Operator	Meaning
+	Result is $src_1 + src_2$
-	Result is $src_1 - src_2$
B-A	Result is $src_2 - src_1$
B	Result is src_2
~B	Result is the bitwise inversion of src_2
AND	Result is bitwise *and* of src_1 and src_2
OR	Result is bitwise *or* of src_1 and src_2
XOR	Result is bitwise *exclusive or* of src_1 and src_2
+carry	Result is $src_1 + src_2$ + carry from previous operation
~AND	Result is bitwise (*not* src_1) *and* src_2
AND~	Result is bitwise (src_1 *and* (*not* src_2)
+IFsign	If the operation two instructions prior to the current operation caused the sign condition then the result is $src_1 + src_2$; otherwise the result is src_2
+4	Result is $src_1 + src_2$ with the first 28 bits set to zero
+8	Result is $src_1 + src_2$ with the first 24 bits set to zero
+16	Result is $src_1 + src_2$ with the first 16 bits set to zero

Figure 24.1 Major ALU operators available on Intel's microengine. The *alu* instruction specifies which operation to perform.