

Application note for DELFEE

DELFEE

The DELFEE language describes where code or data sections should be placed on the actual memory chips. This language has to define the interface between a virtual world (the software) and a physical world (the hardware configuration).

In the DELFEE language the interface between virtual and physical world is described in three parts:

1. Software Part (*.DSC)

The software part belongs to the virtual world and describes the ordering of the data and code sections. The software part may vary for different applications and can even be empty.

2. CPU Part (*.cpu)

The cpu part is the interface between the virtual world and the real world.

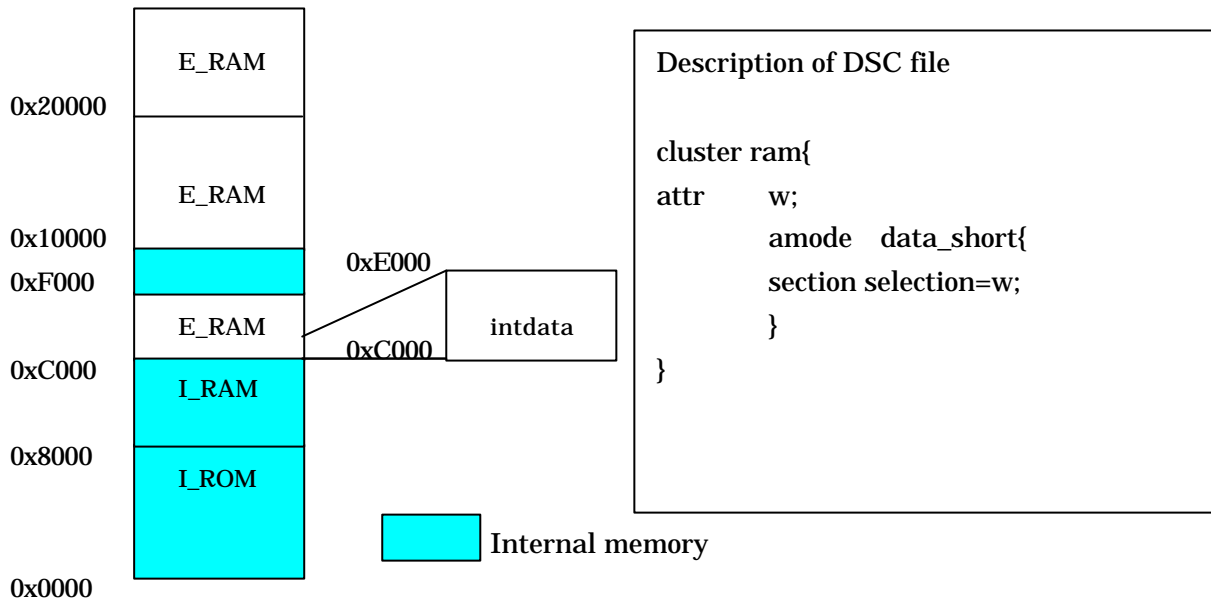
It contains the application independent part of the virtual world (the address translation of addressing modes to the addressing space), and the configuration independent part of the physical world (on-chip memory, address busses). The cpu part is independent of application and configuration.

3. Memory Part (*.mem)

The memory belongs to the physical world. It contains the description of the external memory. The memory part may vary for different configurations and can even be empty (if there is no external memory).

Q1.

If you look at the map file, you will notice that the section intdata is located at chip e_ram.

