

プログラムをRAM上で実行する方法

RL78ファミリ用Cコンパイラ CC-RL

ルネサス システムデザイン株式会社
ツールビジネス本部 ツール技術部

2015/07/24 Rev. 1.00

R20UT3512JJ0100

はじめに

- **本資料は、RL78ファミリ用Cコンパイラ CC-RLを使用して、RAM上にプログラムをコピーして実行する方法を説明しています。**
- **本資料は、次のツール、バージョンで説明をしています。**
 - **RL78ファミリ用Cコンパイラ CC-RL V1.01.00**
 - **統合開発環境 e2 studio V4.0.0.26**
 - **統合開発環境 CS+ V3.01.00**

- **プログラムをRAM上で実行するには**
- **リンク時のマッピングイメージ**
- **マイコンプログラム上での実行時イメージ**
- **Cソース上での関数へのfar属性の追加**
- **Cソース上でのセクション指定の追加**
- **リンカの設定**
 - **ROMからRAMにマップするセクション指定(-romオプション)**
 - **セクションの配置指定(-startオプション)**
- **Cソース上での関数をRAMへコピーする処理の追加**
- **サンプルプログラム**

プログラムをRAM上で実行するには

■ Cソース上での設定

- `__far`修飾子を使用して関数をfar属性に変更
- `#pragma section` で変数のセクション名を変更

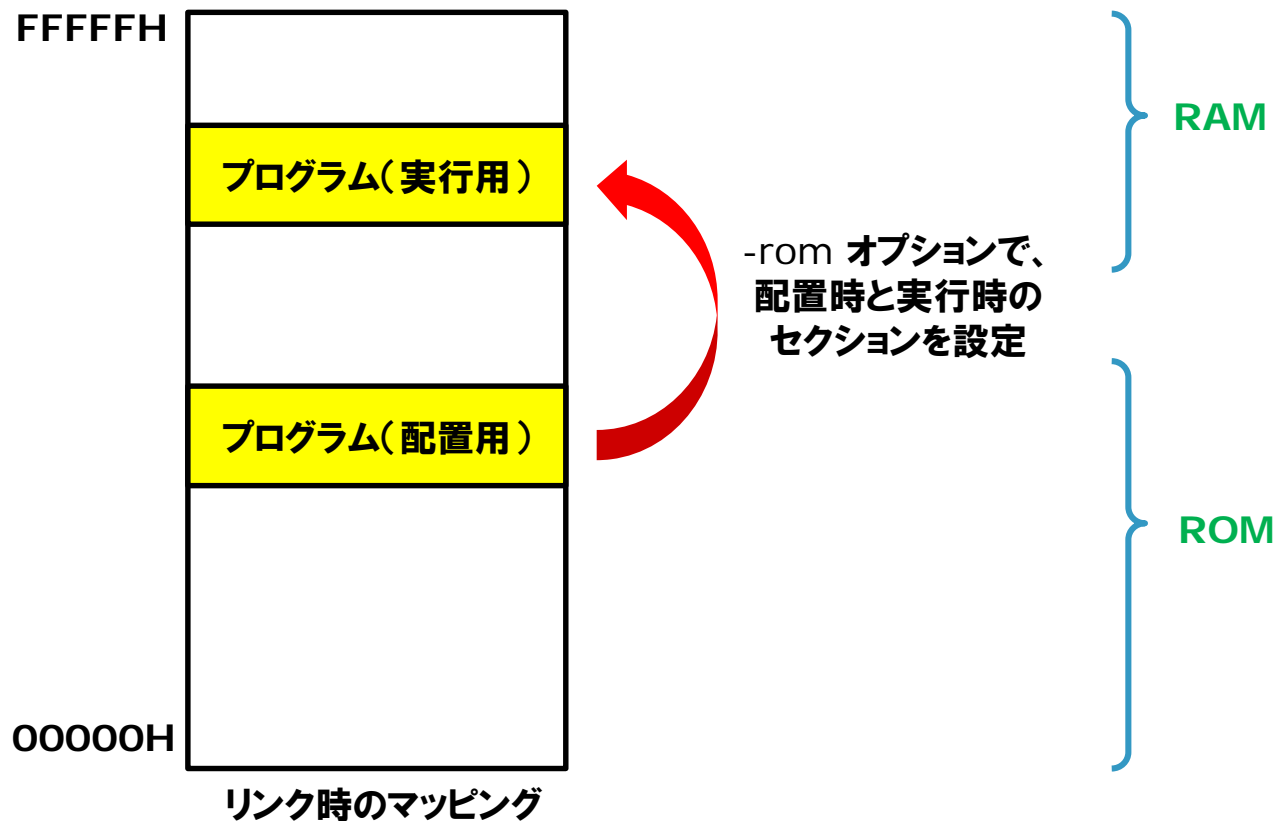
■ リンカの設定の追加

- RAMで実行する関数のセクションをROMからRAMへマップするセクションとして指定(`-rom`オプション)
- RAMで実行する関数のセクションの配置指定(`-start`オプション)

