Document No.:

Application Note

Module name: C0201QIL…Series

Issue date: 2008/03/10

Version: 1.0

Note:

1. The information contained herein may be change without prior notice. It is therefore advisable to contact Chi MEI ELCorp before designed your product based on this specification.

Reversion History

Version	Date	Page	Description
Ver.1.0	2008.03.10	All	Application note was first issued

1. Application Circuit

1.1 DC/DC circuit (NCP5810)

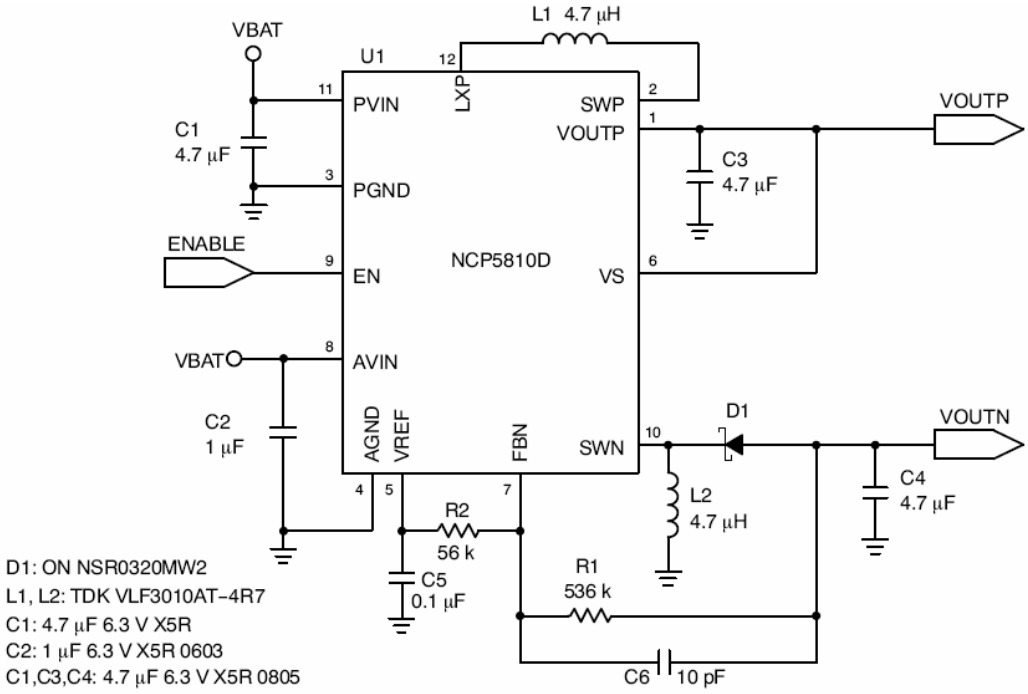
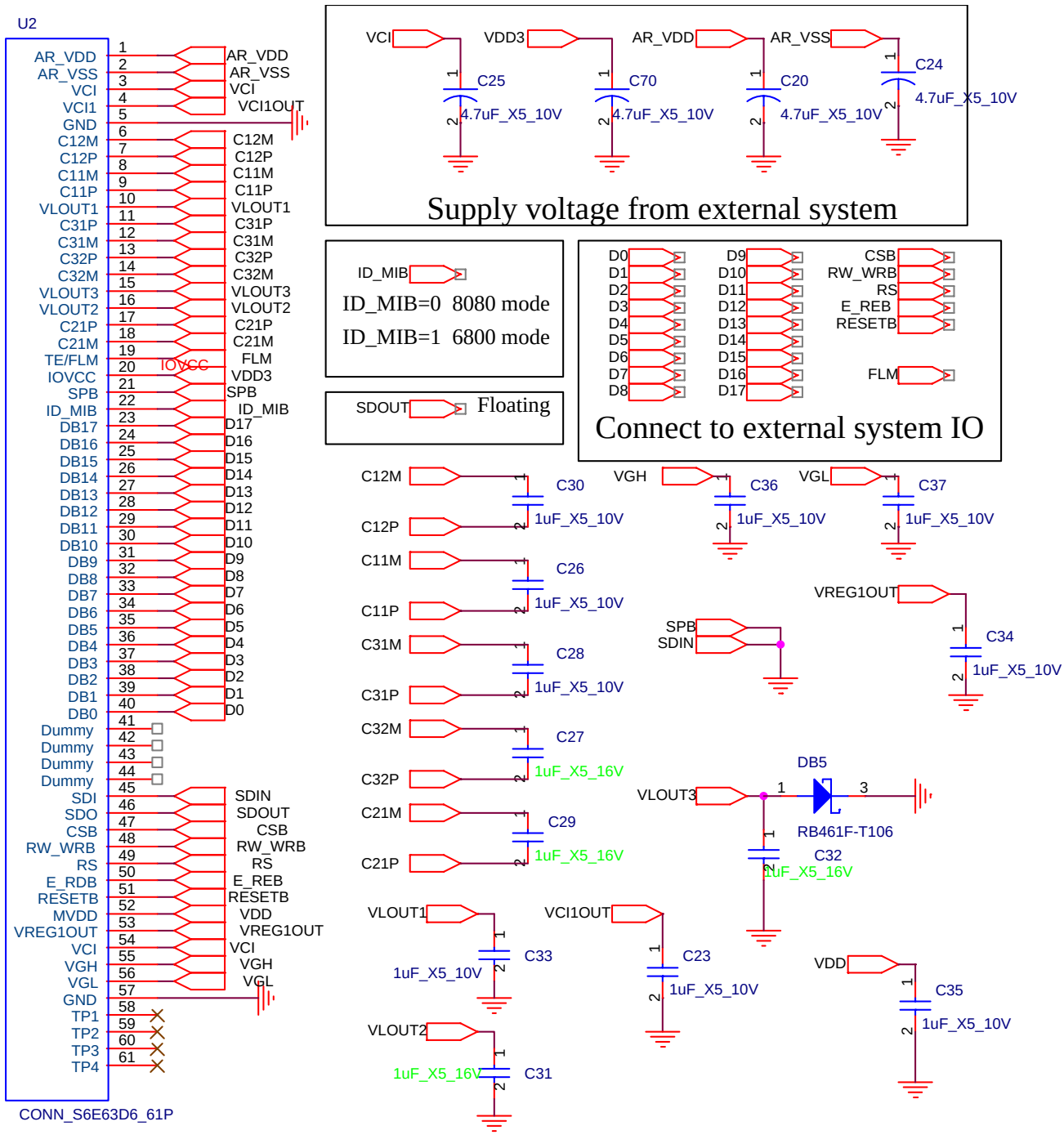


Figure 1. Typical Application Circuit

Note: For more details, please refer to NCP5810D Datasheet.