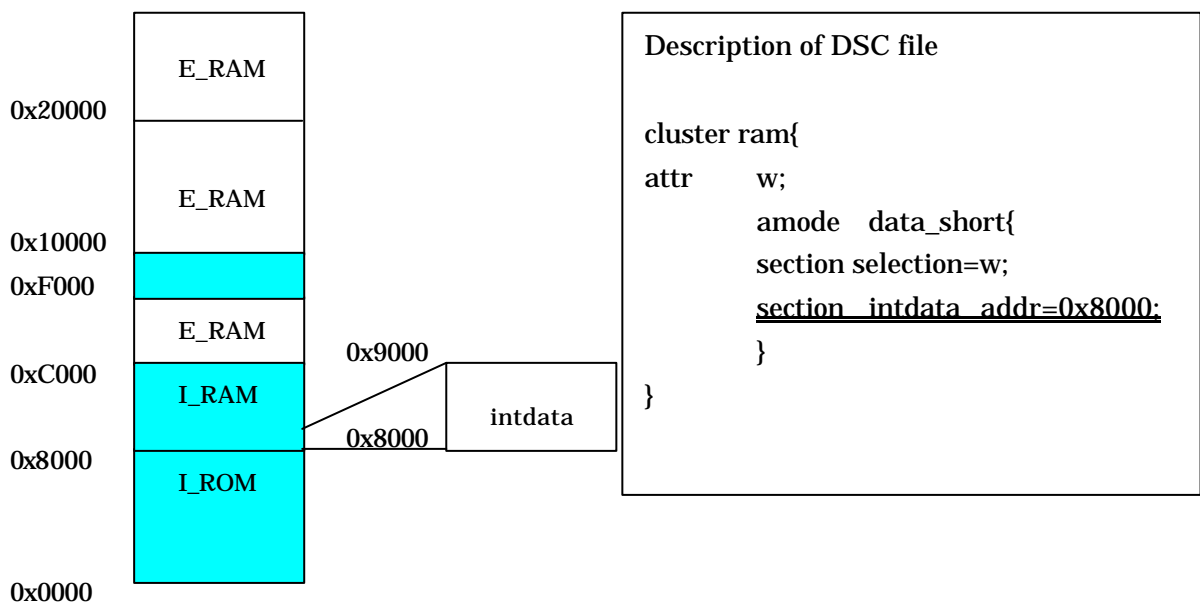


I want this section to be located in (fast) internal memory i_ram.

A1.

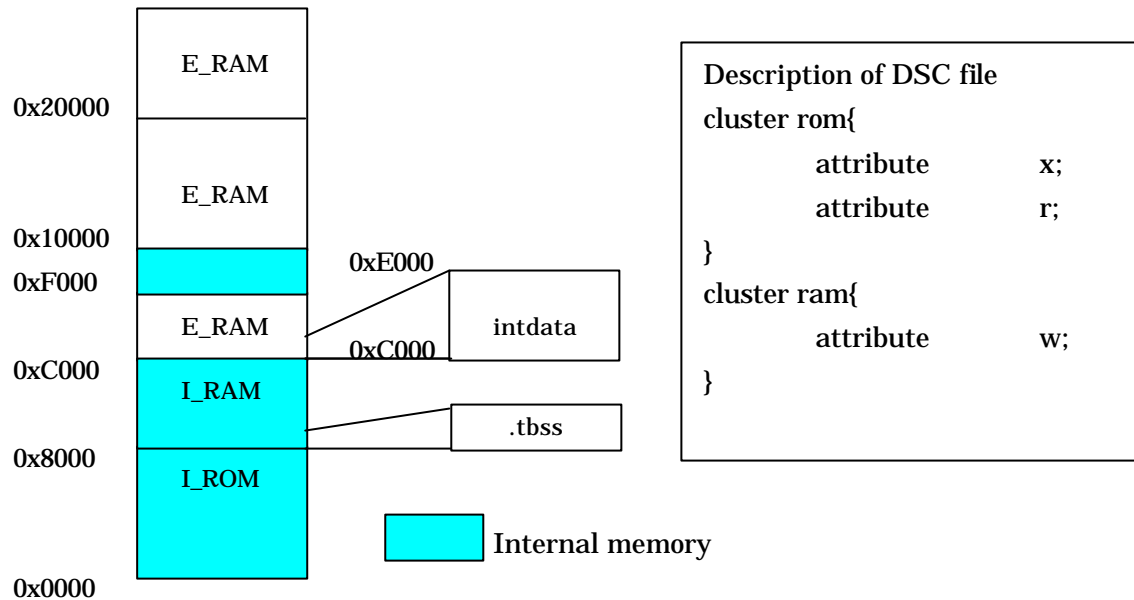
Use an absolute address in the DELFEE file to do so.