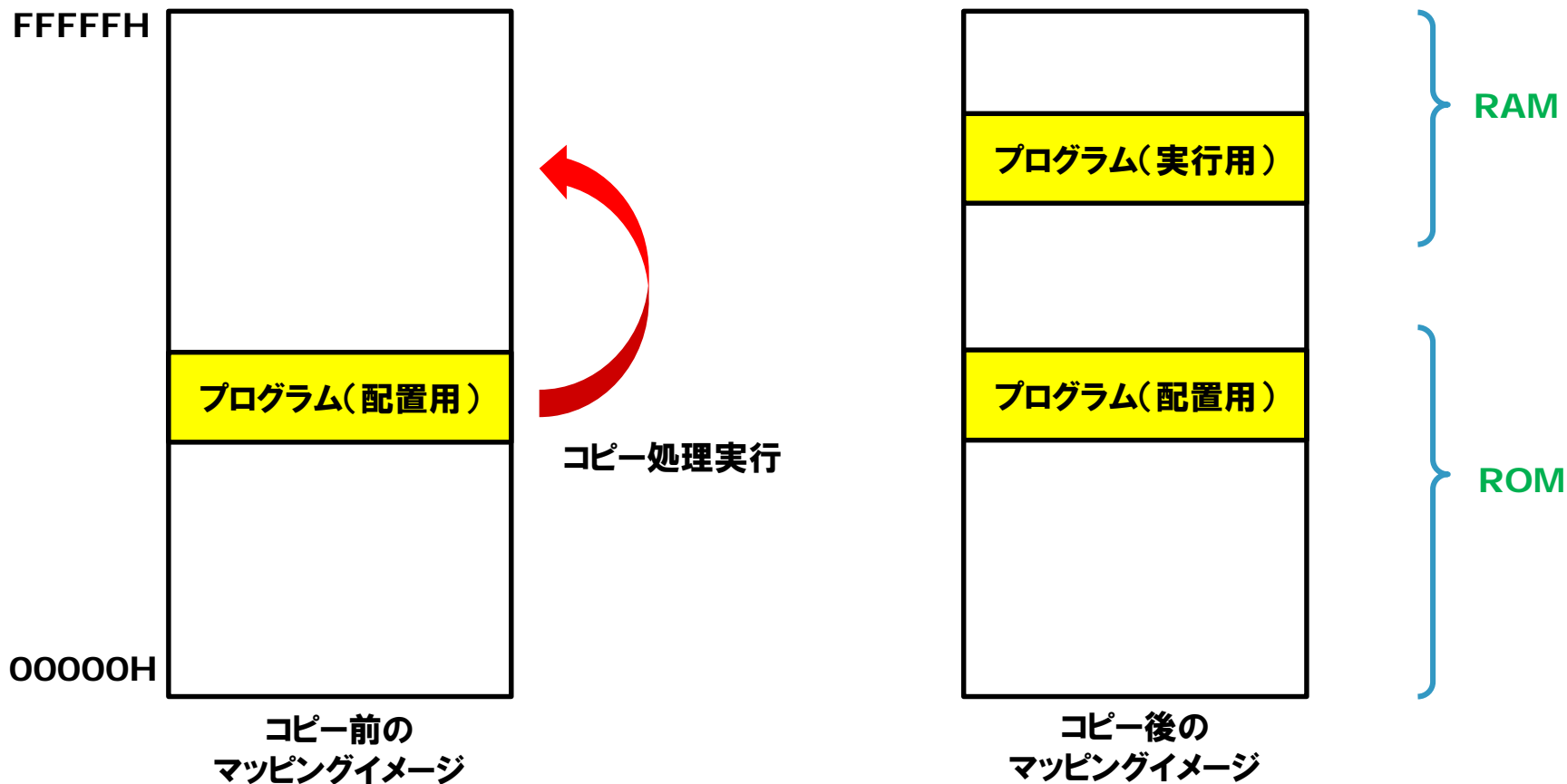
■ 関数をRAMへコピーする処理の追加

- 関数を実行する前に、RAMで実行するプログラムのセクションを、RAMにコピーする処理を追加

リンク時のマッピングイメージ



マイコンプログラム上での実行時イメージ



Cソース上での関数へのfar属性の追加

■ `_far`修飾子を使用

- メモリモデルの設定に影響しないように`__far`修飾子で明示的に関数をfar属性にする
- (例)

```
__far char f1(char a)
{
    if (a > 0) { return a; }
    else      { return 0; }
}
__far int f2(int x)
{
    f1(x);
    return x+x;
}
```

far属性を指定

far属性を指定

Cソース上でのセクション指定の追加

■ #pragma section を使用

- デフォルトで出力されるセクションの名前を変更する

- 記述形式

- #pragma section [セクション種別] [変更セクション名]

- セクション種別

- text、const、data、bss

- (例)

```
#pragma section text ram_text
__far char f1(char a)
{
    if (a > 0) { return a;}
    else      { return 0;}
}
__far int f2(int x)
{
    f1(x);
    return x+x;
}
#pragma section
```