1.2 Drive IC CPU interface definition

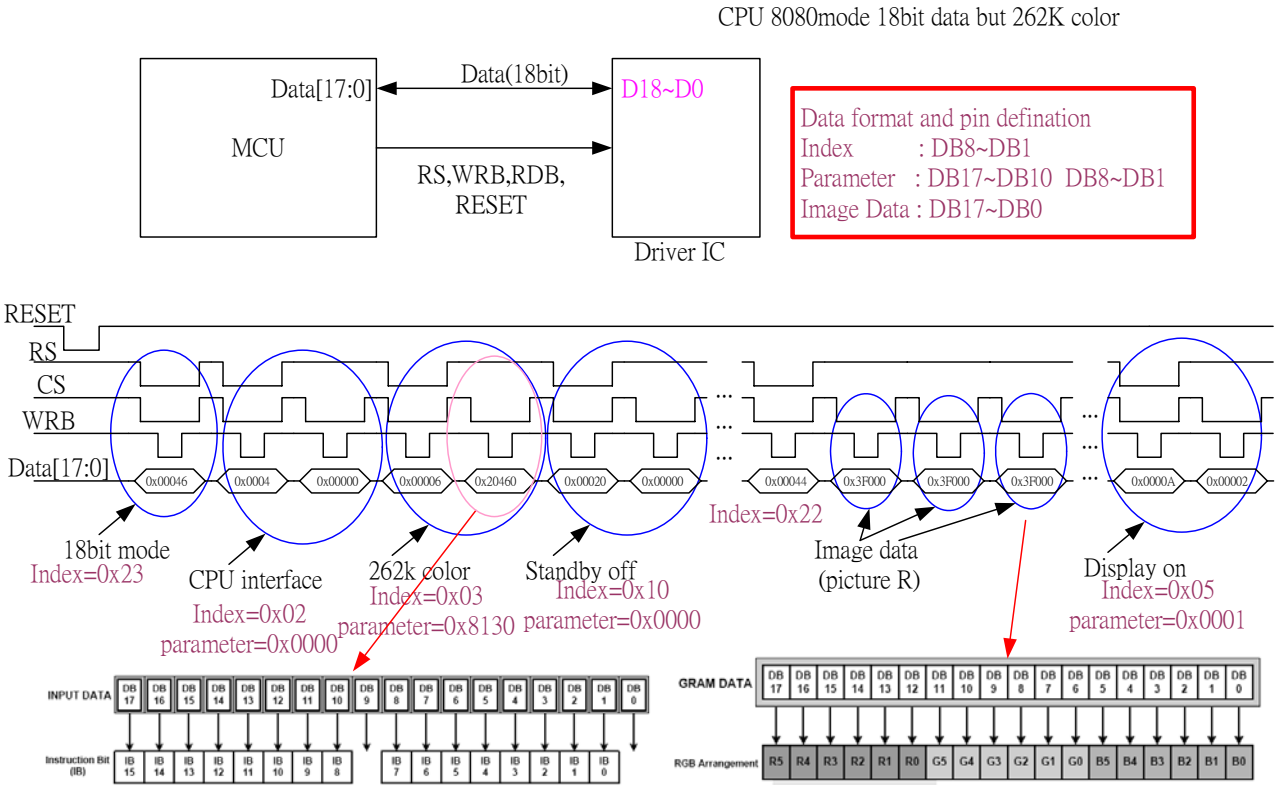


1.3. Drive IC interface spec – BUS spec.

Bus width	Pin selection	note
18-bit interface	DB17-0	
16-bit interface	DB17-10, DB8-1	
9-bit interface	DB8-0	
8-bit interface	DB8-1	

ID_MIB= high 6800 mode

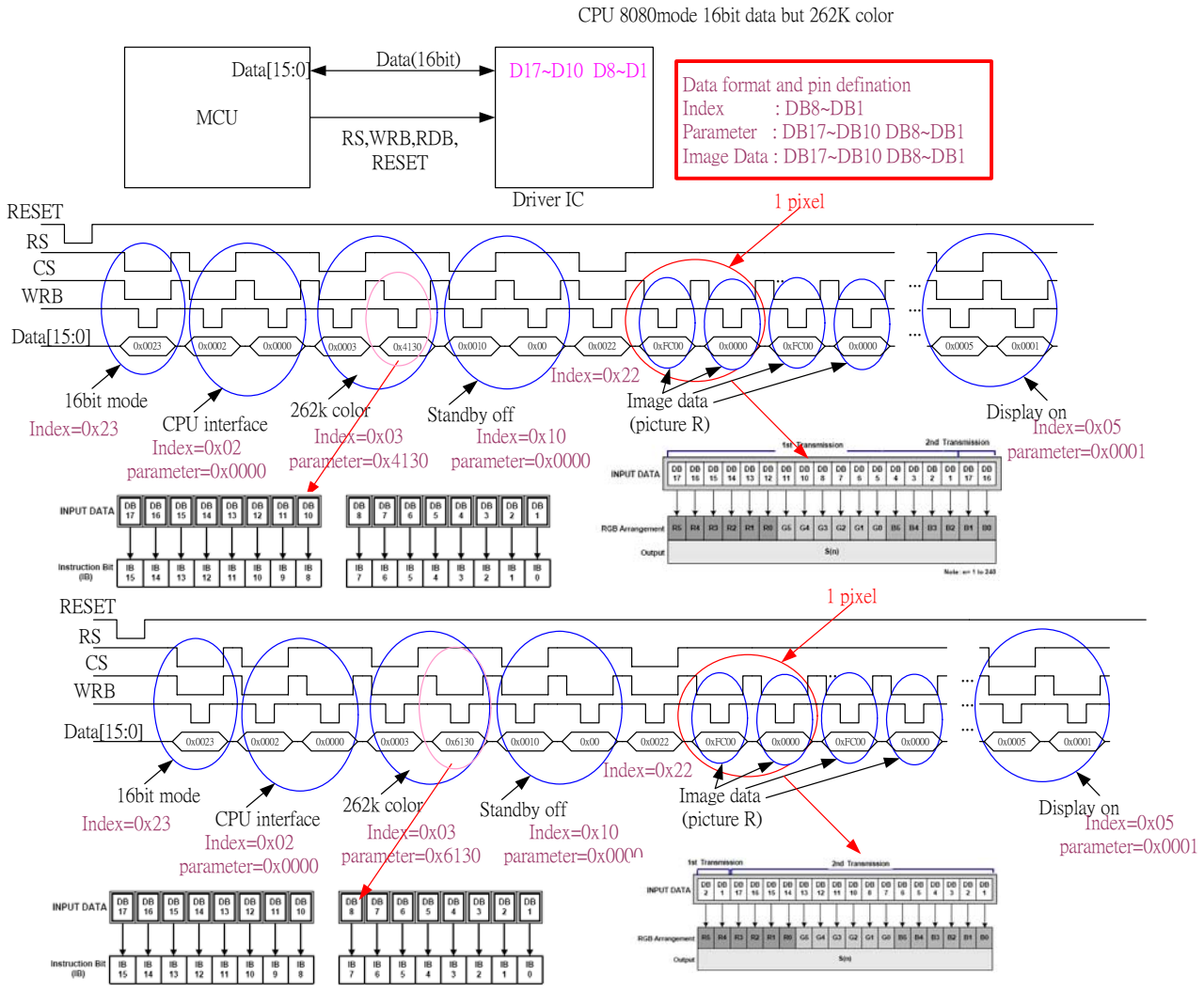
ID_MIB=low 8080 mode



Initial code:

```
Index_out(0x23);
Index_out(0x02);
Parameter_out(0x0000);
Index_out(0x03);
Parameter_out(0x8130);
Index_out(0x10);                    // standby off
Parameter_out(0x0000);
//******* below is gamma setting *********//
Index_out(0x70);
Parameter_out(0x????);
Index_out(0x71);
```

