-22, 7-28, 7-32, 7-35, 7-39, 7-43, 7-47, 7-50, 7-53, 7-56, 7-58, 7-60, 7-62, 7-64, 7-65, 8-4, 8-8, 8-9, 8-13, 8-16, 8-49, 8-56, 8-69, 9-14
- Modify (M0-7) registers 2-3, 6-2, 6-4, 6-7, 6-11, 6-15, 7-3, 7-6, 7-15, 7-24, 7-28, 7-30, 7-34, 7-37, 7-41, 7-46, 7-55, 7-57, 7-63, 9-3, 9-4, 9-5, 9-14, 9-15, 9-17, 9-18, 9-19
- Modify address register—direct 7-65
- Modify address register—indirect 7-63
- Multifunction execution order 6-2
- Multifunction instruction
  - Stall cycles 6-5, 6-9, 6-12, 6-16
- Multifunction instructions 2-24, 6-1
- Multiplier clear instruction 4-17
- Multiplier data registers 2-3
- Multiplier input registers 4-2
- Multiplier instructions 2-20
- Multiplier overflow (MV) bit 2-6, 2-16, 3-6, 3-9, 3-13, 3-16, 3-19, 3-21, 3-24, 3-27, 3-30, 3-33, 3-37, 3-45, 4-6, 4-7, 4-9, 4-12, 4-15, 4-18, 4-20, 4-21, 4-22, 4-25, 5-7, 5-9, 5-11, 5-13, 5-15, 5-18, 5-21, 5-24, 9-12
- Multiplier result (MR) register 4-24
- Multiplier result (MR0-2) registers 2-3, 3-5, 3-6, 3-8, 3-9, 3-12, 3-15, 3-16, 3-18, 3-20, 3-21, 3-23, 3-26, 3-29, 3-32, 3-36, 3-44, 4-2, 4-3, 4-5, 4-6, 4-9, 4-12, 4-14, 4-15, 4-17, 4-19, 4-21, 4-22, 4-24, 5-3, 5-6, 5-8, 5-10, 5-12, 5-15, 5-17, 5-20, 5-23, 6-4, 6-11, 6-15, 6-19, 7-3, 7-22, 7-24, 7-25, 7-28, 7-49, 7-52, 7-59, 7-61, 9-15, 9-20
- Multiplier results mode selection (M\_MODE) bit 2-13, 4-3, 4-6, 7-10, 8-69, 9-4, 9-40
- Multiplier round/transfer instruction 4-19

# INDEX

Multiplier saturate instruction [4-21](#)  
Multiplier status bits [2-6](#), [2-8](#)  
Multiplier x- and y-input (MX MY)  
  registers [1-9](#), [3-5](#), [3-8](#), [3-12](#),  
  [3-15](#), [3-20](#), [3-23](#), [3-26](#), [3-29](#),  
  [3-32](#), [4-2](#), [4-9](#), [4-12](#), [4-15](#),  
  [4-24](#), [5-6](#), [5-8](#), [5-10](#), [5-12](#),  
  [5-15](#), [5-17](#), [5-20](#), [5-23](#), [6-11](#),  
  [6-15](#), [6-19](#), [7-3](#), [7-24](#), [7-28](#),  
  [7-49](#), [7-52](#), [7-59](#), [7-61](#), [9-3](#),  
  [9-5](#), [9-14](#), [9-15](#), [9-19](#)  
Multiply [4-8](#)  
Multiply with cumulative add [4-11](#)  
Multiply with cumulative subtract  
  [4-14](#)

## N

Negate [3-23](#)  
Negative, ALU (AN) bit [2-5](#)  
Nested interrupts [8-16](#)  
No Operation [8-66](#)  
Normalize [5-14](#)  
Normalize (NORM) instruction  
  [5-2](#), [5-3](#), [5-14](#), [5-15](#), [5-18](#), [5-23](#)  
Normalize immediate [5-17](#)  
Normalizing ALU results [5-3](#), [5-4](#),  
  [5-20](#), [5-21](#)  
Not equal to zero (NE) condition  
  [2-15](#), [9-11](#)  
Numeric format modes [4-6](#)