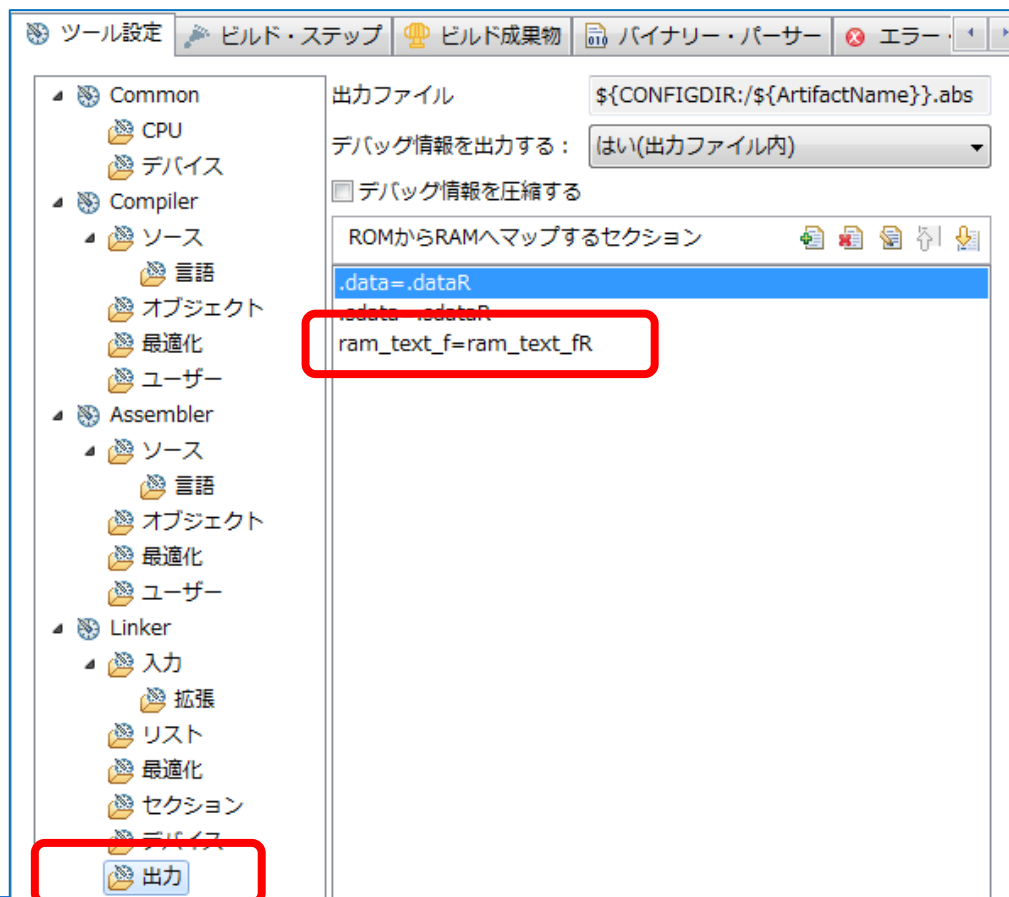
ユーザセクション名に変更

デフォルトのセクション名に戻す

リンカの設定(ROMからRAMにマップするセクション指定)(1/2)