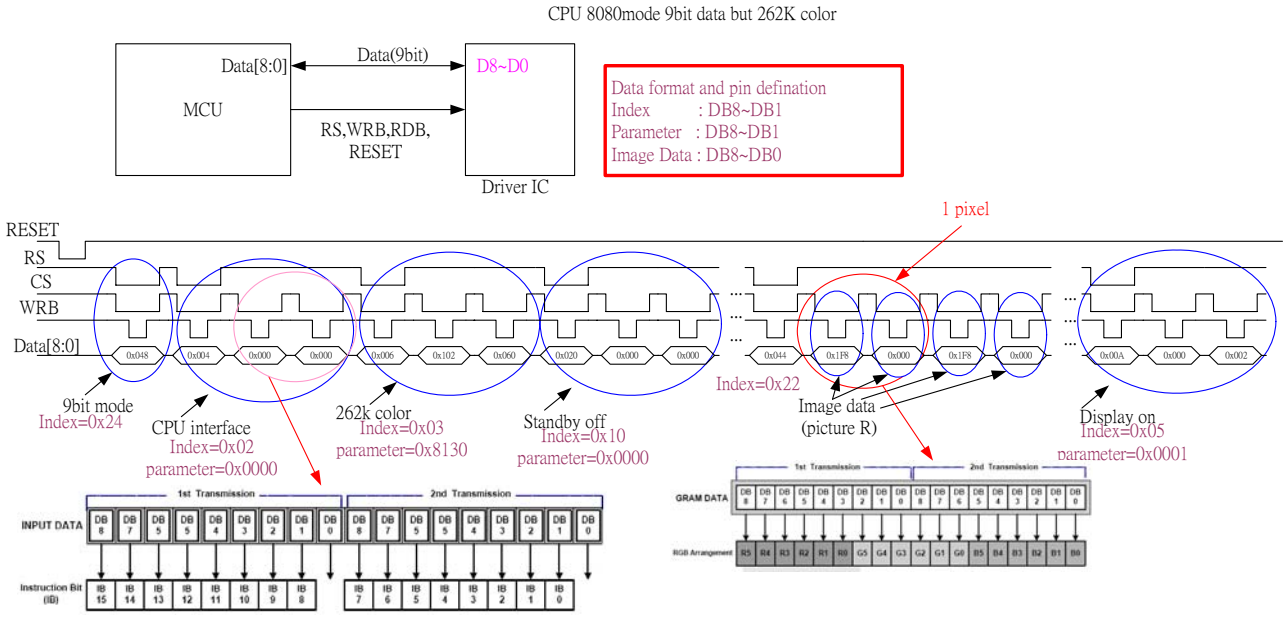
```
Parameter_out(0x????);
……
……
……
Index_out(0x78);
Parameter_out(0x????);
//*************************************************
H_start_address=0x20;
H_end_address=0xAF + H_start_address;
V_start_address= 0x00;
V_end_address= 0xDB;
Temp= (H_start_address << 8) | H_end_address;
Index_out(0x35);
Parameter_out(V_start_address);
Index_out(0x36);
Parameter_out(V_end_address);
Index_out(0x37);
Parameter_out(Temp);
Index_out(0x20);
Parameter_out(H_start_address);
Index_out(0x21);
Parameter_out(V_start_address);
Index_out(0x22);
For(I=0;I<176*220;I++)            // 176*rgb*220 image data
    Parameter_out(0x00000);    // rgb image datra 18bit
//*************************************************
Index_out(0x05);        // display on
Parameter_out(0x0001);
```



Initial code:

Index_out(0x23);
Index_out(0x02);
Parameter_out(0x0000);
Index_out(0x03);
Parameter_out(0x4130); or Parameter_out(0x6130);
Index_out(0x10); // standby off
Parameter_out(0x0000);
//******* below is gamma setting *********//
Index_out(0x70);
Parameter_out(0x????);
Index_out(0x71);
Parameter_out(0x????);
……
……

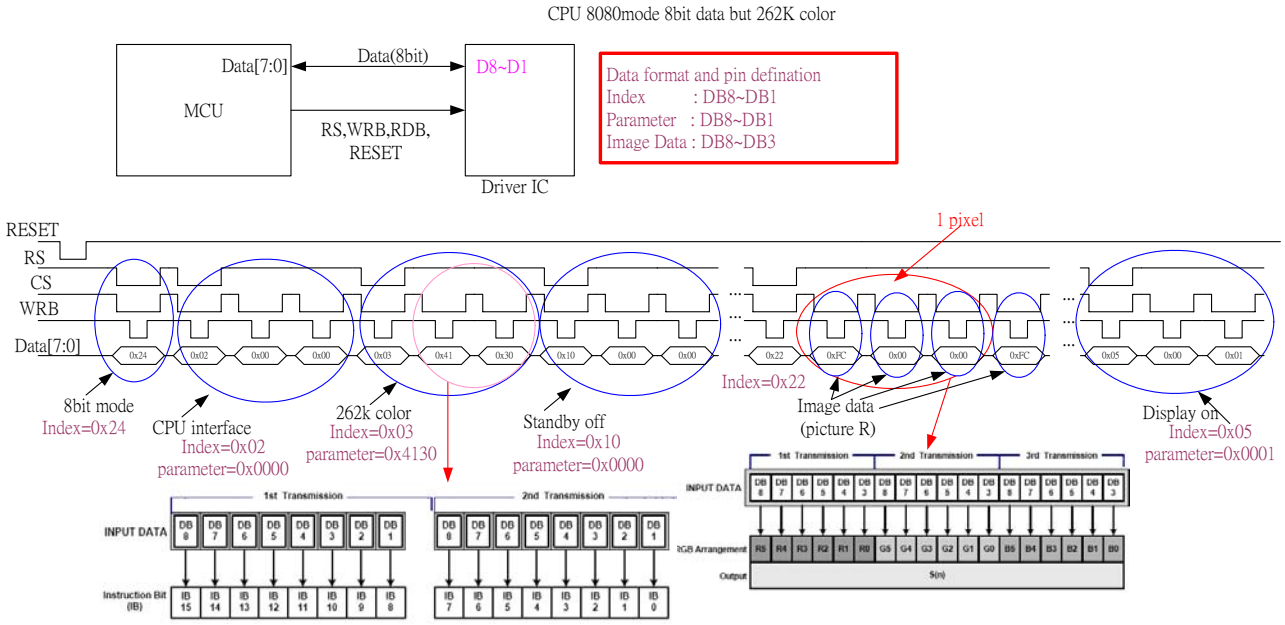
```
……….
Index_out(0x78);
Parameter_out(0x????);
//*************************************************
H_start_address=0x20;
H_end_address=0xAF + H_start_address;
V_start_address= 0x00;
V_end_address= 0xDB;
Temp= (H_start_address << 8) | H_end_address;
Index_out(0x35);
Parameter_out(V_start_address);
Index_out(0x36);
Parameter_out(V_end_address);
Index_out(0x37);
Parameter_out(Temp);
Index_out(0x20);
Parameter_out(H_start_address);
Index_out(0x21);
Parameter_out(V_start_address);
Index_out(0x22);
For(I=0;I<176*220;I++)          // 176*rgb*220 image data
  {
      Parameter_out(0x00000);   // rgb image datra 16bit     1'st
      Parameter_out(0x00000);   // rgb image datra 16bit     2'st
   }
//*************************************************
Index_out(0x05);        // display on
Parameter_out(0x0001);
```



Initial code:

Index_out(0x24);
Index_out(0x02);
Parameter_out(0x0000);
Index_out(0x03);
Parameter_out(0x8130);
Index_out(0x10); // stadnby off
Parameter_out(0x0000);
//******* below is gamma setting *********//
Index_out(0x70);
Parameter_out(0x????);
Index_out(0x71);
Parameter_out(0x????);
……
……
……
Index_out(0x78);
Parameter_out(0x????);
//**
H_start_address=0x20;
H_end_address=0xAF + H_start_address;
V_start_address= 0x00;
V_end_address= 0xDB;
Temp= (H_start_address << 8) | H_end_address;

