**R32V2020 Programmer’s Reference Card (2019-08-10)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Category** | **Name** | **Format** | **Syntax** |
| System | No Operation | NO\_ARGS | **nop** |
| Halt and Catch Fire | **hcf** |
| Arithmetic | Add | BIN\_DEST | **add rd,rs2,rs1** |
| Subtract (rd=rs1-rs2) | BIN\_DEST | **sub rd,rs2,rs1** |
| Multiply | BIN\_DEST | **mul rd,rs2,rs1** |
| Logical | OR | BIN\_DEST | **or rd,rs2,rs1** |
| AND | BIN\_DEST | **and rd,rs2,rs1** |
| XOR | BIN\_DEST | **xor rd,rs2,rs1** |
| Shift | Shift left by 1 | UN\_DEST | **sl1 rd,rs1** |
| Shift left by 8 | UN\_DEST | **sl8 rd,rs8** |
| Shift right by 1 | UN\_DEST | **sr1 rd,rs1** |
| Shift right by 8 | UN\_DEST | **sr8 rd,rs8** |
| Rotate left by 1 | UN\_DEST | **rol1 rd,rs1** |
| Rotate right by 1 | UN\_DEST | **ror1 rd,rs1** |
| Arithmetic Shift right by 1 | UN\_DEST | **asr rd,rs1** |
| Compare | Compare | BIN\_CMP | **cmp rs2,rs1** |
| Swap | Swap Endian | UN\_DEST | **ens rd,rs1** |
| Immediate | Load immediate lower | IMM\_DEST | **lil rd,imm** |
| Load immediate upper | IMM\_DEST | **liu rd,imm** |
| Load immediate extended | IMM\_DEST | **lix rd,imm** |
| Load/Stores Data  [p] post incr | Load Data Byte | R6\_DEST | **ldb[p] rd** |
| Load Data Short | R6\_DEST | **lds[p] rd** |
| Load Data Long | R6\_DEST | **ldl[p] rd** |
| Store Data Byte | UN\_R6\_DEST | **sdb[p] rs1** |
| Store Data Short | UN\_R6\_DEST | **sds[p] rs1** |
| Store Data Long | UN\_R6\_DEST | **sdl[p] rs1** |
| Load/Stores Peripheral  [p] post incr | Load Peripheral Byte | R5\_DEST | **lpb[p] rd** |
| Load Peripheral Short | R5\_DEST | **lps[p] rd** |
| Load Peripheral Long | R5\_DEST | **lpl[p] rd** |
| Store Peripheral Byte | UN\_R5\_DEST | **spb[p] rs1** |
| Store Peripheral Short | UN\_R5\_DEST | **sps[p] rs1** |
| Store Peripheral Long | UN\_R5\_DEST | **spl[p] rs1** |
| Stack | Push to stack | UN\_R4\_DEST | **push rs1** |
| Pull from stack | R5\_DEST | **pull rd** |
| Store to stack | UN\_R4\_DEST | **sss rs1** |
| Load from stack | R5\_DEST | **lss rd** |
| Branches | Branch Always | ADDR | **bra addr** |
| Branch if equal to zero (ALU) | ADDR | **bez addr** |
| Branch if equal to one (ALU) | ADDR | **be1 addr** |
| Branch if not zero (ALU) | ADDR | **bnz addr** |
| Branch if carry clear (ALU) | ADDR | **bcc addr** |
| Branch if carry set (ALU) | ADDR | **bcs addr** |
| Branch if less than (cmp) | ADDR | **blt addr** |
| Branch if greater than (cmp) | ADDR | **bgt addr** |
| Branch if equal (cmp) | ADDR | **beq addr** |
| Branch if not equal (cmp) | ADDR | **bne addr** |
| Branch to subroutine | ADDR | **bsr addr** |

**Instruction Format**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Format** | **D31..D24** | **D23..D20** | **D19..D16** | **D15..D12** | **D11..D00** |
| ADDR | OPCODE | Sign-Extended Offset (24-bits) \* | | | |
| BIN\_CMP | OPCODE | X | rs2 | rs1 | X |
| BIN\_DEST | OPCODE | X | rs2 | rs1 | X |
| IMM\_DEST | OPCODE | rd | Signed-Extended Immed (20-bits) \*\* | | |
| NO\_ARGS | OPCODE | X | X | X | X |
| R4\_DEST | OPCODE | rd | X | (r4) | X |
| R5\_DEST | OPCODE | rd | X | (r5) | X |
| R6\_DEST | OPCODE | rd | X | (r6) | X |
| R7\_DEST | OPCODE | rd | X | (r7) | X |
| UN\_DEST | OPCODE | rd | X | rs1 | X |
| UN\_R4\_DEST | OPCODE | (r4) | X | rs1 | X |
| UN\_R5\_DEST | OPCODE | (r5) | X | rs1 | X |
| UN\_R6\_DEST | OPCODE | (r6) | X | rs1 | X |

\* 24-bit range = -8,388,608 to 8,388,607

\*\* 20-bit range = -524,288 to 524,287

(rN) = register as pointer to address space

**Register Aliases**

|  |  |
| --- | --- |
| r0 = ZERO (0x00000000) | r4 = SAR (Stack Pointer) |
| r1 = ONE (0x00000001) | r5 = PAR (Peripheral Pointer) |
| r2 = MINUS 1 (0xFFFFFFFF) | r6 = DAR (Data Pointer) |
| r3 = Condition Code Register | r7 = PC (Program Counter) |

r8-r15 = General Purpose Registers