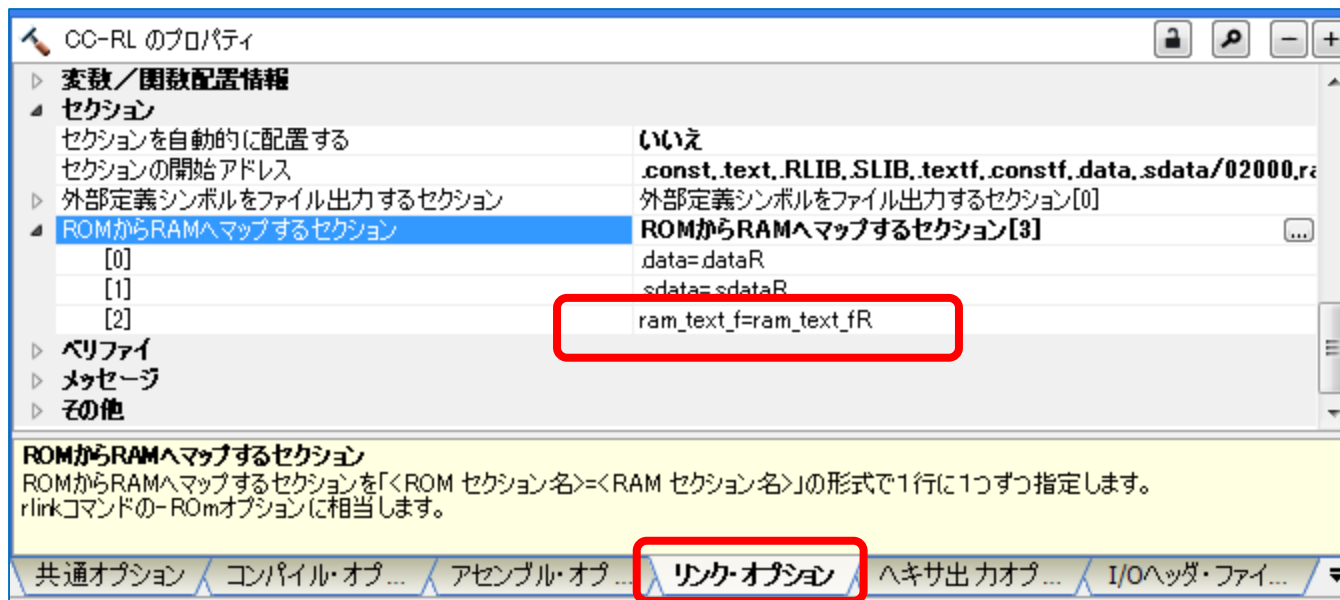
- リンカの-romオプションで、RAMで実行する関数のROM、RAMのセクションの指定してください。

- (例) e² studio



リンクの設定(ROMからRAMにマップするセクション指定)(2/2)