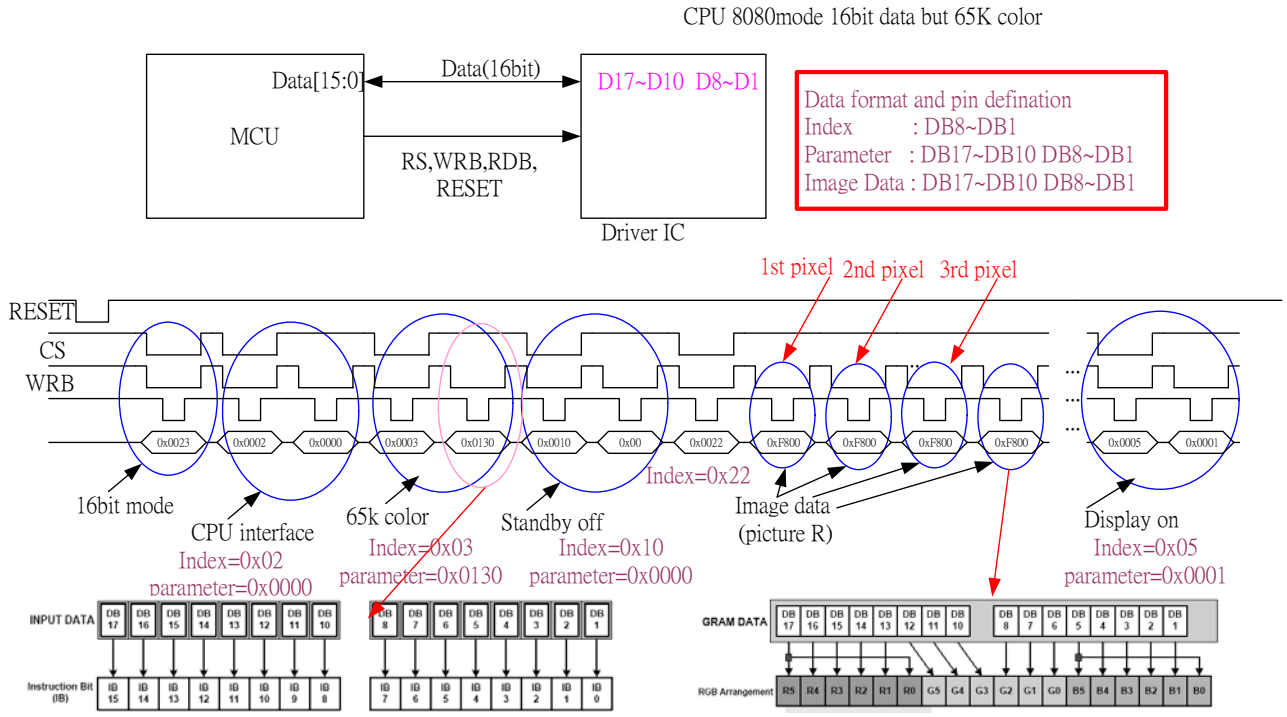
```
Index_out(0x35);
Parameter_out(V_start_address);
Index_out(0x36);
Parameter_out(V_end_address);
Index_out(0x37);
Parameter_out(Temp);
Index_out(0x20);
Parameter_out(H_start_address);
Index_out(0x21);
Parameter_out(V_start_address);
Index_out(0x22);
For(I=0;I<176*220;I++)            // 176*rgb*220 image data
  {
      Parameter_out(0x00000);   // rgb image datra      1'st
      Parameter_out(0x00000);   // rgb image datra      2'st
   }
//*************************************************
Index_out(0x05);        // display on
Parameter_out(0x0001);
```



Initial code:

Index_out(0x24);
Index_out(0x02);
Parameter_out(0x0000);
Index_out(0x03);
Parameter_out(0x4130);
Index_out(0x10);
Parameter_out(0x0000); // standby off
//******* below is gamma setting *********//
Index_out(0x70);
Parameter_out(0x????);
Index_out(0x71);
Parameter_out(0x????);
……
……
……
Index_out(0x78);
Parameter_out(0x????);
//**
H_start_address=0x20;
H_end_address=0xAF + H_start_address;
V_start_address= 0x00;
V_end_address= 0xDB;

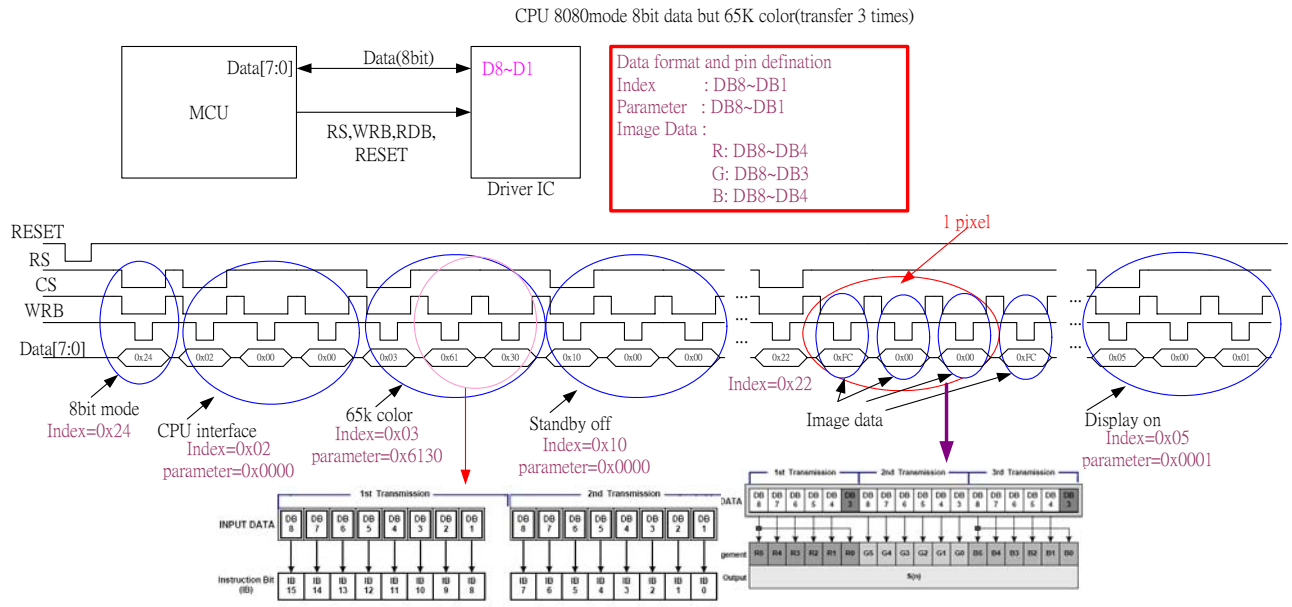
```
Temp= (H_start_address << 8) | H_end_address;
Index_out(0x35);
Parameter_out(V_start_address);
Index_out(0x36);
Parameter_out(V_end_address);
Index_out(0x37);
Parameter_out(Temp);
Index_out(0x20);
Parameter_out(H_start_address);
Index_out(0x21);
Parameter_out(V_start_address);
Index_out(0x22);
For(I=0;I<176*220;I++)            // 176*rgb*220 image data
  {
      Parameter_out(0x000);   // rgb image datra      1'st
      Parameter_out(0x000);   // rgb image datra      2'st
       Parameter_out(0x00);   // rgb image datra      3'st
   }
//*************************************************
Index_out(0x05);        // display on
Parameter_out(0x0001);
```



Initial code:

```
Index_out(0x23);
Index_out(0x02);
Parameter_out(0x0000);
Index_out(0x03);
Parameter_out(0x0130);
Index_out(0x10);              // standby off
Parameter_out(0x0000);
//******* below is gamma setting *********//
Index_out(0x70);
Parameter_out(0x????);
Index_out(0x71);
Parameter_out(0x????);
……
……
……
Index_out(0x78);
Parameter_out(0x????);
//*************************************************
H_start_address=0x20;
H_end_address=0xAF + H_start_address;
```

```
V_start_address= 0x00;
V_end_address= 0xDB;
Temp= (H_start_address << 8) | H_end_address;
Index_out(0x35);
Parameter_out(V_start_address);
Index_out(0x36);
Parameter_out(V_end_address);
Index_out(0x37);
Parameter_out(Temp);
Index_out(0x20);
Parameter_out(H_start_address);
Index_out(0x21);
Parameter_out(V_start_address);
Index_out(0x22);
For(I=0;I<176*220;I++)            // 176*rgb*220 image data
  {
      Parameter_out(0x000);    // rgb image datra 16bit     1'st
   }
//*************************************************
Index_out(0x05);        // display on
Parameter_out(0x0001);
```