Specify the address 0x8000 in DSC file, The section intdata is located in internal memory.



Q2.

The following is the memory map.

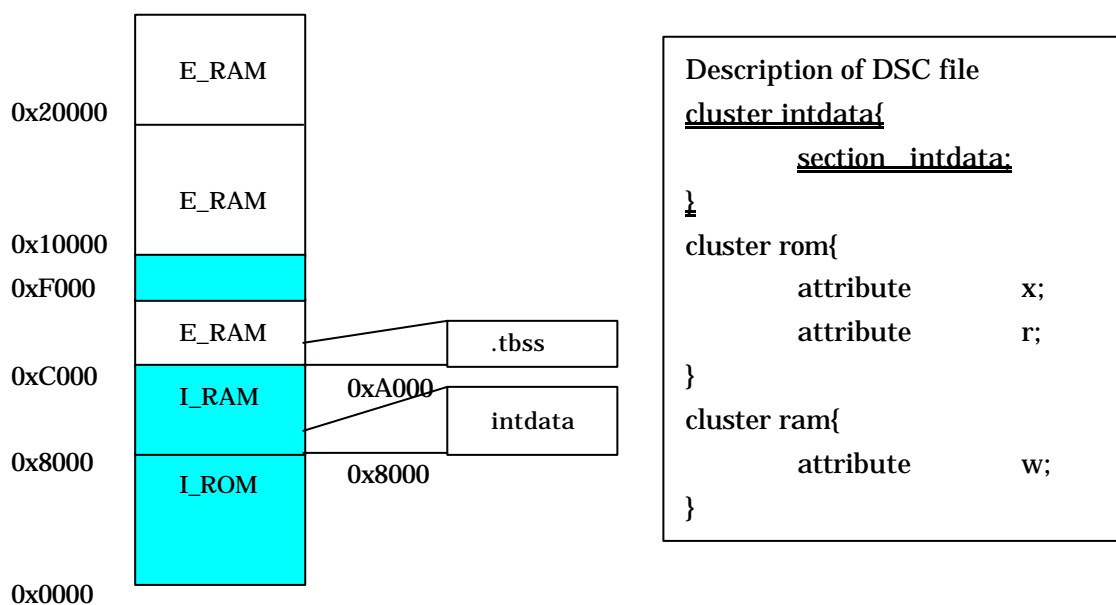


How do we give section intdata a higher priority than section .tbss so intdata gets located first?

A2.

The order of priority of the location of memory depends on the order of clusters in the description.

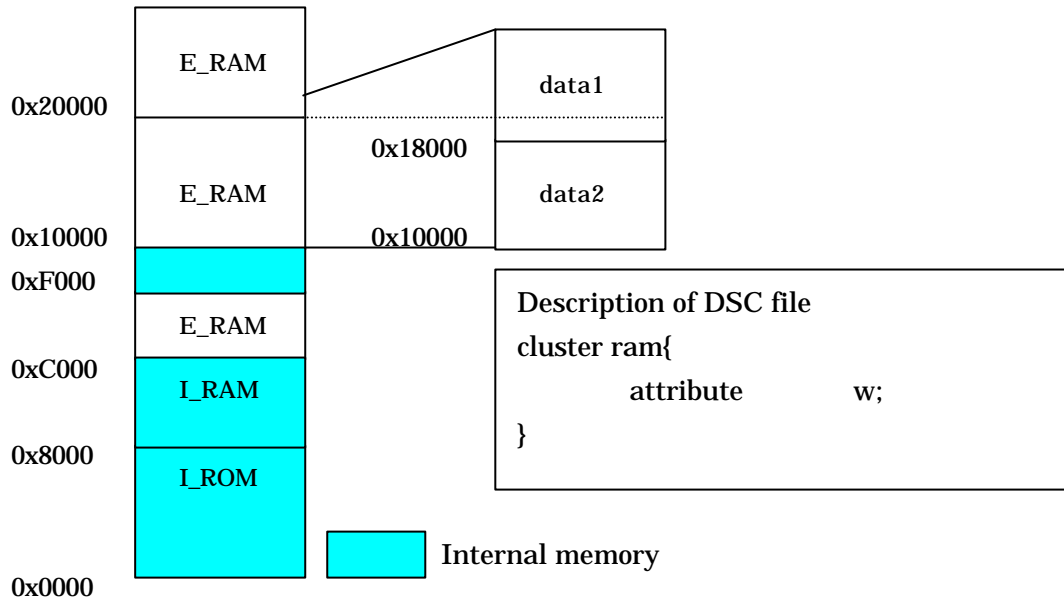
Create intdata entry on rom entry.