## O

Opcode definitions [9-20](#)

Opcode mnemonics [9-1](#)  
Optimizing performance [8-17](#)  
Output registers [3-2](#)  
Overflow, ALU (AV) bit [2-5](#)  
Overflow, multiplier (MV) bit [2-6](#)  
Overflow, shifter (SV) bit [2-6](#)  
Overflow, stack [8-13](#)

## P

Page, data memory (DMPGx)  
  registers [2-4](#)  
Page, I/O memory (IOPG) register  
  [2-4](#)  
Page, indirect jump memory (IJPG)  
  register [2-4](#)  
PASS [3-20](#)  
PC [8-7](#)  
PC loop stack address  
  (LPCSTACKA) register [2-4](#)  
PC loop stack interrupt enable  
  (LPCSTKE) bit [2-9](#)  
PC loop stack pointer  
  (LCPSTACKP) register [2-4](#)  
PC relative branches [8-6](#)  
PC stack  
  Restrictions [8-7](#)  
PC stack address (STACKA)  
  register [2-4](#)  
PC stack empty status  
  (PCSTKEMPTY) bit [2-14](#),  
  [8-22](#), [8-31](#), [8-38](#), [8-43](#), [8-49](#),  
  [8-53](#), [8-56](#), [8-63](#), [8-65](#)

- PC stack full status (PCSTKFULL)
    - bit 2-14, 8-22, 8-31, 8-38, 8-43, 8-49, 8-53, 8-56, 8-63, 8-65
  - PC stack interrupt enable (PCSTKE) bit 2-8
  - PC stack level status (PCSTKLVL)
    - bit 2-14, 8-13, 8-22, 8-31, 8-38, 8-43, 8-49, 8-53, 8-56, 8-63, 8-65
  - PC stack operation 8-8
  - PC stack pointer (STACKP) register 2-4
  - POP instruction 8-7, 8-9, 8-11, 8-17, 8-24, 8-28, 8-31, 8-43, 8-46, 8-58
    - Automatic pops 8-58
  - POP Stacks 8-55
  - Postmodify addressing 1-9, 6-12, 6-16, 7-13, 7-45
  - Powerdown interrupt mask (PWDN) bit 2-10
  - Premodify addressing 1-9, 7-13, 7-45
  - Program counter (PC) register 8-22, 8-27, 8-30, 8-34
  - Program flow instructions 2-23
  - Program memory bus exchange (PX) register 2-4, 6-5, 6-8, 6-12, 6-16, 7-3, 7-4, 7-5, 7-28, 7-37, 7-38, 7-41, 7-42, 9-15
  - Programming information 1-1
  - PUSH instruction 8-7, 8-9, 8-11, 8-24, 8-28, 8-31, 8-43, 8-46, 8-57
    - Automatic pushes 8-57
  - PUSH Stacks 8-55
  - PUSH/POP loops 8-11
- Q**
- Quotient (division) 3-35
  - Quotient, ALU (AQ) bit 2-6
- R**
- Register to register move 7-22
  - Registers
    - DSP core 2-3
    - Load latencies 7-9
    - Register groups (Reg0-3) 7-2, 7-4, 7-27
  - Related documents 1-7
  - Result registers 2-3
  - Return from Interrupt 8-48
  - Return from interrupt (RTI)
    - instruction 8-5, 8-7, 8-9, 8-13, 8-16, 8-24, 8-28, 8-31, 8-43, 8-46, 8-48, 8-58
  - Return from Subroutine 8-52
  - Return from subroutine (RTS)
    - instruction 8-5, 8-7, 8-8, 8-24, 8-28, 8-31, 8-38, 8-43, 8-46, 8-52, 8-58
    - Restrictions 8-52
  - Round/transfer multiplier
    - instruction 4-19

