

**(S, D) Register Mode (Register contains operand)**

```
mov.w #0x0100, R14
add.w #0x0004, R14
add.w R12, R14
```

**(S) Immediate Mode (#OperandValue)**

```
mov.w #0x0100, R14
add.w #0x0004, R14
```

**(S, D) Absolute Mode (&AddressOfOperand)**

```
var1: .word 0x100
var2: .word 0x200

move.w &var1, R14
add.w &var1, &var2
```

**(S, D) Indexed Mode**

```
mov.w X(Rn), &var1
```

Operand is value at address  $Rn+X$

*array1    array1+2    array1+4*

```
array1: .word 0x100, 0x200, 0x300
; array1[0] = 0x100    index 0
; array1[1] = 0x200    index 1
; array1[2] = 0x300    index 2
```

```
mov.w 0x2, R14    ; R14 used as array index
```

```
mov.w array1(R14), &var1
; move value at address array1+2 to var1
; move array1[1] to var1
```

```
incd R14    ;next index
mov.w array1(R14), &var1
; move value at address array1+4 to var1
; move array1[2] to var1
```

## Program: Array1

```
        .data
array1: .word 0x100, 0x200, 0x300
var1:   .space 2

        .text
        .
        .

        mov.w    #0x0, R14                ; index
        mov.w    array1(R14), &var1

        incd.w   R14
        mov.w    array1(R14), &var1

        incd.w   R14
        mov.w    array1(R14), &var1

        add.w    &var1, array1(R14)

Loop:   jmp      Loop
```