MISC Processor instruction set summary

Tobias Markus

November 6, 2013

1 add, addc:

instruction	opcode
add/addc r_d, r_{s2}, r_{s1}	0001 $r_{d3}r_{d2}r_{d1}r_{d0}$ $r_{s13}r_{s12}r_{s11}r_{s10}$ $r_{s23}r_{s22}r_{s21}r_{s20}$ 000C 1110 0000 0000

Operation:

add: $r_d \leftarrow r_{s2} + r_{s1}$

addc: $r_d \leftarrow r_{s2} + r_{s1} + Carry$

Cycles: 1

2 and

instruction	opcode
and r_d, r_{s2}, r_{s1}	0001 $r_{d3}r_{d2}r_{d1}r_{d0}$ $r_{s13}r_{s12}r_{s11}r_{s10}$ $r_{s23}r_{s22}r_{s21}r_{s20}$ 0100 1100 0000 0000

Operation:

and: $r_d \leftarrow r_{s2} \wedge r_{s1}$

Cycles: 1

3 branch

instruction	opcode
branch b, s, k	0110 0bs_1s_0 $k_{23}k_{22}k_{21}k_{20}$ $k_{19}k_{18}k_{17}k_{16}$ $k_{15}k_{14}k_{13}k_{12}$ $k_{11}k_{10}k_9k_8$ $k_7k_6k_5k_4$ $k_3k_2k_1k_0$

Operation:

branch: $PC \leftarrow k$ if s in sreg equals b.

Cycles: when jump 2, else 1

4 call

instruction	opcode
call k	1110 0000 $k_{23}k_{22}k_{21}k_{20}$ $k_{19}k_{18}k_{17}k_{16}$ $k_{15}k_{14}k_{13}k_{12}$ $k_{11}k_{10}k_9k_8$ $k_7k_6k_5k_4$ $k_3k_2k_1k_0$

Operation:

call: Writes PC to address stored in the stack pointer $r_{15}$and jumps to k.

$r_{15} \leftarrow r_{15} - 1$
Cycles: 2

5 in

instruction	opcode
in r_d,r_s	1001 $r_{d3}r_{d2}r_{d1}r_{d0}$ 0000 $r_{s3}r_{s2}r_{s1}r_{s0}$ 0000 0000 0000 0000

Operation:
in: Reads the databus value in r_d from address stored in r_s.
Cycles: 1

6 jump

instruction	opcode
jump k	1110 1111 $k_{23}k_{22}k_{21}k_{20}$ $k_{19}k_{18}k_{17}k_{16}$ $k_{15}k_{14}k_{13}k_{12}$ $k_{11}k_{10}k_9k_8$ $k_7k_6k_5k_4$ $k_3k_2k_1k_0$

Operation:
jump: $PC \leftarrow k$
Cycles: 2

7 load

instruction	opcode
load r_d, b, k	0101 $r_{d3}r_{d2}r_{d1}r_{d0}$b000 0000 $k_{15}k_{14}k_{13}k_{12}$ $k_{11}k_{10}k_9k_8$ $k_7k_6k_5k_4$ $k_3k_2k_1k_0$

Operation:
load: if b=0 $r_d(15:0) \leftarrow k$, else $r_d(31:16) \leftarrow k$
Cycles: 1

8 nop

instruction	opcode
nop	0000 0000 0000 0000 0000 0000 0000 0000

Operation:
nop: no operation
Cycles: 1

9 out

instruction	opcode
out r_{s1},r_{s2}	0010 0000 $r_{s13}r_{s12}r_{s11}r_{s10}$ $r_{s23}r_{s22}r_{s21}r_{s20}$ 0000 0000 0000 0000

Operation:

out: Writes the content of $r_{s1}$to the address stored in $r_{s2}$to the data bus.

Cycles: 1

10 or

instruction	opcode
or r_d, r_{s2}, r_{s1}	0001 $r_{d3}r_{d2}r_{d1}r_{d0}$ $r_{s13}r_{s12}r_{s11}r_{s10}$ $r_{s23}r_{s22}r_{s21}r_{s20}$ 0011 1100 0000 0000

Operation:

or: $r_d \leftarrow r_{s2} \vee r_{s1}$

Cycles: 1

11 return

instruction	opcode
return	0000 0000 0000 0000 0000 0000 0000 0001

Operation:

return: Jumps to address stored in r_{15}.

$r_{15} = r_{15} + 1$

Cycles: 2

12 sub/subc

instruction	opcode
sub/subc r_d, r_{s2}, r_{s1}	0001 $r_{d3}r_{d2}r_{d1}r_{d0}$ $r_{s13}r_{s12}r_{s11}r_{s10}$ $r_{s23}r_{s22}r_{s21}r_{s20}$ 011C 1110 0000 0000

Operation:

sub: $r_d \leftarrow r_{s2} - r_{s1}$

subc: $r_d \leftarrow r_{s2} - r_{s1} - Carry$

Cycles: 1

13 svr

instruction	opcode
svr r_d, r_s	0001 $r_{d3}r_{d2}r_{d1}r_{d0}$ $r_{s3}r_{s2}r_{s1}r_{s0}$ 0000 0010 1110 0000 0000

Operation:

svr: shift r_s on bit to the right and put the result in r_d.

Cycles: 1

14 xor

instruction	opcode
xor r_d, r_{s2}, r_{s1}	0001 $r_{d3}r_{d2}r_{d1}r_{d0}$ $r_{s13}r_{s12}r_{s11}r_{s10}$ $r_{s23}r_{s22}r_{s21}r_{s20}$ 0101 1100 0000 0000

Operation:

xor: $r_d \leftarrow r_{s2} \otimes r_{s1}$

Cycles: 1