In the above example, the intdata section is located before the other writable section of attributes.

Q3.

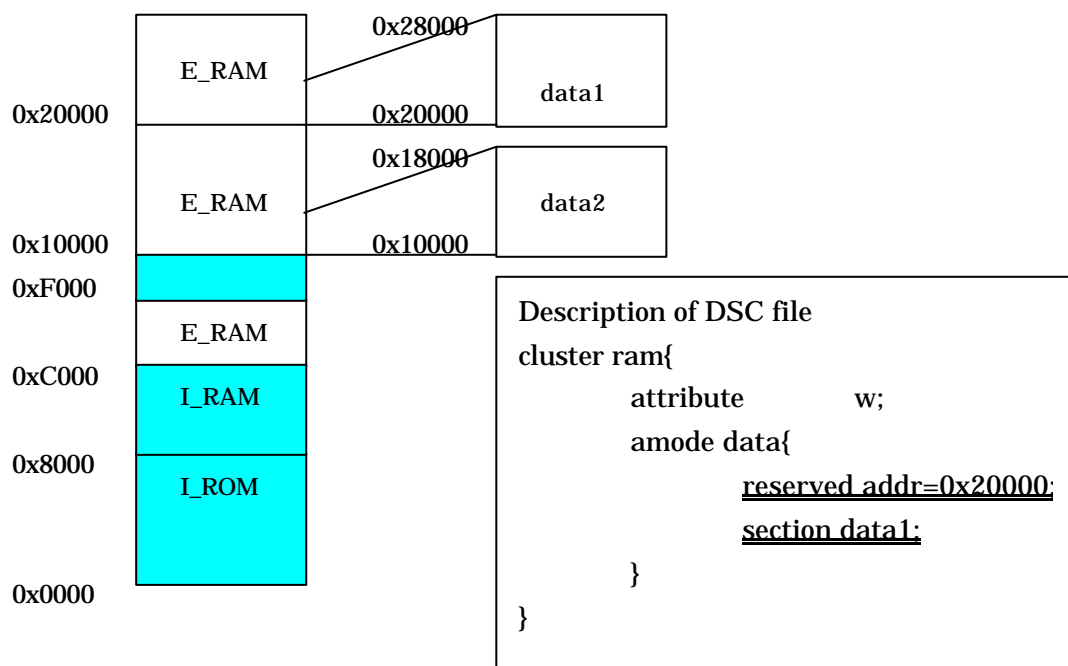
Data1 section gets located on both e_ram1 and e_ram2.



How do we arrange it such that data1 and data2 are both in their own RAM chip?

A3.

