Registers

Register	Description
R0	16 bit, General Purpose
R1	16 bit, General Purpose
R2	16 bit, General Purpose
R3	16 bit, General Purpose
R4	16 bit, General Purpose
R5	16 bit, General Purpose
R6	16 bit, General Purpose
R7	16 bit, General Purpose
R8	16 bit, General Purpose
R9	16 bit, General Purpose
R10	16 bit, General Purpose
R11	16 bit, General Purpose
R12	16 bit, General Purpose
AP	16 bit, Argument Pointer, General Purpose
SP	16 bit, Stack Pointer
PC	16 bit, Program Pointer
SR I V S C Z	8 bit, Status Register
All general purpose registers are 16 bit long. 8 bit instructions use the lower byte leaving the upper one unaffected and updating flags appropriately ignoring the upper byte.	

Instructions

	Encoding				Machine Name	Assembly Instruction	Description
No operation							
T0	0000	0000	0000	0000	nop	nop	No operation
Constant Immediate Arithmetic							
T1	0001	kkkk	kkkk	Rd	mov16kr	mov.w K, Rd	Copy sign extended K into Rd.w
	0010	kkkk	kkkk	Rd	cmp8kr	cmp.b K, Rd	Compare Rd.b with K and update flags
	0011	kkkk	kkkk	Rd	cmp16kr	cmp.w K, Rd	Compare Rd.w with sign extended K and update flags
	0100	kkkk	kkkk	Rd	add8kr	add.b K, Rd	Add K to Rd.b and update flags
	0101	kkkk	kkkk	Rd	add16kr	add.w K, Rd	Add sign extended K to Rd.b and update flags
	0110	kkkk	kkkk	Rd	sub8kr	sub.b K, Rd	Subtract K from Rd.b and update flags
	0111	kkkk	kkkk	Rd	sub16kr	sub.w K, Rd	Subtract sign extended K from Rd.w and update flags
PC Relative Branch							
	1000 00	aa	aaaa	aaaa	blt	blt label	Branch PC relative if S != V (signed less than)
	1000 01	aa	aaaa	aaaa	ble	ble label	Branch PC relative if (S != V) || Z (signed less or equal)

	Encoding				Machine Name	Assembly Instruction	Description
T2	1000 10	aa	aaaa	aaaa	bge	bge label	Branch PC relative if S == V (signed greater than or equal)
	1000 11	aa	aaaa	aaaa	bgt	bgt label	Branch PC relative if (S == V) && !Z (signed greater than)
	1001 00	aa	aaaa	aaaa	bult	bult label blo label bnc label	Branch PC relative if !C (unsigned lower) (not carry)
	1001 01	aa	aaaa	aaaa	bule	bule label bls label	Branch PC relative if !C || Z (unsigned lower or same)
	1001 10	aa	aaaa	aaaa	buge	buge label bhs label bc label	Branch PC relative if C (unsigned higher or same) (carry)
	1001 11	aa	aaaa	aaaa	bugt	bugt label bhi label	Branch PC relative if C && !Z (unsigned higher)
	1010 00	aa	aaaa	aaaa	beq	beq label bz label	Branch PC relative if Z (equal) (zero)
	1010 01	aa	aaaa	aaaa	bne	bne label bnz label	Branch PC relative if !Z (not equal) (not zero)
	1010 10	aa	aaaa	aaaa	bs	bs label	Branch PC relative if N (negative)
	1010 11	aa	aaaa	aaaa	bns	bv label	Branch PC relative if V (overflow)
Long PC Relative Jump and Call							
T5	1011	aaaa	aaaa	aaaa	jmp	jmp label	Unconditional Branch PC relative
	1110	aaaa	aaaa	aaaa	call	call label	Call Subroutine PC relative
Two Operand Arithmethic							
T3	1100	0000	Rs	Rd	mov8rr	mov.b Rs, Rd	Copy Rs.b to Rd.b
	1100	0001	Rs	Rd	mov16rr	mov.w Rs, Rd	Copy Rs.w to Rd.w
	1100	0010	Rs	Rd	cmp8rr	cmp.b Rs, Rd	Compare Rs.b with Rd.b and update flags
	1100	0011	Rs	Rd	cmp16rr	cmp.w Rs, Rd	Compare Rs.w with Rd.w and update flags
	1100	0100	Rs	Rd	add8rr	add.b Rs, Rd	Add Rs.b to Rd.b and update flags
	1100	0101	Rs	Rd	add16rr	add.w Rs, Rd	Add Rs.w to Rd.w and update flags
	1100	0110	Rs	Rd	adc8rr	addc.b Rs, Rd	Add Rs.b+C to Rd.b and update flags
	1100	0111	Rs	Rd	adc16rr	addc.w Rs, Rd	Add Rs.w+C to Rd.w and update flags
	1100	1000	Rs	Rd	sub8rr	sub.b Rs, Rd	Subtract Rs.b from Rd.b and update flags
	1100	1001	Rs	Rd	sub16rr	sub.w Rs, Rd	Subtract Rs.w from Rd.w and update flags
	1100	1010	Rs	Rd	sbc8rr	subc.b Rs, Rd	Subtract Rs.b+!C from Rd.b and update flags
	1100	1011	Rs	Rd	sbc16rr	subc.w Rs, Rd	Subtract Rs.w+!C from Rd.w and update flags
	1100	1100	Rs	Rd	-	-	Reserved
	1100	1101	Rs	Rd	-	-	Reserved
	1100	1110	Rs	Rd	-	-	Reserved
	1100	1111	Rs	Rd	-	-	Reserved
Two Operand Logical							
	1101	0000	Rs	Rd	or8rr	or.b Rs, Rd	Logic OR with Rs.b to Rd.b and update flags
	1101	0001	Rs	Rd	or16rr	or.w Rs, Rd	Logic OR with Rs.w to Rd.w and update flags
	1101	0010	Rs	Rd	and8rr	and.b Rs, Rd	Logic AND with Rs.b to Rd.b and update flags