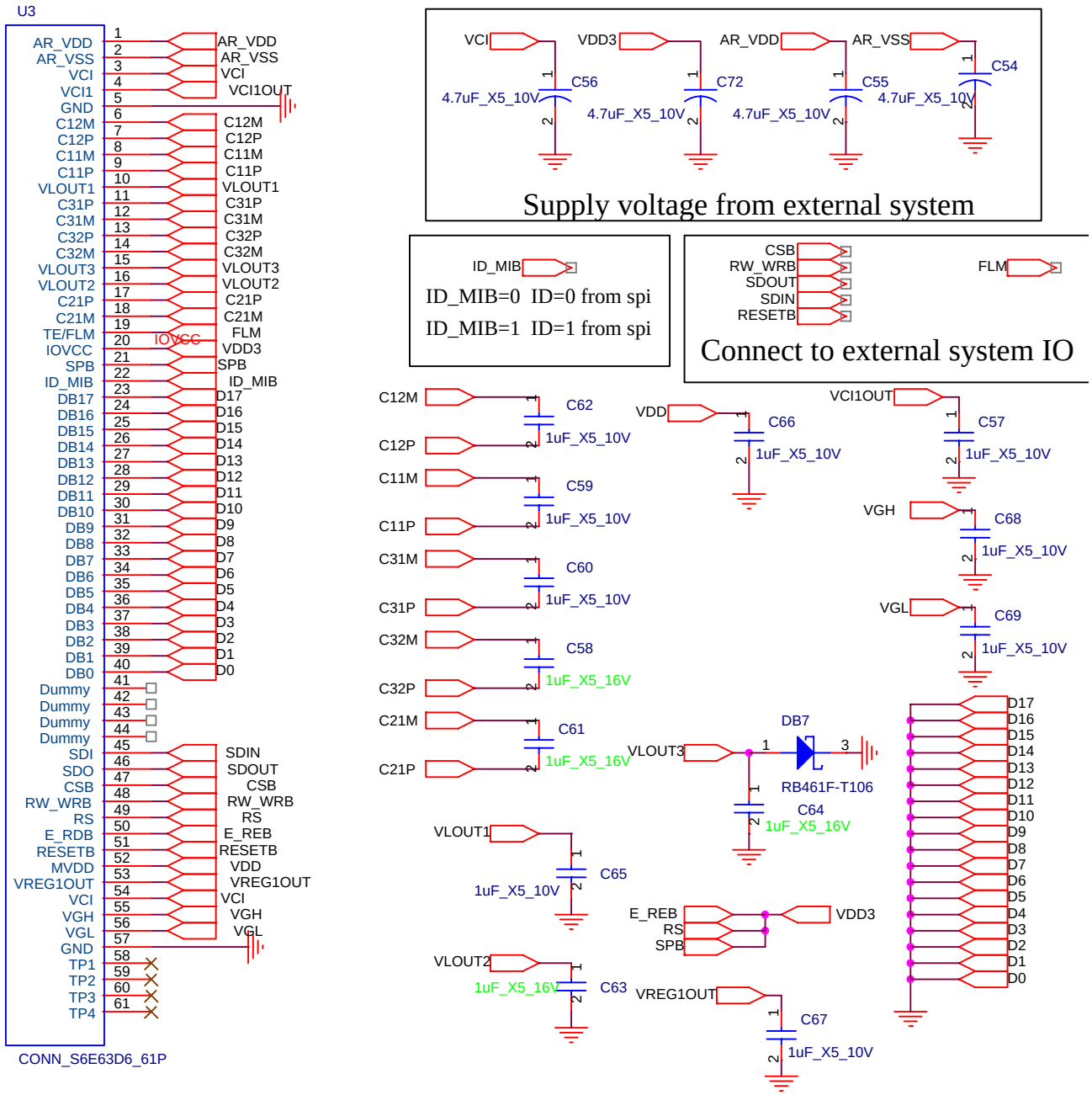


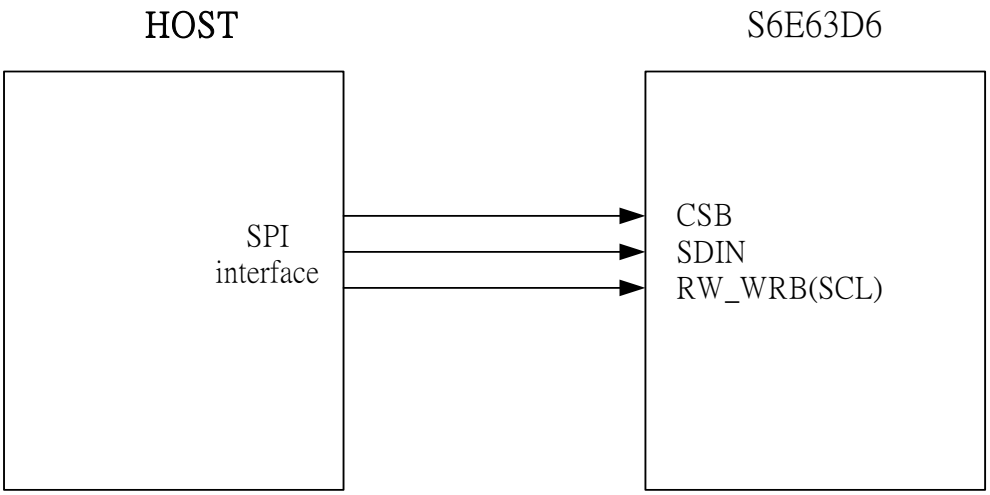
Initial code:

```
Index_out(0x24);
Index_out(0x02);
Parameter_out(0x0000);
Index_out(0x03);
Parameter_out(0x6130);
Index_out(0x10);              // standby off
Parameter_out(0x0000);
//******* below is gamma setting *********//
Index_out(0x70);
Parameter_out(0x????);
Index_out(0x71);
Parameter_out(0x????);
……
……
……
Index_out(0x78);
Parameter_out(0x????);
//*************************************************
H_start_address=0x20;
H_end_address=0xAF + H_start_address;
V_start_address= 0x00;
V_end_address= 0xDB;
Temp= (H_start_address << 8) | H_end_address;
```

```
Index_out(0x35);
Parameter_out(V_start_address);
Index_out(0x36);
Parameter_out(V_end_address);
Index_out(0x37);
Parameter_out(Temp);
Index_out(0x20);
Parameter_out(H_start_address);
Index_out(0x21);
Parameter_out(V_start_address);
Index_out(0x22);
For(I=0;I<176*220;I++)            // 176*rgb*220 image data
  {
      Parameter_out(0x000);   // rgb image datra      1'st
       Parameter_out(0x000);   // rgb image datra     2'st
       Parameter_out(0x000);   // rgb image datra     3'st
   }
//*************************************************
Index_out(0x05);        // display on
Parameter_out(0x0001);
```