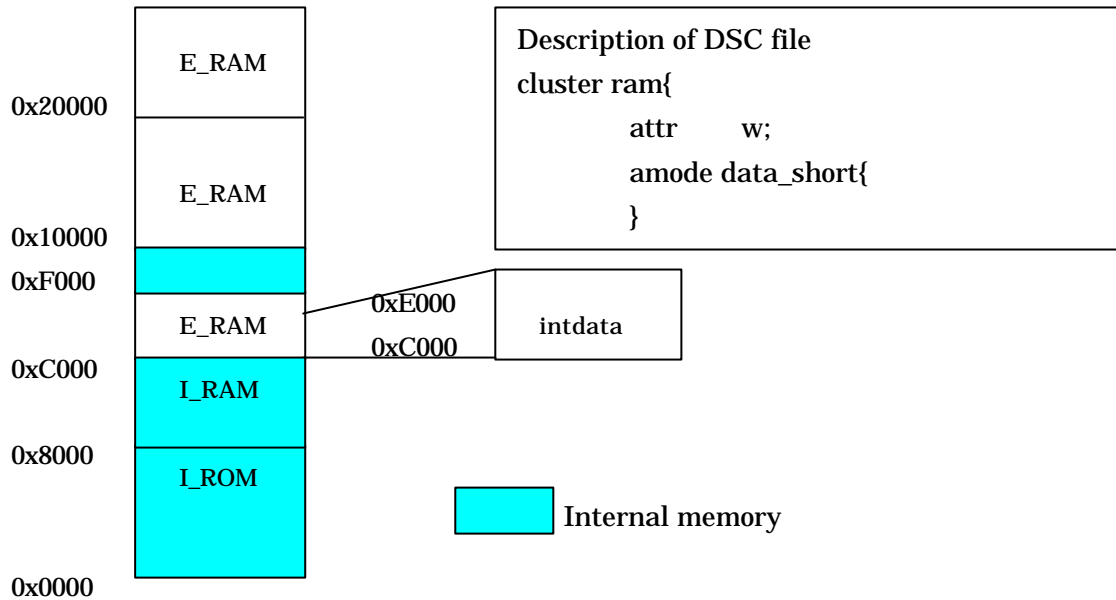
The section data1 is located in address 0x2000.



You can use keyword "fixed addr=0x2000;" instead of "reserved".

Q4.

Section intdata is located in RAM.



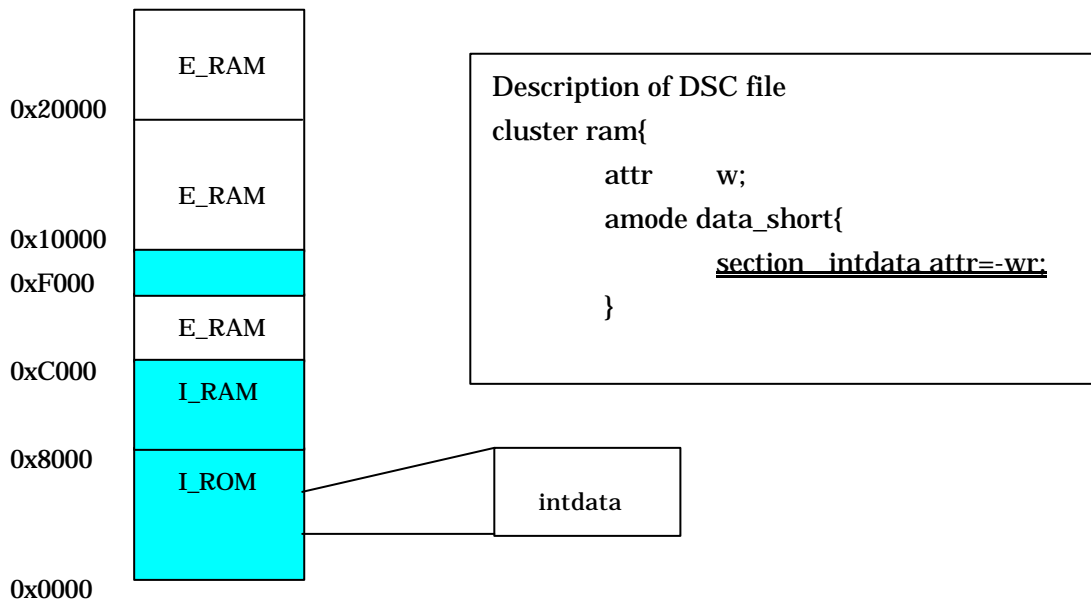
Section intdata has some integer data. These data are constant.