# INDEX

Rounding (RND) multiplier option  
4-3, 4-19

Rounding modes 4-4

## S

Saturate multiplier instruction 4-21

Secondary DAG registers 7-8

Secondary DAG registers enable  
(SEC\_DAG) bit 2-13, 7-8,  
7-10, 7-62, 8-69

Secondary registers enable  
(SEC\_REG) bit 2-11, 7-10,  
8-69

Set bit (SETBIT) instruction 3-18

Set Interrupt 8-62

Set interrupt (SETINT) instruction  
9-60

Shifter block exponent (SB) register  
2-4, 5-2, 5-3, 5-23, 5-24, 7-3,  
7-28, 9-14

Shifter data register 2-3

Shifter exponent (SE) register 2-4,  
5-3, 5-4, 5-6, 5-10, 5-14, 5-20,  
5-21, 5-26, 6-11, 6-15, 6-19,  
7-3, 7-28, 9-14

Shifter function (SF) codes 9-15

Shifter input (SI) register 3-5, 3-8,  
3-12, 3-15, 3-20, 3-23, 3-26,  
3-29, 3-32, 4-9, 4-12, 4-15,  
5-3, 5-6, 5-8, 5-10, 5-12, 5-15,  
5-17, 5-20, 5-23, 6-11, 6-15,  
6-19, 7-3, 7-24, 7-28, 7-49,  
7-52, 7-59, 7-61, 9-15, 9-19

Shifter instructions 2-21

Shifter options 5-3, 5-4, 5-7, 5-9,  
5-11, 5-13, 5-15, 5-18, 5-20,  
5-21, 5-24

Shifter overflow (SV) bit 2-6, 3-6,  
3-9, 3-13, 3-16, 3-19, 3-21,  
3-24, 3-27, 3-30, 3-33, 3-37,  
3-45, 4-6, 4-7, 4-9, 4-12, 4-15,  
4-18, 4-20, 4-21, 4-22, 4-25,  
5-5, 5-7, 5-9, 5-11, 5-13, 5-15,  
5-18, 5-21, 5-24

Shifter registers 5-2

Shifter Result (SR) register 4-24

Shifter result (SR0-2) registers 2-3,  
3-5, 3-6, 3-8, 3-9, 3-12, 3-15,  
3-16, 3-18, 3-20, 3-21, 3-23,  
3-26, 3-29, 3-32, 3-36, 3-44,  
4-2, 4-3, 4-5, 4-6, 4-9, 4-12,  
4-14, 4-15, 4-17, 4-19, 4-21,  
4-22, 4-24, 5-2, 5-3, 5-6, 5-8,  
5-10, 5-12, 5-14, 5-15, 5-17,  
5-20, 5-23, 6-11, 6-15, 6-16,  
6-19, 7-3, 7-22, 7-24, 7-25,  
7-28, 7-49, 7-52, 7-59, 7-61,  
9-15, 9-19, 9-20

Shifter signed (SS) bit 2-6, 3-6, 3-9,  
3-13, 3-16, 3-19, 3-21, 3-24,  
3-27, 3-30, 3-33, 3-37, 3-45,  
4-9, 4-12, 4-15, 4-18, 4-20,  
4-21, 4-25, 5-5, 5-7, 5-9, 5-11,  
5-13, 5-15, 5-18, 5-21, 5-24