- (例) CS+

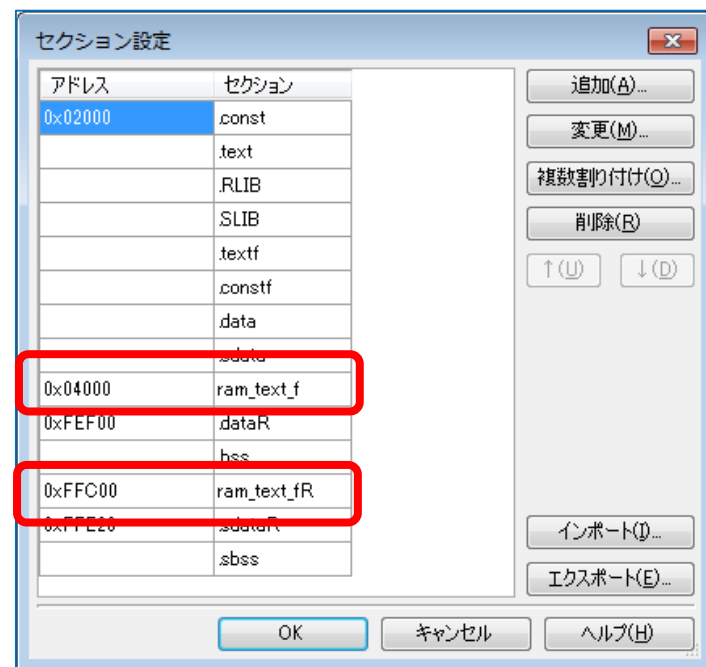
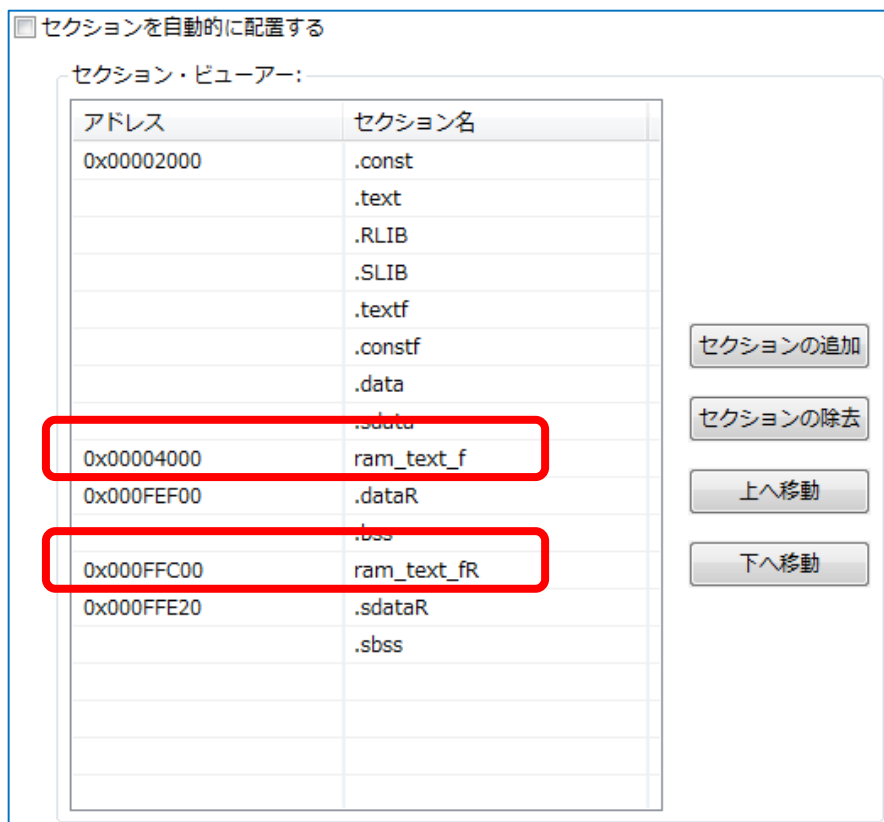


リンカの設定(セクションの配置指定)

