ISA for enhanced 552 processor

20-bit Instruction Word

| **Instruction** | **Encoding** | **Sample Instruction** | **Encoding** | **Explanation** | **Other Comments** |
| --- | --- | --- | --- | --- | --- |
| ADD | 00000 dddd ssss tttt 000 | ADD R2, R4, R7 | 0x01238 | R2 🡨 R4 + R7 | All flags updated |
| ADDC | 00001 dddd ssss tttt 000 | ADDC R1, R9, R2 | 0x08C90 | R1 🡨 R9 + R2 + C | All flags updated |
| SUB | 00010 dddd ssss tttt 000 | SUB R8, R3, R4 | 0x141A0 | R8 🡨 R3 + R4 | All flags updated |
| SUBB | 00011 dddd ssss tttt 000 | SUBB R11, R4, R3 | 0x1DA18 | R11 🡨 R4 – R3 - C̅ | All flags updated |
| AND | 00100 dddd ssss tttt 000 | AND R2, R4, R7 | 0x21238 | R2 🡨 R4 & R7 | Z & N flags updated |
| OR | 00101 dddd ssss tttt 000 | OR R1, R9, R2 | 0x28C90 | R1 🡨 R9 | R2 | Z & N flags updated |
| NAND | 00110 dddd ssss tttt 000 | NAND R8, R3, R4 | 0x341A0 | R8 🡨 ~(R3 & R4) | Z & N flags updated |
| XOR | 00111 dddd ssss tttt 000 | XOR R11, R4, R3 | 0x3EA18 | R11 🡨 R4 ^ R3 | Z & N flags updated |
| ADDI | 01000 dddd ssss llll lll | ADDI R12, R9, 0x01 | 0x46481 | R12 🡨 R9 + 1 | Flags cond updated |
| ADDIEQ | 01001 dddd ssss llll lll | ADDIEQ R3, R1, 3 | 0x49883 | R12 🡨 R1 + 3 only if Z = 1 | Flags cond updated |
| ADDINE | 01010 dddd ssss llll lll | ADDINE R4, R13, 63 | 0x526BF | R4 🡨 R13 + 63 only if Z = 0 | Flags cond updated |
| ADDIGT | 01011 dddd ssss llll lll | ADDIGT R7, R6, R10 | 0x5BB00 | R7 🡨 R6 + R10 only if Z = 0, N = 0 | Flags cond updated |
| SUBI | 01100 dddd ssss llll lll | SUBI R4, R5, 7 | 0x62287 | R4 🡨 R5 + 7 | All flags updated |
| ANDI | 01101 dddd ssss llll lll | ANDI R3, R2, 0x0F | 0x6990F | R3 🡨 R2 & 0x0F | Z & N flags updated |
| ORI | 01110 dddd ssss llll lll | ORI R13, R10, 0x70 | 0x76D70 | R13 🡨 R10 | 0x70 | Z & N flags updated |
| XORI | 01111 dddd ssss llll lll | XORI R4, R6, -1 | 0x7A37F | R4 🡨 R6 ^ 0xFFFF | Z & N flags updated |
| SLL | 10000 dddd ssss 0001 111 | SLL R7, R11, 5 | 0x83D85 | R7 🡨 R11 << 5 C 🡨 last bit shifted out | Z & N flags updated |
| SRL | 10001 dddd ssss 000l lll | SRL R3, R13, 14 | 0x89E8E | R3 🡨 R13 >> 14 C 🡨 last bit shifted out | Z & N flags updated |
| SRA | 10010 dddd ssss 000l lll | SRA R7, R2, 2 | 0x93902 | R7 🡨 R2 >>> 2 C 🡨 last bit shifted out | Z & N flags updated |
| B <cc> | 10011 cccl llll llll lll  NEQ EQ GT LT GTE LTE OVFL UNCOND | B OVFL, Lasdf | 0x9EFFD | PC 🡨PC + 12-bit signed immediate |  |
| LW | 10100 dddd aaaa llll lll | LW R5, R3, 0 | 0xA2980 | R5 🡨 mem[R3 + 0] | Z & N flags updated |
| SW | 10101 ssss aaaa llll lll | SW R14, R2, 1 | 0xAF101 | mem[R2 + 1] 🡨 R14 | No flags modified |
| LHB | 10110 dddd 000l llll lll | LHB R2, 0x01 | 0xB9089 | R2[15:8] 🡨 0x01 | No flags modified |
| LLB | 10111 dddd 000l llll lll | LLB R2, 0x89 | 0xB1001 | R2 🡨 sign\_ext(0x89) | No flags modified |
| JAL | 11000 llll llll llll lll | JAL Label | 0xC7FE7 | PC 🡨 PC + 15-bit signed imm, R15 🡨 PC | Use JR R15 to return |
| JR | 11001 0000 aaaa 0000 000 | JR R15 | 0xC8780 | PC 🡨 R15 | Can be other than R15 |
| RTI | 11010 0000 0000 0000 000 | RTI | 0xD0000 | Restores PC, PSW, SP | Return from Interrupt |
| PUSH | 11011 ssss 0000 0000 000 | PUSH R8 | 0xDC000 | mem[SP] 🡨 R8, SP++ | SP Starts at 0xE000 |
| POP | 11100 dddd 0000 0000 000 | POP R13 | 0xE6800 | R13 🡨 mem[SP-1], SP-- |  |
| MOVC | 11101 dddd aaaa llll lll | MOVC R6, R3, 0x03 | 0xEB183 | R6 🡨 instrMem[R3 + 3] | Read constant from instruction memory  N & Z flags updated |
| MULU MULS | 11110 dddd ssss tttt 00x | MULU R2, R4, R5  MULS R4, R7, R11 | 0xF1228  0xF23D9 | {MULH,R2} 🡨 $unsigned(R4\*R5)  {MULH,R4} 🡨 $signed(R7\*R11) | x=0 🡺 unsigned  x=1 🡺 signed  Z & N flags update |
| XMULH | 11111 dddd 0000 0000 000 | XMULH R3 | 0xF9800 | R3 🡨 MULH | Z & N flags update |