Shifter status bits 2-6

Shifter status flags 5-5

Shifter x-input (XOP) 1-9

Shifter y-input (YOP) 1-9

- Signed division 3-37, 3-39
- Signed input (S) multiplier option 4-3
- Signed inputs (SS) multiplier option 4-4
- Signed, ALU (AS) bit 2-5
- Signed, shifter (SS) bit 2-6
- Signed/unsigned inputs (SU) multiplier option 4-4
- Single-step (SS) RTI option 8-48
- Single-step interrupt mask (SSTEP) bit 2-10
- Software condition (SWCOND) bits 2-7
- Software condition true (SWCOND) condition 2-16, 8-4, 9-12
- Spill-fill mode 8-13
- Stack address, PC (STACKA) register 2-4, 7-3, 7-28, 8-9, 8-11, 8-55, 8-57, 8-58, 9-15
- Stack interrupt 8-13
- Stack interrupt mask (STACK) bit 2-10
- Stack overflow status (STKOVERFLOW) bit 2-15, 8-12, 8-22, 8-31, 8-38, 8-43, 8-49, 8-53, 8-56, 8-63, 8-65
- Stack pointer, PC (STACKP) register 2-4, 7-3, 7-28, 8-9, 8-11, 8-55, 8-57, 8-58, 9-15
- Stack status flags 8-12
- Stack, PC interrupt enable (PCSTKE) bit 2-8
- Stacks 8-7
- Status flags 3-4, 3-6, 3-9, 3-13, 3-16, 3-19, 3-20, 3-21, 3-24, 3-27, 3-30, 3-33, 3-37, 3-44, 3-45, 4-7, 4-9, 4-12, 4-15, 4-18, 4-20, 4-21, 4-24, 4-25, 5-5, 5-7, 5-9, 5-11, 5-13, 5-15, 5-18, 5-21, 5-24, 6-1, 6-4, 6-8, 6-11, 6-15, 6-19, 8-12, 8-22, 8-28, 8-31, 8-35, 8-38, 8-40, 8-43, 8-46, 8-49, 8-53, 8-56, 8-61, 8-63, 8-65, 8-66, 8-67, 8-70
- Status stack empty status (STSSTKEMPTY) bit 2-15, 8-22, 8-31, 8-38, 8-43, 8-49, 8-53, 8-56, 8-63, 8-65
- Status stack operation 8-8
- Status, arithmetic (ASTAT) register 2-4
- Status, mode (MSTAT) register 2-4
- Status/Control registers (G3reg) 7-4
- Subtract X-Y/Subtract X-Y with Borrow 3-8
- Subtract Y-X/Subtract Y-X with Borrow 3-12
- Support (technical or customer) 1-6
- System control (SYSCTL) register 7-61
- System control register read/write 7-61
- System status (SSTAT) register 2-4, 2-14, 7-3, 7-10, 7-23, 7-28, 7-31, 7-35, 7-38, 7-42, 8-10,

# INDEX

- 8-12, 8-16, 8-22, 8-31, 8-38,  
8-43, 8-49, 8-53, 8-56, 8-63,  
8-65, 9-14
- Latency 7-10, 8-10, 8-12, 8-56
- T**
- Technical support 1-6
- Telex for information 1-6
- Termination (TERM) condition  
1-9
- Termination conditions, loops 8-22
- Test bit (TSTBIT) instruction 3-18
- TESTBIT, SETBIT, CLRBIT,  
TGLBIT 3-18
- Timer enable (TIMER) bit 2-13,  
7-10, 8-69, 9-7, 9-40
- Toggle bit (TGLBIT) instruction  
3-18
- TRUE condition 2-16, 9-7, 9-11,  
9-12
- U**
- Unsigned division 3-37, 3-40
- Unsigned input (U) multiplier  
option 4-3
- Unsigned inputs (UU) multiplier  
option 4-4
- Unsigned/signed inputs (US)  
multiplier option 4-4
- V**
- VisualDSP 1-3
- W**
- Web site 1-6
- X**
- X-input operand (XOP) 1-9, 2-3,  
3-2, 3-6, 3-9, 3-12, 3-16, 3-18,  
3-21, 3-23, 3-26, 3-36, 3-44,  
4-2, 4-9, 4-12, 4-15, 4-24
- XOP and YOP Codes 9-19
- Y**
- Y operand (YOP) 2-3
- Y-input operand (YOP) 1-9, 3-2,  
3-6, 3-9, 3-12, 3-16, 3-21,  
3-23, 3-29, 3-32, 3-36, 3-44,  
4-2, 4-9, 4-12, 4-15, 4-24
- Z**
- Zero, ALU (AZ) bit 2-5