- リンカの-startオプションで、RAMで実行する関数のROM、RAMのセクションの配置を指定してください。

- (例) e² studio

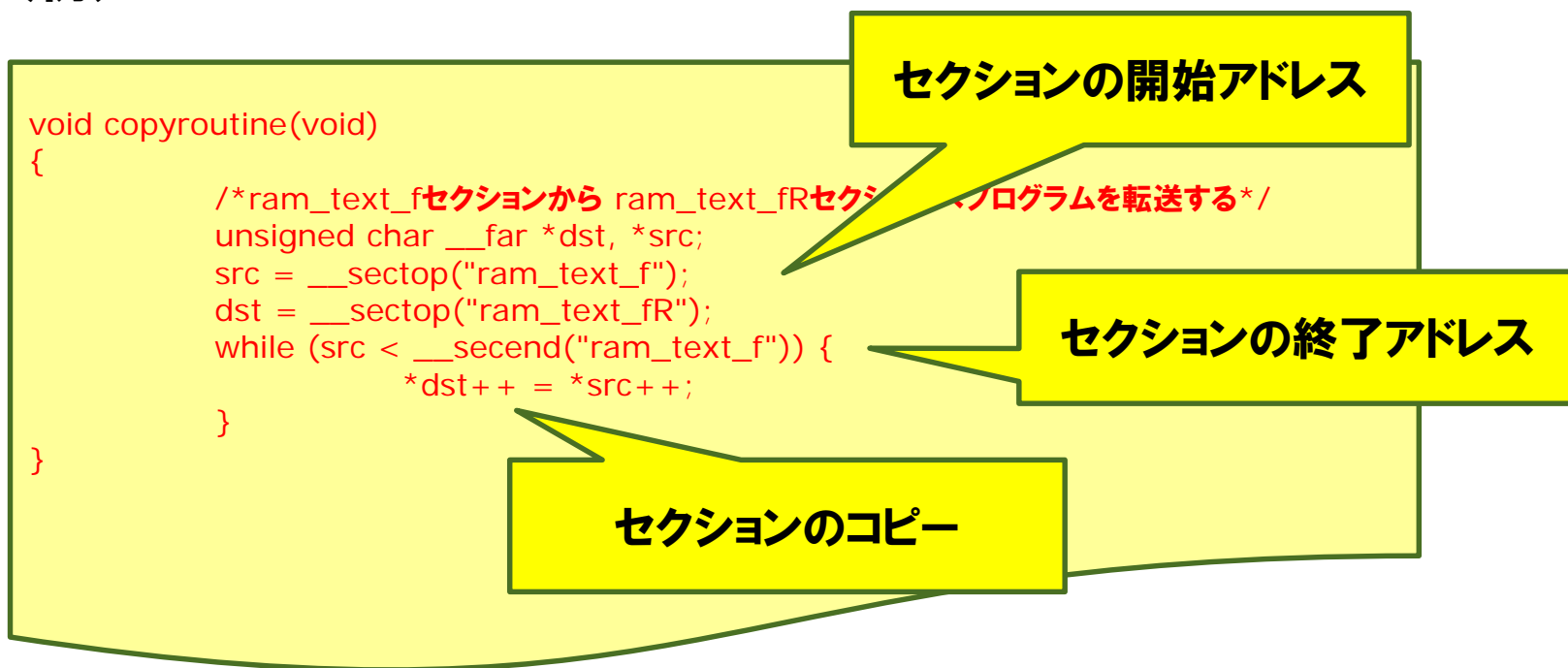
CS+



Cソース上での関数をRAMへコピーする処理の追加

■ RAMで実行する関数をコピーする処理を追加する

- `__sectop`、`__secend`を利用してコピーする
- このコピー関数をRAMで実行する前に実行する
- (例)



サンプルプログラム

- 今まで作成したプログラムを使用した例を記載します。

```
void copyroutine(void);          /* コピー処理の関数のプロトタイプ宣言 */
__far char f1(char a);          /* RAMで実行する関数のプロトタイプ宣言 */
__far int f2(int x);            /* RAMで実行する関数のプロトタイプ宣言 */

int a;

void func(void)
{
    ...
    /* コピーする処理を呼び出す */
    copyroutine();
    /* RAMで実行する関数を呼び出す */
    a = f2(a);
    ...
}
```



ルネサス システムデザイン株式会社

©2015 Renesas System Design Co., Ltd. All rights reserved.