1.4 Drive IC SPI interface definition



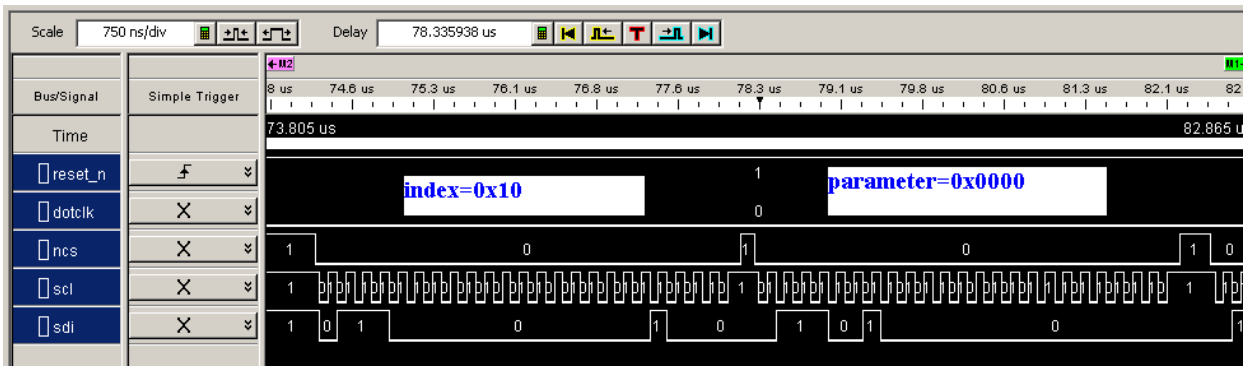


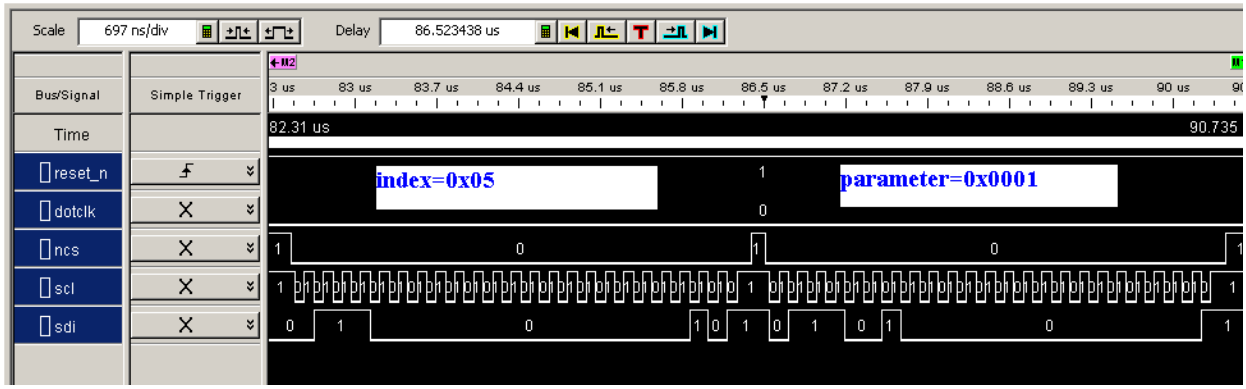
Initial code:

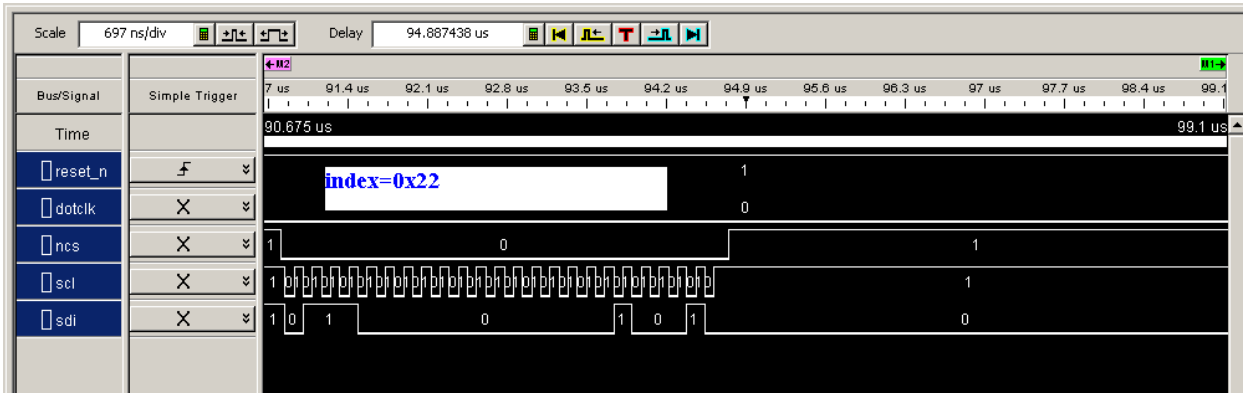
```
Index_out(0x10);
Parameter_out(0x0000);          // standby off
//******* below is gamma setting *********//
Index_out(0x70);
Parameter_out(0x????);
Index_out(0x71);
Parameter_out(0x????);
……
……
……
Index_out(0x78);
Parameter_out(0x????);
//*************************************************
H_start_address=0x20;
H_end_address=0xAF + H_start_address;
V_start_address= 0x00;
V_end_address= 0xDB;
Temp= (H_start_address << 8) | H_end_address;
Index_out(0x35);
Parameter_out(V_start_address);
Index_out(0x36);
Parameter_out(V_end_address);
Index_out(0x37);
```

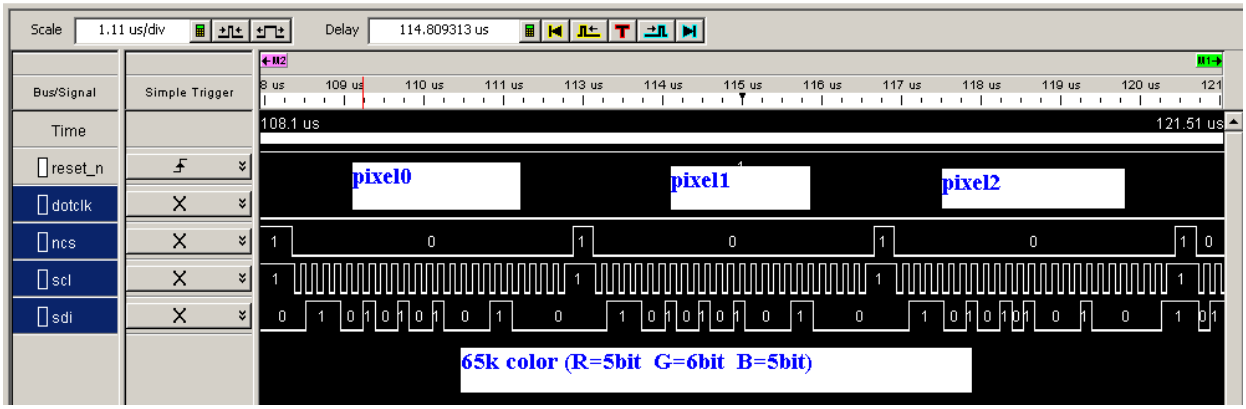
```
Parameter_out(Temp);
Index_out(0x20);
Parameter_out(H_start_address);
Index_out(0x21);
Parameter_out(V_start_address);
Index_out(0x22);
For(I=0;I<176*220;I++)            // 176*rgb*220 image data
  {
      Parameter_out(0x000);    // rgb image datra
   }
//*************************************************
Index_out(0x05);        // display on
Parameter_out(0x0001);
```

