Topics

Tables

Notes

5 Instruction Set Summary

This chapter summarizes the MSP430 family instruction set.

5.1 Symbols and Abbreviations

The following table lists the instruction set symbols and abbreviations used throughout the rest of this chapter.

Symbol	Definition	Symbol	Definition
src	The source operand defined by As and S-reg	dst	The destination operand defined by Ad and D-reg
As	The bits representing the addressing mode used for the source	Ad	The bit representing the addressing mode used for the destination
S-reg	The used Working Register for the source src	D-reg	The used Working Register for the destination dst
R0 or PC	Register 0 or Program Counter	R1 or SP	Register 1 or Stack Pointer
R2 or SR/CG1	Register 2 or Status Register/Constant Generator 1	R3 or CG2	Register 3 or Constant Generator 2
R4 to R15	Working Register, general purpose	Rn	Working Register with n=4-15, general purpose
#	Immediate Data	@	Register indirect addressing
&	Absolute address	-->	Data transfer direction
label	16-bit label	TOS	Top of Stack
C	Carry Bit	N	Negative Bit
V	Overflow Bit	Z	Zero Bit
.B	The suffix .B at the instruction memonic will result in a byte operation	.W	The suffix .W or no suffix at the instruction memonic will result in a word operation
MSB	Most significant Bit	LSB	Least significant Bit

Table 5.1: Symbols and Abbreviations used in the Instruction Set Summary

5.2 Addressing Modes

All seven addressing modes for the source operand and all four addressing modes for the destination operand can address the complete address space. The bit numbers show the contents of the As resp. Ad mode bits.

As	Ad	Addressing Mode	Syntax	Description
00	0	Register Mode	Rn	Register contents are operand
01	1	Indexed Mode	X(Rn)	(Rn + X) points to the operand. X is stored in the next word
01	1	Symbolic Mode	ADDR	(PC + X) points to the operand. X is stored in the next word. Indexed Mode X(PC) is used
01	1	Absolute Mode	&ADDR	The word following the instruction contains the absolute address.
10	-	Indirect Register Mode	@Rn	Rn is used as a pointer to the operand
11	-	Indirect Autoincrement	@Rn+	Rn is used as a pointer to the operand. Rn is incremented afterwards
11	-	Immediate Mode	#N	The word following the instruction contains the immediate constant N. Indirect Autoincrement Mode @PC+ is used

Table 5.2: Addressing Modes

Note: Addressing Modes

The addressing modes using the PC as the working register use the normal effects of the addressing modes. The special addressing modes are caused by the pointing of the PC to the ROM word following the currently executed instruction.

5.3 Instruction Set Summary

				Status Bits			
				V	N	Z	C
*	ADC(.B)	dst	dst + C → dst	x	x	x	x
	ADD(.B)	src,dst	src + dst → dst	x	x	x	x
	ADDC(.B)	src,dst	src + dst + C → dst	x	x	x	x
	AND(.B)	src,dst	src .and. dst → dst	0	x	x	x
	BIC(.B)	src,dst	.not.src .and. dst → dst	-	-	-	-
	BIS(.B)	src,dst	src .or. dst → dst	-	-	-	-
	BIT(.B)	src,dst	src .and. dst	0	x	x	x
*	BR	dst	Branch to	-	-	-	-
	CALL	dst	PC+2 → stack, dst → PC	-	-	-	-
*	CLR(.B)	dst	Clear destination	-	-	-	-
*	CLRC		Clear carry bit	-	-	-	0
*	CLRN		Clear negative bit	-	0	-	-
*	CLRZ		Clear zero bit	-	-	0	-
	CMP(.B)	src,dst	dst - src	x	x	x	x
*	DADC(.B)	dst	dst + C → dst (decimal)	x	x	x	x
	DADD(.B)	src,dst	src + dst + C → dst (decimal)	x	x	x	x
*	DEC(.B)	dst	dst - 1 → dst	x	x	x	x
*	DECD(.B)	dst	dst - 2 → dst	x	x	x	x
*	DINT		Disable interrupt	-	-	-	-
*	EINT		Enable interrupt	-	-	-	-
*	INC(.B)	dst	Increment destination, dst +1 → dst	x	x	x	x
*	INCD(.B)	dst	Double-Increment destination, dst+2→dst	x	x	x	x
*	INV(.B)	dst	Invert destination	x	x	x	x
	JC/JHS	Label	Jump to Label if Carry-bit is set	-	-	-	-
	JEQ/JZ	Label	Jump to Label if Zero-bit is set	-	-	-	-
	JGE	Label	Jump to Label if (N .XOR. V) = 0	-	-	-	-
	JL	Label	Jump to Label if (N .XOR. V) = 1	-	-	-	-
	JMP	Label	Jump to Label unconditionally	-	-	-	-
	JN	Label	Jump to Label if Negative-bit is set	-	-	-	-

Legend:
0 Status bit always cleared
1 Status bit always set
x Status bit cleared or set on results
- Status bit not affected
* Emulated Instructions

