AVR202: 16-Bit Arithmetics

8-Bit AVR MCU with Downloadable Flash

Application Note

Features

- **Easily Expandable to 32-Bit or Any Word Length**
- **Code Density and Speed Matches 16-Bit Controllers**
- **Runable Example Program**

Introduction

This application note lists program examples for arithmetic operation on 16-bit values. A listing of all implementations with key performance specifications is given in Table 1.

Table 1. Performance Figures Summary

Application	Code Size (Words)	Execution Time (Cycles)
Add two 16-bit register variables	2	2
Add 16-bit immediate to 16-bit register variable	2	2
Subtract two 16-bit register variables	2	2
Subtract 16-bit immediate from 16-bit reg. variable	2	2
Compare two 16-bit register variables	2	2
Compare 16-bit immediate to 16-bit reg. variable	3	3
Negate a 16-bit register variable	4	4

16 + 16-Bit Register Addition

This operation is done as follows:

1. Add low bytes.
2. Add with carry high bytes.

By adding more Add with Carry instructions, numbers of *n*-byte width can be added using *n* instructions.

16-Bit Register + 16-Bit Immediate Addition

As the AVR has no add immediate or add immediate with carry, the subtract immediate and subtract immediate with carry instructions are used. The operation is done as follows:

1. Subtract immediate low byte of negated number from register low byte.
2. Subtract immediate with carry high byte of negated number from register high byte.

By adding more Add with Carry instructions, numbers of *n*-byte width can be added using *n* instructions.

16 – 16-Bit Register Subtraction

This operation is done as follows:

1. Subtract low bytes.
2. Subtract with carry high bytes.

By adding more Subtract with Carry instructions, numbers of *n*-byte width can be subtracted using *n* instructions.

0937A-A–8/97

16-Bit Register + 16-Bit Immediate Subtraction

This operation is done as follows:

1. Subtract immediate low byte from register low byte.
2. Subtract with carry immediate high byte from register high byte.

By adding more Subtract with Carry instructions, numbers of *n*-byte width can be subtracted using *n* instructions.

Compare two 16-bit Register Variables

This operation is done as follows:

1. Compare low bytes.
2. Compare with carry high bytes.

Note that the Compare with Carry instruction supports zero-propagation, which means that all conditional branch instructions can be used following the two-step compare operation. By adding more Compare with Carry instructions, numbers of *n*-byte width can be compared using *n* instructions.

Compare a 16-bit Register with a 16-Bit Immediate

This operation is done as follows:

1. Compare register low byte to immediate low byte.
2. Store immediate high byte to a third register.
3. Compare with carry high bytes.

Negate (2's Complement) a 16-Bit Register Variable

This operation is done as follows:

1. Invert (1's Complement) low byte
2. Invert (1's Complement) high byte
3. Subtract $FF from low byte.
4. Subtract with carry $FF from high byte.

Note: Steps 3 and 4 are equivalent to adding $0001 to the 16-bit number.