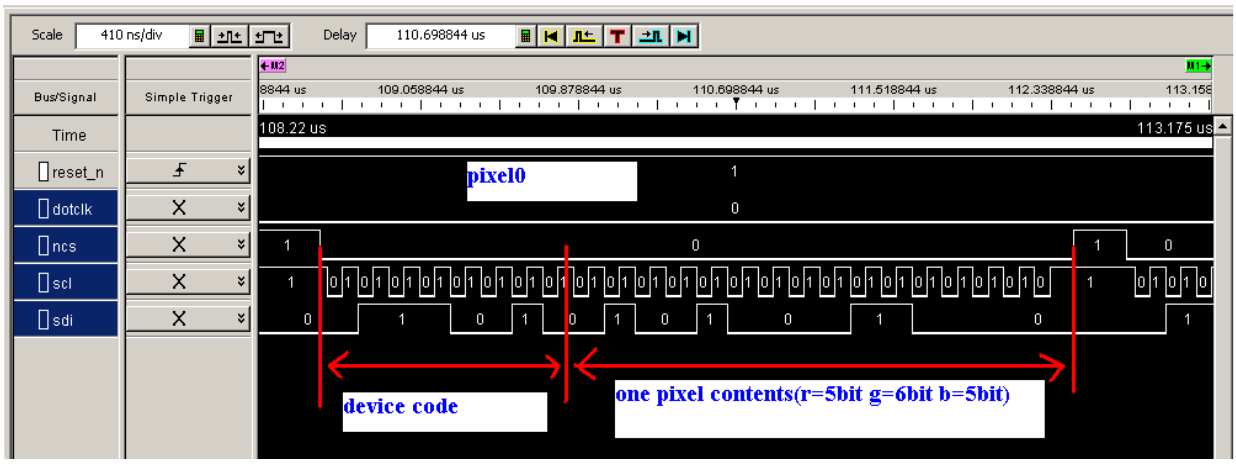
SPI waveform





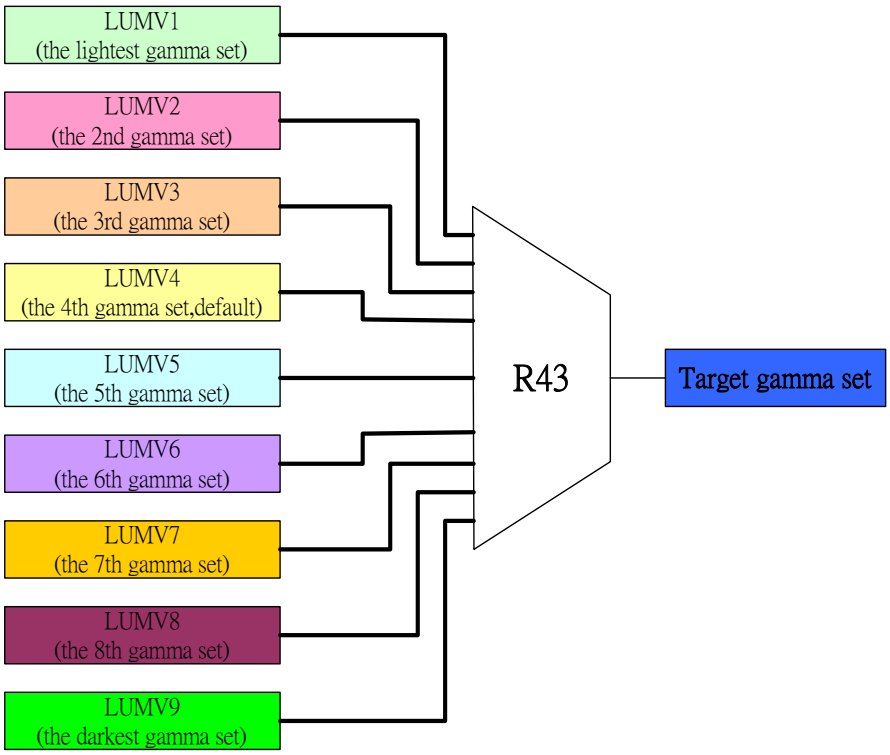





Scale
410 ns/div
Delay
110.698844 us
Bus/Signal
Simple Trigger
Time
reset_n
dotclk
ncs
scl
sdi
108.22 us
113.175 us
pixel0
device code
one pixel contents(r=5bit g=6bit b=5bit)

2. Brightness control

3.1



Switch(R43) ////// (IC behavior)
{
case 0x00: Target gamma set = LUMV1;
case 0x01: Target gamma set = LUMV2;
case 0x02: Target gamma set = LUMV3;
case 0x03: Target gamma set = LUMV4;
case 0x04: Target gamma set = LUMV5;
case 0x05: Target gamma set = LUMV6;
case 0x06: Target gamma set = LUMV7;
case 0x07: Target gamma set = LUMV8;
case 0x08: Target gamma set = LUMV9;
}

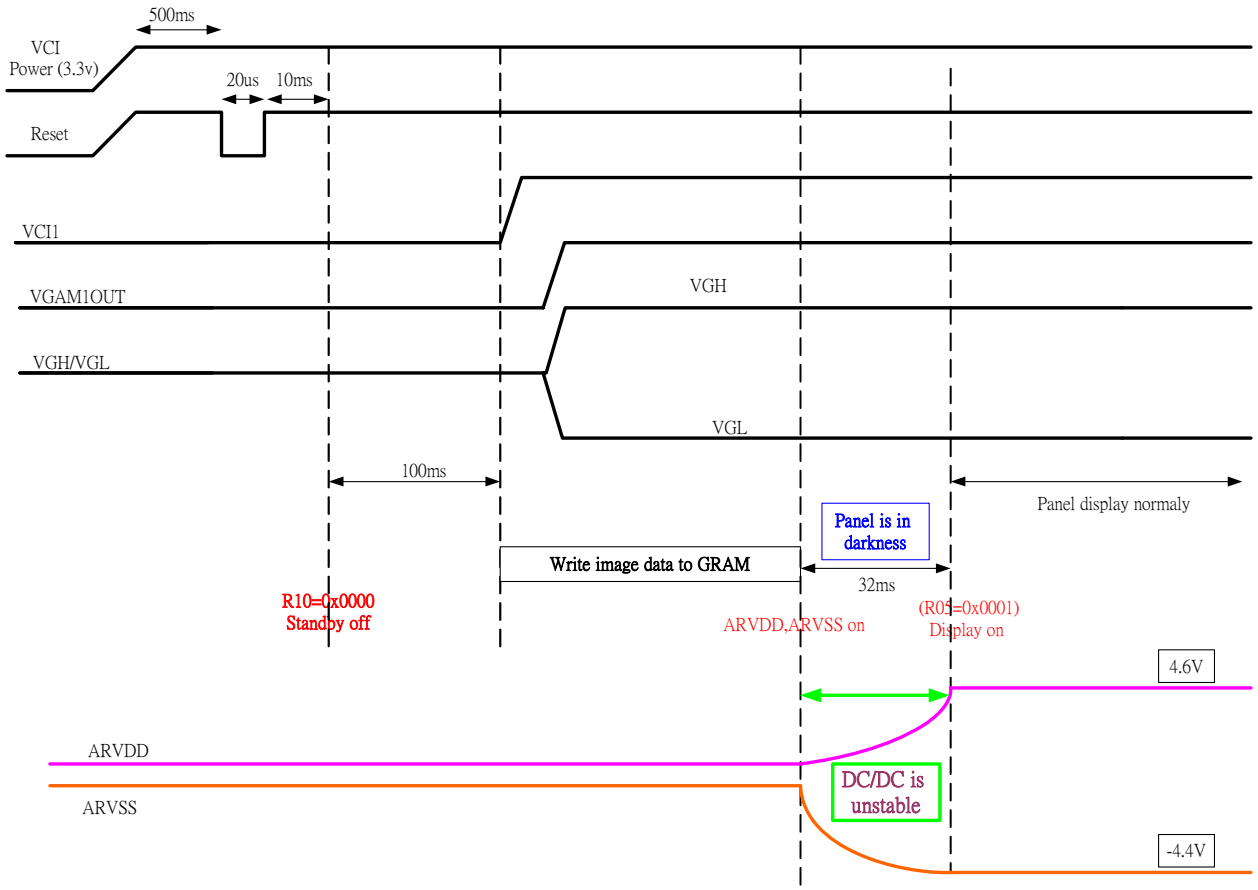
3.2 Program different gamma set:

Set R80=01 , then write R70 ~ R78 to register. The data will store in LUMV1 gamma set.
Set R80=02 , then write R70 ~ R78 to register. The data will store in LUMV2 gamma set.
Set R80=03 , then write R70 ~ R78 to register. The data will store in LUMV3 gamma set.
Set R80=04 , then write R70 ~ R78 to register. The data will store in LUMV4 gamma set.
Set R80=05 , then write R70 ~ R78 to register. The data will store in LUMV5 gamma set.

