

CONTENTS

Chapter 1 THE SF ARCHITECTURE

1.1	GENERAL DESCRIPTION	1-1
1.2	MANUAL ORGANIZATION	1-2
1.3	SF ARCHITECTURE CORE	1-3
1.3.1	Instruction Processing in the SF Architecture Core	1-3
1.3.2	Debugging and Self-test Features	1-7
1.3.3	Support for High Level Languages	1-7
1.4	EMBEDDED CONTROL APPLICATIONS	1-8
1.5	MULTIPROCESSING AND PARALLEL PROCESSING	1-9

Chapter 2 PROGRAMMING MODEL

2.1	PROGRAM MODES	2-1
2.2	DATA TYPES	2-2
2.2.1	Integer Data	2-2
2.2.2	Floating-Point Data	2-2
2.3	INSTRUCTION SET	2-4
2.4	REGISTER SET	2-6
2.4.1	General Purpose Registers	2-8
2.4.2	Floating-Point Data Registers	2-8
2.4.3	Dedicated Address Registers	2-8
2.4.4	Processor Status Register	2-9
2.4.5	Configuration Register	2-11
2.4.6	Floating-Point Status Register	2-12
2.4.7	Floating-Point Trapped Result Registers	2-14
2.5	DATA ORGANIZATION	2-15
2.5.1	Bit and Byte Order for Data References	2-15
2.5.2	Stacks	2-16
2.5.3	Memory-Mapped I/O	2-17
2.6	ADDRESSING MODES	2-18

Chapter 3 EXCEPTIONS

3.1	INTRODUCTION	3-1
3.1.1	Interrupts	3-2

3.1.2	Traps	3-3
3.2	RESET	3-5
3.3	EXCEPTION PROCESSING	
3.3.1	Instruction Endings	3-6
3.3.2	Acknowledging an Exception	3-6
3.3.3	Exception Service Procedures	3-6
3.3.4	Returning From Exception Service Procedures	3-8
3.3.5	Priority Among Exceptions	3-8
3.3.6	Floating-Point Exceptional Conditions	3-8
3.3.7	Operation with Special Floating-Point Operands	3-9
		3-11
		3-15

Chapter 4 ADDITIONAL TOPICS

4.1	DEBUGGING SUPPORT	
4.1.1	Instruction Tracing	4-1
4.1.2	Controlling the Pipeline	4-1
4.1.3	The Breakpoint Instruction	4-2
4.1.4	PC Match	4-2
4.1.5	Address-Compare	4-3
4.1.6	Event Counting	4-3
4.1.7	Checking for Debug Conditions	4-4
4.1.8	Controlling the Debug Options	4-6
4.2	ON-CHIP CACHES	4-7
4.2.1	Instruction Cache	4-9
4.2.2	Data Cache	4-9
4.3	INSTRUCTION EXECUTION ORDER	4-10
4.3.1	Instruction Pipeline and Parallel Execution	4-14
4.3.2	Serializing Operations	4-15
		4-17

Chapter 5 INSTRUCTION SET

5.1	INTRODUCTION	5-1
5.2	INSTRUCTION DEFINITIONS	5-1
	ABSf, ABSUf (Absolute Value Floating)	5-3
	ADDf, ADDUf (Add Floating)	5-4
	ADDi, ADDUi (Add Integer)	5-5
	ADDCi (Add Integer with Carry)	5-6
	ANDi (Bitwise Logical AND)	5-7
	ASHUi (Arithmetic Shift)	5-8
	Bcond (Conditional Branch)	5-10
	BAL (Branch and Link)	5-11
	BICi (Bit Clear)	5-12
	BR (Unconditional Branch)	5-14
	CINV (Cache Invalidate)	5-15

CMPf, CMPUF (Compare Floating).....	5-16
CMPi (Compare Integer).....	5-17
CMPCHD (Compare Characters in Double-Word)	5-19
COMi (Complement Integer)	5-20
DIVf, DIVUF (Divide Floating).....	5-21
EOI (End-of-Interrupt).....	5-22
EXCP (Exception)	5-23
FLOORf, FLOORUF (Floor Floating to Integer)	5-24
JUMP (Jump).....	5-25
LOADf, LOADUF (Load Floating).....	5-26
LOADi (Load Integer).....	5-27
LOADNF, LOADNUF (Load Non-Cacheable Floating).....	5-28
LOADNi (Load Non-Cacheable Integer)	5-30
LPR (Load Processor Register)	5-31
LSHi (Logical Shift Integer).....	5-32
MBITI (Move Bit Interlocked)	5-33
MHHD (Multiply High Word by High Word).....	5-35
MLHD (Multiply Low Word by High Word).....	5-36
MLLD (Multiply Low Word by Low Word)	5-37
MOVf, MOVUF (Move Floating).....	5-38
MOVi (Move Integer).....	5-39
MOVDL, MOVUDL (Convert Integer to Long).....	5-40
MOVFL, MOVUFL (Convert Float to Long)	5-41
MOVLF, MOVULF (Convert Long to Float)	5-42
MOVXiD (Move with Sign Extension).....	5-43
MOVZiD (Move with Zero Extension)	5-44
MULf, MULUF (Multiply Floating).....	5-45
MULi, MULUi (Multiply Integer)	5-46
NEGf, NEGUF (Negate Floating).....	5-47
NEGi (Negate Integer)	5-48
NOP (No Operation)	5-49
ORi (Bitwise Logical OR)	5-50
RETX (Return From Exception)	5-51
ROUNDf, ROUNDUF (Round Floating to Integer)	5-52
Scondi (Save Condition as Boolean)	5-53
SPR (Store Processor Register).....	5-55
STORf, STORUF (Store Floating)	5-57
STORi (Store Integer)	5-58
SUBf, SUBUF (Subtract Floating)	5-59
SUBi, SUBUi (Subtract Integer)	5-60
SUBCi (Subtract Integer with Carry)	5-62
TBIT (Test Bit)	5-63
TRDF, TRUDF (Transfer Integer to Floating).....	5-64
TRFD, TRUFD (Transfer Floating to Integer).....	5-65
TRUNCf, TRUNCUF (Truncate Floating to Integer)	5-66
UCMPf, UCMPUF (Unordered Compare Floating)	5-67
WAIT (Wait for Interrupt)	5-68
XORi (Bitwise Logical Exclusive OR).....	5-69

Appendix A INSTRUCTION EXECUTION TIMING

A.1	INSTRUCTION PAIRING	A-2
A.1.1	The Pairing Algorithm	A-2
A.1.2	Dependencies Between Paired Instructions	A-3
A.2	INTEGER PIPELINE ORGANIZATION	A-3
A.2.1	Branching Delays	A-4
A.2.2	Data Delays	A-5
A.2.3	Memory Load Delays	A-6
A.2.4	Operand Size	A-6
A.2.5	Read-After-Write Contentions	A-7
A.3	FLOATING-POINT PIPELINE ORGANIZATION	A-8
A.3.1	Floating-Point Bypass Registers and Data Delays	A-9
A.3.2	Floating-Point Resource Contentions	A-10
A.3.3	CMP[U]f and UCMP[U]f Delay	A-13
A.3.4	Floating-Point No-Trap and Normal-Completion Conditions	A-14
A.4	CACHE DELAYS	A-15
A.4.1	Data Cache Misses and Scheduled Loads	A-16
A.4.2	Instruction Cache Misses	A-16
A.5	SPECIAL FUNCTION INSTRUCTIONS	A-17
A.6	EXCEPTIONS	A-18
A.7	INSTRUCTION EXECUTION TIMING TABLES	A-20
		A-21

Appendix B PERIPHERALS

B.1	NS32SF641 PERIPHERALS	B-1
B.2	INTERRUPT CONTROL UNIT (ICU)	B-2
B.2.1	Interrupt Control Unit (ICU) Registers	B-2
B.2.2	ICU Operation	B-3
B.3	TIMER	B-5
B.3.1	Timer Registers	B-5
B.3.2	Timer Operation	B-7
B.4	DIRECT MEMORY ACCESS (DMA) CONTROLLER	B-10
B.4.1	The DMA Controller Registers	B-10
B.4.2	DMA Operation	B-15

Appendix C INSTRUCTION SET ENCODING

C.1	INTRODUCTION	C-1
C.2	SF INSTRUCTION SET SUMMARY	C-1

Appendix D CALLING CONVENTION

D.1	INTRODUCTION	D-1
D.2	CALLING CONVENTION ELEMENTS	D-2
D.2.1	Parameter Passing Procedure	D-2
D.2.2	The Argument Stack	D-3
D.2.3	Register Usage Conventions	D-4
D.2.4	Return Address	D-4
D.2.5	Returned Value	D-4
D.3	EXAMPLES	D-5

INDEX