Table 5.3: MPS430 Family Instruction Set Summary

				Status Bits			
				V	N	Z	C
	JNC/JLO	Label	Jump to Label if Carry-bit is reset	-	-	-	-
	JNE/JNZ	Label	Jump to Label if Zero-bit is reset	-	-	-	-
	MOV(.B)	src,dst	src → dst	-	-	-	-
*	NOP		No operation	-	-	-	-
*	POP(.B)	dst	Item from stack, SP+2 → SP	-	-	-	-
	PUSH(.B)	src	SP - 2 → SP, src → @SP	-	-	-	-
	RETI		Return from interrupt TOS → SR, SP + 2 → SP TOS → PC, SP + 2 → SZP	x	x	x	x
*	RET		Return from subroutine TOS → PC, SP + 2 → SP	-	-	-	-
*	RLA(.B)	dst	Rotate left arithmetically	x	x	x	x
*	RLC(.B)	dst	Rotate left through carry	x	x	x	x
	RRA(.B)	dst	MSB → MSB ....LSB → C	0	x	x	x
	RRC(.B)	dst	C → MSB .....LSB → C	x	x	x	x
*	SBC(.B)	dst	Subtract carry from destination	x	x	x	x
*	SETC		Set carry bit	-	-	-	1
*	SETN		Set negative bit	-	1	-	-
*	SETZ		Set zero bit	-	-	1	-
	SUB(.B)	src,dst	dst + .not.src + 1 → dst	x	x	x	x
	SUBC(.B)	src,dst	dst + .not.src + C → dst	x	x	x	x
	SWPB	dst	swap bytes	-	-	-	-
	SXT	dst	Bit7 → Bit8 ..... Bit15	0	x	x	x
*	TST(.B)	dst	Test destination	x	x	x	x
	XOR(.B)	src,dst	src .xor. dst → dst	x	x	x	x

Legend:
0 The Status Bit is cleared
1 The Status Bit is set
x The Status Bit is affected
- The Status Bit is not affected
* Emulated Instructions

Table 5.3: MPS430 Family Instruction Set Summary (Concluded)

Note: Emulated Instructions

All marked instructions (*) are emulated instructions. The emulated instructions use core instructions combined with the architecture and implementation of the CPU for higher code efficiency and faster execution.

5.4 Clock cycles, Length of Instruction

The operating speed of the CPU is independent from individual instructions. It depends on the instruction format and the addressing modes. The number of clock cycles refer to the internal oscillator frequency.

5.4.1 Format I Instructions

Address Mode		#of cycles	Length of instruction	Example
As	Ad			
00, Rn	0, Rm	1	1	MOV R5,R8
	0,PC	2	1	BR R9
00, Rn	1, x(Rm)	4	2	ADD R5,3(R6)
	1, EDE		2	XOR R8,EDE
	1, &EDE		2	MOV R5,&EDE
01, x(Rn)	0, Rm	3	2	MOV 2(R5),R7
01, EDE			2	AND EDE,R6
01, &EDE				MOV &EDE,R8
01, x(Rn)	1, x(Rm)	6	3	ADD 3(R4),6(R9)
01, EDE	1, TONI		3	CMP EDE,TONI
01, &EDE	1, &TONI		3	MOV 2(R5),&TONI
				ADD EDE,&TONI
10, @Rn	0, Rm	2	1	AND @R4,R5
10, @Rn	1, x(Rm)	5	2	XOR @R5,8(R6)
	1, EDE		2	MOV @R5,EDE
	1, &EDE		2	XOR @R5,&EDE
11, @Rn+	0, Rm	2	1	ADD @R5+,R6
	0, PC	3	1	BR @R9+
11, #N	0, Rm	2	2	MOV #20,R9
	0, PC	2	2	BR #2AEh
11, @Rn+	1, x(Rm)	5	2	MOV @R9+,2(R4)
11, #N	1, EDE		3	ADD #33,EDE
11, @Rn+	1, &EDE		2	MOV @R9+,&EDE
11, #N			3	ADD #33,&EDE

Table 5.4: Format I Instructions

Note: Cycle Time of the DADD Instruction

The DADD instruction needs 1 extra cycle.

5.4.2 Format II Instructions

Address Mode $A_{(s/d)}$	#of cycles		Length of instruction [words]	Example
	RRC RRA SWPB SXT	PUSH/ CALL		
00, Rn	1	3/4	1	SWPB R5
01, x(Rn) 01, EDE	4 4	5 5	2 2	CALL 2(R7) PUSH EDE
10, @Rn	3	4	1	RRC @R9
11, @Rn+ see Note 11, #N	3 3	4/5 4/5	1 2	SWPB @R10+ CALL #81h

Table 5.5: Format II Instructions

Note: Immediate mode in destination field

Instructions should not use immediate mode in the destination field. This would result in unpredictable program operation.

5.4.3 Format III Instructions

Jxx - instructions need all the same #-of-cycles independent of a successfull Jump or not.

Clock Cycle: 2 Cycle.
Length of Instruction: 1 word.

5.4.4 Miscellanous Instructions or Operators

RETI	Clock Cycle:	5 Cycle.
	Length of instruction:	1 word.
Interrupt	Clock Cycle:	6 Cycle.