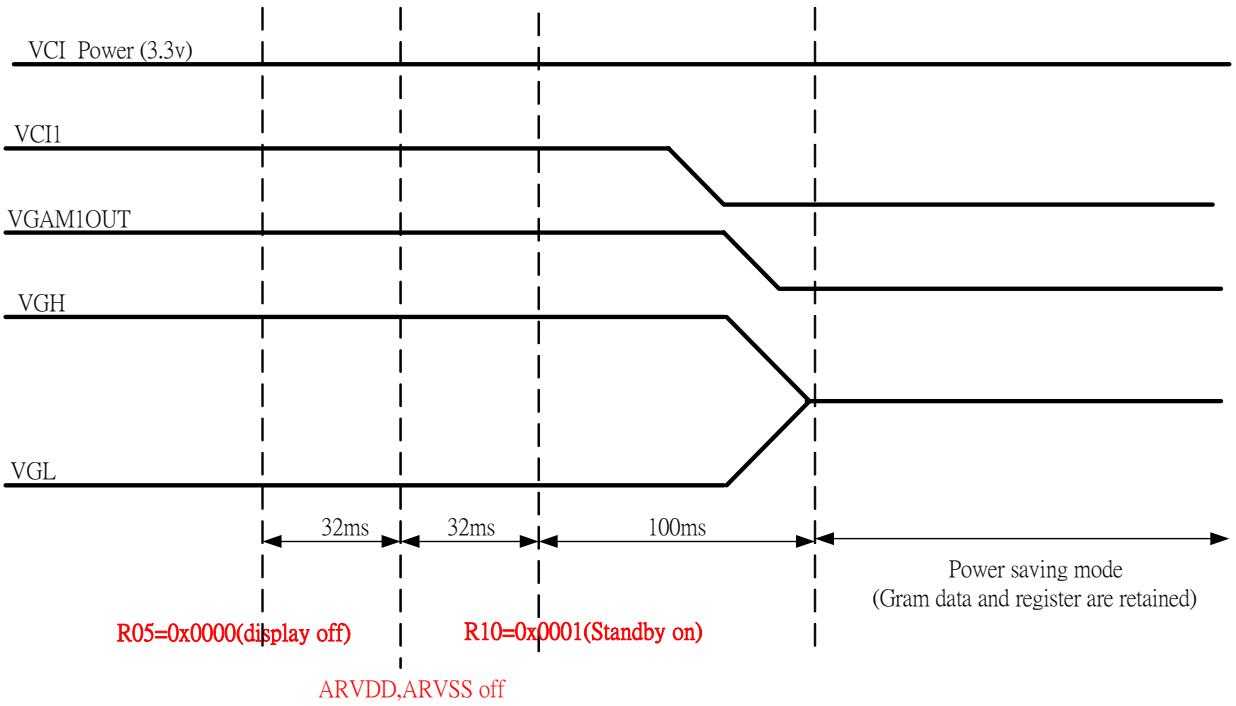
Set R80=06 , then write R70 ~ R78 to register. The data will store in LUMV6 gamma set.
Set R80=07 , then write R70 ~ R78 to register. The data will store in LUMV7 gamma set.
Set R80=08 , then write R70 ~ R78 to register. The data will store in LUMV8 gamma set.
Set R80=09 , then write R70 ~ R78 to register. The data will store in LUMV9 gamma set.

3. Power Sequence

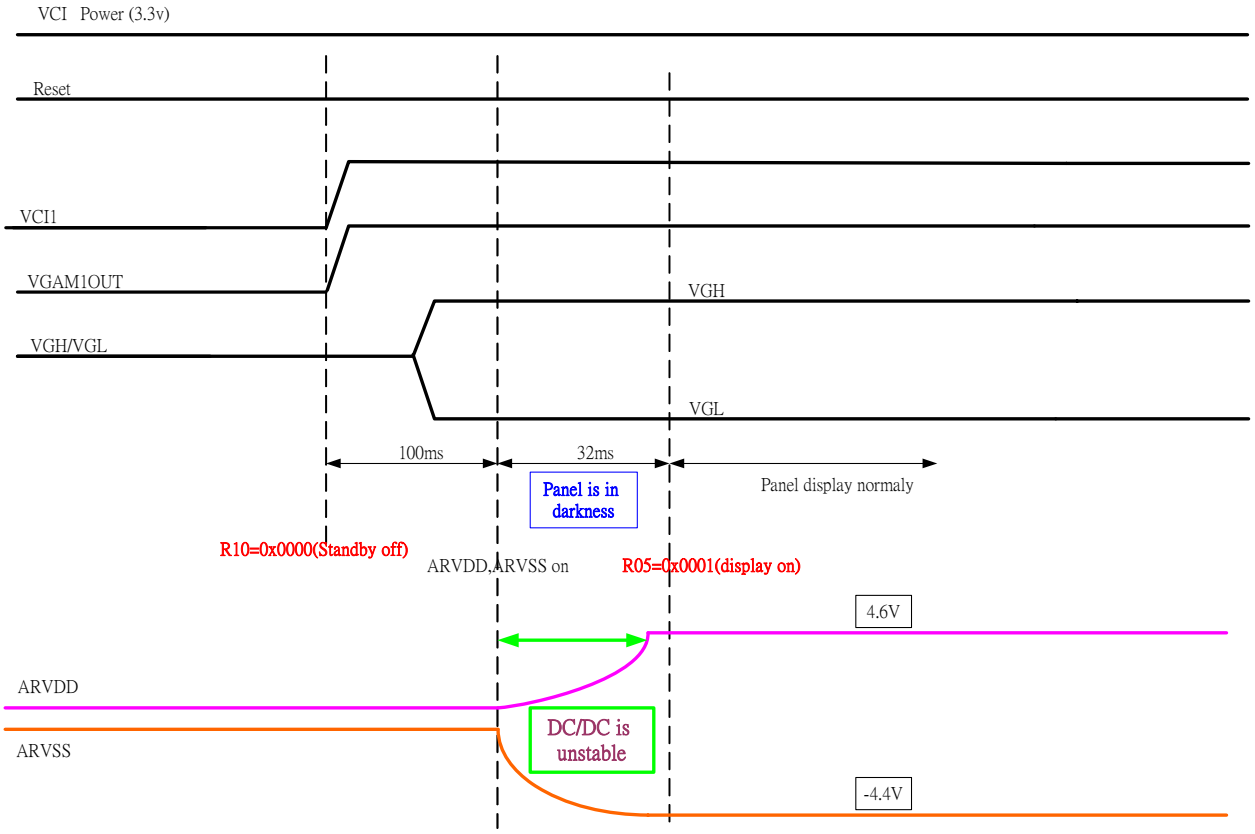
4.1 Power on sequence:



4.2 Standby on sequence:



4.3 Standby off sequence:



4.4 Power off sequence: