

## Microcontroller Technical Information

CA850 V850 C Compiler Package Document Modifications	Document No.	ZBG-CD-08-0007	1/1	
	Date issued	March 3, 2008		
Related documents CA850 Ver. 3.20 C Language: U18513EJ1 CA850 Ver. 3.20 Assembly Language: U18514EJ1 CA850 Ver. 3.20 Operation: U18512EJ1 CA850 Ver. 3.20 Link Directives: U18515EJ1 PM+ Ver. 6.30 Project Manager User's manual: U18416EJ1 CA850 Ver. 2.50 Coding Technique: U16076EJ1 CA850 Ver. 3.20 Operating Precautions: ZUD-CD-07-0168	Issued by	Development Tool Solution Group Multipurpose Microcomputer Systems Division Microcomputer Operations Unit NEC Electronics Corporation		
	Notification classification		Usage restriction	
			Upgrade	
		√	Document modification	
		Other notification		

1. Affected product

CA850 Ver. 3.20 (product name: CA703000)

2. Affected document

CA850 Ver. 3.20 C Compiler Package Assembly Language User's Manual  
 (document number: U18514EJ1V0)

3. Modified items

See the attachment.

4. Action

Revision of the above document is not planned, so refer to this document with the above documents.

## Modifications in CA850 User's Manual

### 1. Modification List

No.	Document Name	Document Number	Page
1	CA850 Ver. 3.20 C Compiler Package - Assembly Language	U18514EJ1V0	139
2	CA850 Ver. 3.20 C Compiler Package - Assembly Language	U18514EJ1V0	142
3	CA850 Ver. 3.20 C Compiler Package - Assembly Language	U18514EJ1V0	158

### 2. Modifications Details

No. 1 Correction of [Function] of *adf/adfcond* instructions

Document Name	Document Number	Page
CA850 Ver. 3.20 C Compiler Package - Assembly Language	U18514EJ1V0	139

[Modification]

Description of [Function] of the *adf/adfcond* instructions on page 139 is corrected.

Before change:

**[Function]**

- *adf*

Adds the word data of the register specified by the second operand to the word data of the register specified by the third operand.

It then compares the flag condition of the addition result with the flag condition indicated by the value of the lower 4 bits of the absolute expression (refer to Table 3 - 6) specified by the first operand. If the values match, 1 is added to the addition result and that result is stored in the register specified by the fourth operand; otherwise, 0 is added to the addition result and that result is stored in the register specified by the fourth operand.

- *adfcond*

Adds the word data of the register specified by the first operand to the word data of the register specified by the second operand.

It then compares the flag condition of the addition result with the flag condition indicated by the string in the *cond*"part. If the values match, 1 is added to the addition result and that result is stored in the register specified by the third operand; otherwise, 0 is added to the addition result and that result is stored in the register specified by the third operand.

After change:

**[Function]**

- *adf*

It compares the **current flag condition** with the flag condition indicated by the value of the lower 4 bits of the absolute expression (refer to Table 3 - 6) specified by the first operand.

If the values match, **the word data of the register specified by the second operand is added to the word data of the register specified by the third operand**, 1 is added to the addition result and that result is stored in the register specified by the fourth operand; otherwise, **the word data of the register specified by the second operand is added to the word data of the register specified by the third operand**, 0 is added to the addition result and that result is stored in the register specified by the fourth operand.

- *adfcond*

It compares the **current flag condition** with the flag condition indicated by the string in the "cond" part.

If the values match, **the word data of the register specified by the first operand is added to the word data of the register specified by the second operand**, 1 is added to the addition result and that result is stored in the register specified by the third operand; otherwise, **the word data of the register specified by the first operand is added to the word data of the register specified by the second operand**, 0 is added to the addition result and that result is stored in the register specified by the third operand.

No. 2 Correction of [Function] of *sbf/sbfcond* instructions

Document Name	Document Number	Page
CA850 Ver. 3.20 C Compiler Package - Assembly Language	U18514EJ1V0	142

**[Modification]**

Description of [Function] of the *sbf/sbfcond* instructions on page 142 is corrected.

Before change:

**[Function]**

- *sbf*

Subtracts the word data of the register specified by the second operand from the word data of the register specified by the third operand.

It then compares the flag condition of the subtraction result with the flag condition indicated by the value of the lower 4 bits of the absolute expression (refer to Table 3 - 7) specified by the first operand. If the values match, 1 is subtracted from the subtraction result and that result is stored in the register specified by the fourth operand; otherwise, 0 is subtracted from the subtraction result and that result is stored in the register specified by the fourth operand.

- *sbfcond*

Subtracts the word data of the register specified by the first operand from the word data of the register specified by the second operand.

It then compares the flag condition of the subtraction result with the flag condition indicated by the string in the "cond" part. If the values match, 1 is subtracted from the subtraction result and that result is stored in the register specified by the third operand; otherwise, 0 is subtracted from the subtraction result and that result is stored in the register specified by the third operand.

After change:

**[Function]**

- sbf

It compares the **current flag condition** with the flag condition indicated by the value of the lower 4 bits of the absolute expression (refer to Table 3 - 7) specified by the first operand.

If the values match, **the word data of the register specified by the second operand is subtracted from the word data of the register specified by the third operand**, 1 is subtracted from the subtraction result and that result is stored in the register specified by the fourth operand; otherwise, **the word data of the register specified by the second operand is subtracted from the word data of the register specified by the third operand**, 0 is subtracted from the subtraction result and that result is stored in the register specified by the fourth operand.

- sbfcond

It compares the **current flag condition** of the subtraction result with the flag condition indicated by the string in the "cond" part. If the values match, **the word data of the register specified by the first operand is subtracted from the word data of the register specified by the second operand**, 1 is subtracted from the subtraction result and that result is stored in the register specified by the third operand; otherwise, **the word data of the register specified by the first operand is subtracted from the word data of the register specified by the second operand**, 0 is subtracted from the subtraction result and that result is stored in the register specified by the third operand.

No. 3 Correction of [Caution] for satsubi instruction

Document Name	Document Number	Page
CA850 Ver. 3.20 C Compiler Package - Assembly Language	U18514EJ1V0	158

[Modification]

Description of [Caution] for the satsubi instruction on page 158 is corrected.

Before change:

**[Caution]**

- If r0 is specified by the second operand when the V850Ex is used as the target device, the as850 outputs the following message and stops assembling.

```
E3240: illegal operand (can not use r0 as destination in V850E mode)
```

After change:

**[Caution]**

- If r0 is specified by the **third** operand when the V850Ex is used as the target device, the as850 outputs the following message and stops assembling.

```
E3240: illegal operand (can not use r0 as destination in V850E mode)
```