	Encoding				Machine Name	Assembly Instruction	Description
T3	1101	0011	Rs	Rd	and16rr	and.w Rs, Rd	Logic AND with Rs.w into Rd.w and update flags
	1101	0100	Rs	Rd	xor8rr	xor.b Rs, Rd	Logic XOR with Rs.b into Rd.b and update flags
	1101	0101	Rs	Rd	xor16rr	xor.w Rs, Rd	Logic XOR with Rs.w into Rd.w and update flags
	1101	0110	Rs	Rd	-	-	Reserved
	1101	0111	Rs	Rd	-	-	Reserved
Indirect and Indexed Memory Addressing Load and Store							
T3	1101	1000	Rs	Rd	mov8ir	mov.b (Rs), Rd	Copy the byte contents of memory address Rs.w into Rd.b
	1101	1001	Rs	Rd	mov16ir	mov.w (Rs), Rd	Copy the word contents of word aligned memory address Rs.w into Rd.w
	1101	1010	Rs	Rd	mov8ri	mov.b Rs, (Rd)	Store Rs.b in memory address Rd.w
	1101	1011	Rs	Rd	mov16ri	mov.w Rs, (Rd)	Store Rs.w in word aligned memory address Rd.w
	1101	1100	Rs	Rd	mov8mr	mov.b K(Rs), Rd	Copy the byte contents of memory address Rs.w+K into Rd.b
	kkkk	kkkk	kkkk	kkkk			
	1101	1101	Rs	Rd	mov16mr	mov.w K(Rs), Rd	Copy the word contents of word aligned memory address Rs.w+K into Rd.w
	kkkk kkkk kkkk kkkk						
	1101	1110	Rs	Rd	mov8mr	mov.b Rs, K(Rd)	Store Rs.b in memory address Rd.w+K
	kkkk	kkkk	kkkk	kkkk			
	1101	1111	Rs	Rd	mov16mr	mov.w Rs, K(Rd)	Store Rs.w in word aligned memory address Rd.w+K
	kkkk kkkk kkkk kkkk						
Immediate and Direct Absolute Memory Addressing Load and Store							
T4	1111	0000	0000	Rd	-	-	Reserved
	xxxx xxxx xxxx xxxx						
	1111	0001	0000	Rd	mov16Kr	mov.w K, Rd	Copy K into Rd.w
	kkkk kkkk kkkk kkkk						
	1111	0010	0000	Rd	mov8ar	mov.b @A, Rd	Copy the byte contents of memory address A into Rd.b
	aaaa aaaa aaaa aaaa						
	1111	0011	0000	Rd	mov16ar	mov.w @A, Rd	Copy the word contents of word aligned memory address A into Rd.b
	aaaa aaaa aaaa aaaa						
	1111	0010	0000	Rd	mov8ra	mov.b Rs, @A	Store Rs.b byte in memory address A
	aaaa aaaa aaaa aaaa						
	1111	0011	0000	Rd	mov16ra	mov.w Rs, @A	Store Rs.w in word aligned memory address A
	aaaa aaaa aaaa aaaa						
Single Operand Instructions							
	1111	0000	0001	Rd	lsr8r	lsr.b Rd	1 bit shift right of Rd.b and update flags
	1111	0001	0001	Rd	lsr16r	lsr.w Rd	1 bit shift right of Rd.w and update flags
	1111	0010	0001	Rd	lsl8r	lsl.b	1 bit shift left of Rd.b and update flags
	1111	0011	0001	Rd	lsl16r	lsl.w	1 bit shift left of Rd.w and update flags
	1111	0100	0001	Rd	asr8r	asr.b	1 bit signed shift right of Rd.b and update flags
	1111	0101	0001	Rd	asr16r	asr.w	1 bit signed shift right of Rd.w and update flags

	Encoding				Machine Name	Assembly Instruction	Description
T6	1111	0110	0001	Rd	zex16r	zex.bw Rd	Zero extend Rd.b into Rd.w
	1111	0111	0001	Rd	sex16r	sex.bw Rd	Sign extend Rd.b into Rd.w
	1111	1000	0001	Rd	swap16r	swap.bw Rd	Exchange upper and lower bytes of Rd.w
	1111	1001	0001	Rd	-	-	Reserved
	1111	1010	0001	Rd	-	-	Reserved
	1111	1011	0001	Rd	-	-	Reserved
	1111	1100	0001	Rd	-	-	Reserved
	1111	1101	0001	Rd	-	-	Reserved
	1111	1110	0001	Rd	-	-	Reserved
	1111	1111	0001	Rd	-	-	Reserved
Other Single Operand Instructions							
T6	1111	0000	0010	Rd	push16r	push Rd	Push Rd.w onto the stack
	1111	0001	0010	Rd	pop16r	pop Rd	Pop Rd.w from the stack
	1111	0010	0010	Rd	call16i	call *Rd	Call Subroutine pointed to by Rd.w
	1111	0011	0010	Rd	jump16i	jump *Rd	Jump to program memory address Rd.w
	-	-	-	-	-	-	-
Zero Operand Instructions							
T7	1111	0000	0100	0000	ret	ret	Return from subroutine
	1111	0001	0100	0000	reti	reti	Return from interrupt
	1111	0010	0100	0000	dint	dint	Disable interrups
	1111	0011	0100	0000	eint	eint	Enable interrupts
	-	-	-	-	-	-	-