How can we change the DELFEE file so they end up in ROM?

A4.

If more attributes are specified, only sections with all attributes are selected. If a minus sign '-' precedes the attribute, only sections not having the attribute are selected.

Replace the w attribute with r attribute.

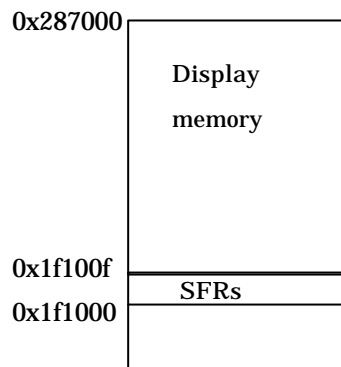


Q5.

You have added an external LCD driver to your board.

It has SFRs from 0x1f0000-0x1f000f and display memory from 0x1f1000-0x287000.

In our program, we access the display memory with `__lc_ub_disp_start`.



What changes are required in the MEM and DSC file?

Description of MEM file

```
memory {  
    bus external_bus {  
        mau 8;  
    }  
}
```

Description of DSC file

```
cluster ram{  
    attr w;  
    amode data{  
    }  
}
```

A5.

Add a mapping for the new chip onto the bus.

Add the new chip.

Description of MEM file

```
memory {  
    bus external_bus {  
        mau 8;  
        mem addr=0x1f1000 chips=lcd_drv;  
        mem addr=0x1f1010 chips=dis_mem;  
    }  
    chips lcd_drv attr=s mau=8 size=0x10;  
    chips dis_mem attr=s mau=8 size=0x96000;  
}
```

The label of the section disp_start is made in 0x1f1000.

Description of DSC file

```
Cluster ram{  
    attr w;  
    amode data{  
        reserved label=disp_start addr=0x1f1000;  
    }  
}
```


Q6.

The original model had its LCD sfrs at address 0x0f00, A new model has the LCD sfrs at address 0xff00.

LCD_SFRS address 0x0f00 -> 0xff00

Description of DSC file

```
cluster ram{
    attr    w;
    amode data{
    }
}
```

How should we change the sfrs address?

A6.

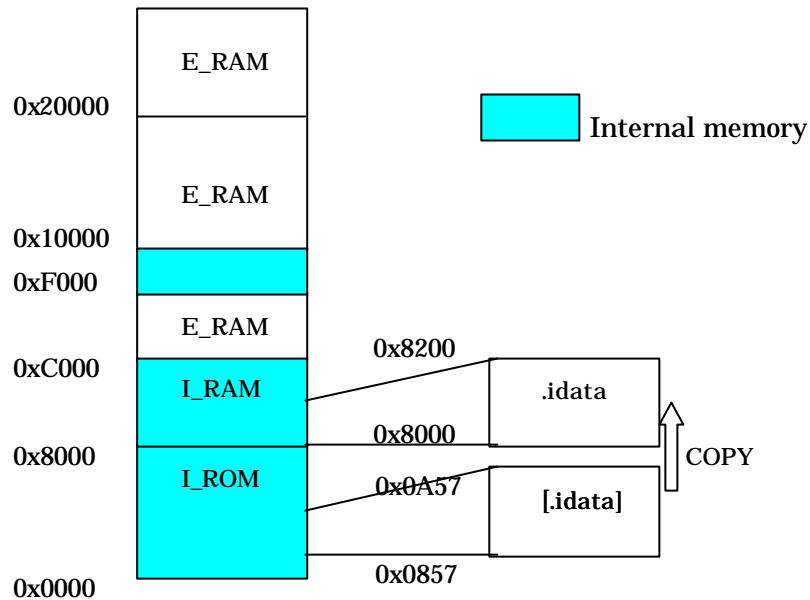
Locate a section LCD_sfrs at a certain address.

Description of DSC file

```
cluster ram{
    attr    w;
    amode data{
        section _LCD_sfrs addr=0xff00:
    }
}
```

Q7.

The section `.idata` has a copy in ROM (in your map file this is represented as `[_idata]`). This is copied from ROM to RAM during startup.



Note that the section `.idata` in ROM only consists of zeros.

What modifications are required for the program to be located there and `.idata` is still initialized properly.

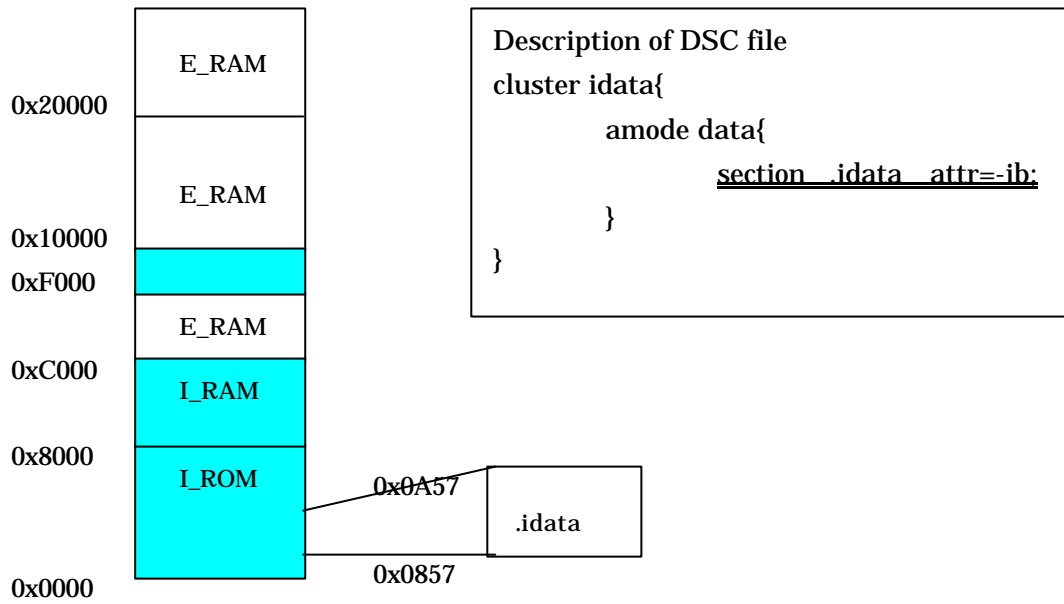
```

Description of DSC file
cluster idata{
    amode data{
    }
}

```

A7.

Add b attribute and remove i attributes by the - option.



Q8.

In the DELFEE file the area from 0x0-0x40 is reserved.
It is used for interrupt vectors.

__lc_u_reset_vector 0x20

Change the DELFEE file so the area from 0x0-0x40 is still reserved and
__lc_u_reset_vector points to the correct address as well.

Description of DSC file

```
cluster rom {  
    attribute x;  
    attribute r;  
    amode code_short {  
        reserved addr=0x0 length=0x40;  
    }  
}
```

A8.

Address is reserved by the label "reset_vector".

Description of DSC file

```
cluster rom {  
    attribute x;  
    attribute r;  
    amode code_short {  
        reserved label=reset_vector      addr=0x20;  
        reserved addr=0x0 length=0x40;  
    }  
}
```

Q9.

Suppose the section my_data and all of “ b” attributes section must be located after 0x1234 .

What changes are required in the DSC file?

Description of DSC file

```
cluster rom {  
    attribute x;  
    attribute r;  
    amode data{  
    }  
}
```

A9.

Use fixed keyword.

It is possible that it is located after the address specified with the fixed key word.

Description of DSC file

```
cluster rom {  
    attribute x;  
    attribute r;  
    amode data{  
        fixed addr=0x1234:  
        section my_data:  
        section selection=b:  
    }  
}
```