

Interfacing mDDR and DDR2 Memories with i.MX25

by *Multimedia Applications Division*
Freescale Semiconductor, Inc.
Austin, TX

This application note shows the interface between the i.MX25 processor and the mDDR and DDR2 memories. Also, this application note defines the routing guidelines for these two memories, with pictures and examples.

1 i.MX25 SDRAM Controller

The SDRAM controller can interface with either SDR-SDRAM, Mobile DDR, or DDR2-SDRAM memories.

The MX25 DDR controller has the following signals for memory interface.

- Data bus and its buffer control signals
 - SD0 - SD15
 - DQS0 - DQS1
 - DQM0 - DQM1
- Address bus and its bank control signals
 - A0 - A9, A11 - A12
 - SDBA0 - SDBA1
 - MA10
- Control
 - RAS

Contents

1. i.MX25 SDRAM Controller	1
2. i.MX25 Memory Interface	4
2.1. mDDR Memory	4
2.2. DDR2 Memory	4
2.3. mDDR and DDR2 JTAG Script Configuration	5
3. mDDR and DDR2 Routing Guidelines	9
4. mDDR and DDR2 Layout	10
4.1. mDDR Signal	10
4.2. DDR2 Signal	12
5. Revision History	13

i.MX25 SDRAM Controller

- CAS
- SDCKE0
- SDWE
- CDS0
- Clock
 - SDCLK
 - SDCLK_B

The DDR2 JEDEC standard includes differential signals for DQS, but the i.MX25 processor does not include these DQS signals. Therefore, the DQS signals must be handled separately and these signals require a termination connection.

Figure 1 shows the signals related to the i.MX25 DDR. The SDCLK and SDCLK_B 22 Ω series resistors are also shown.

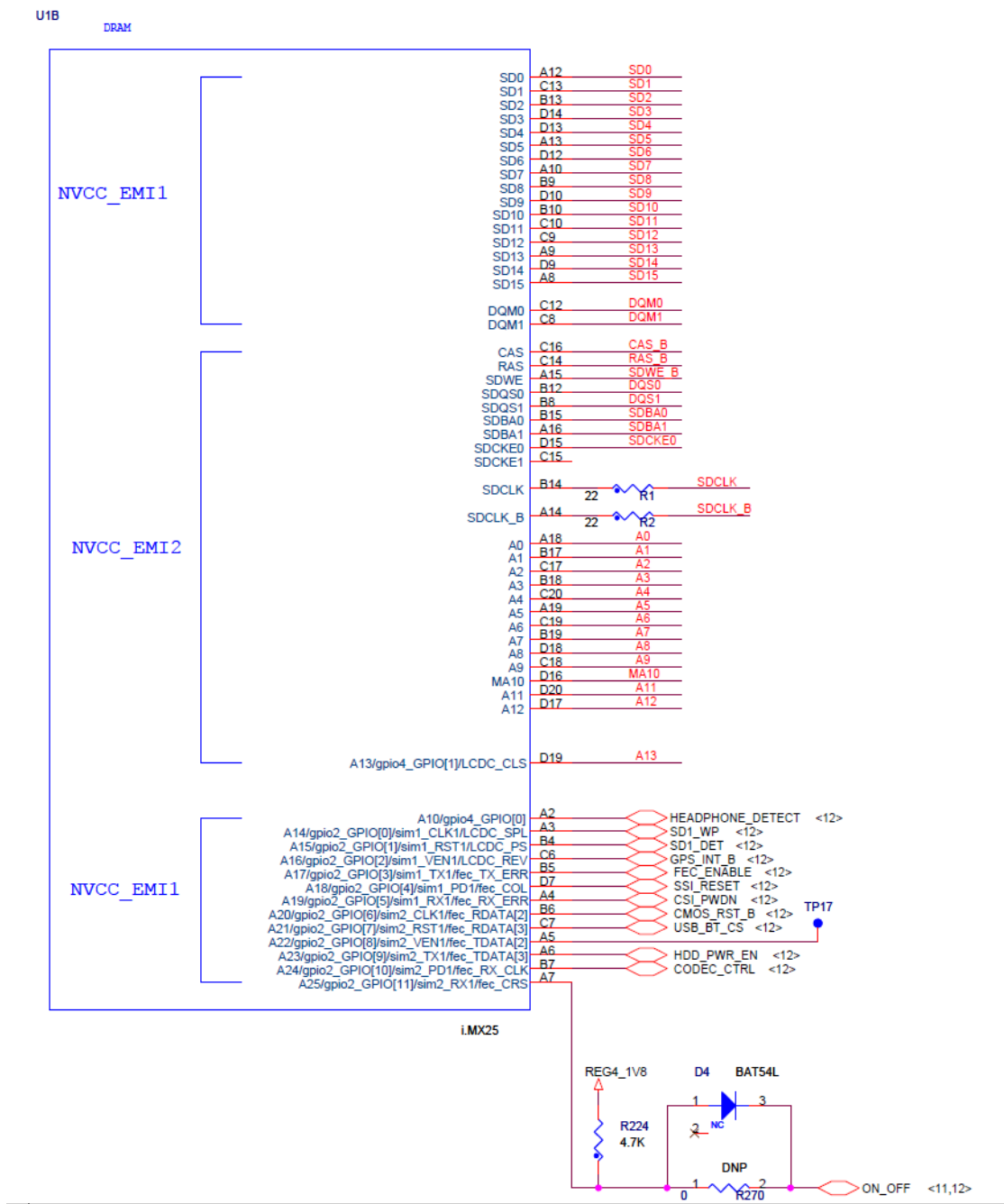


Figure 1. i.MX25 DDR Interface

2 i.MX25 Memory Interface

2.1 mDDR Memory

Figure 2 shows the mDDR memory connection. Here, the mDDR device is the HYB18M512160AF.

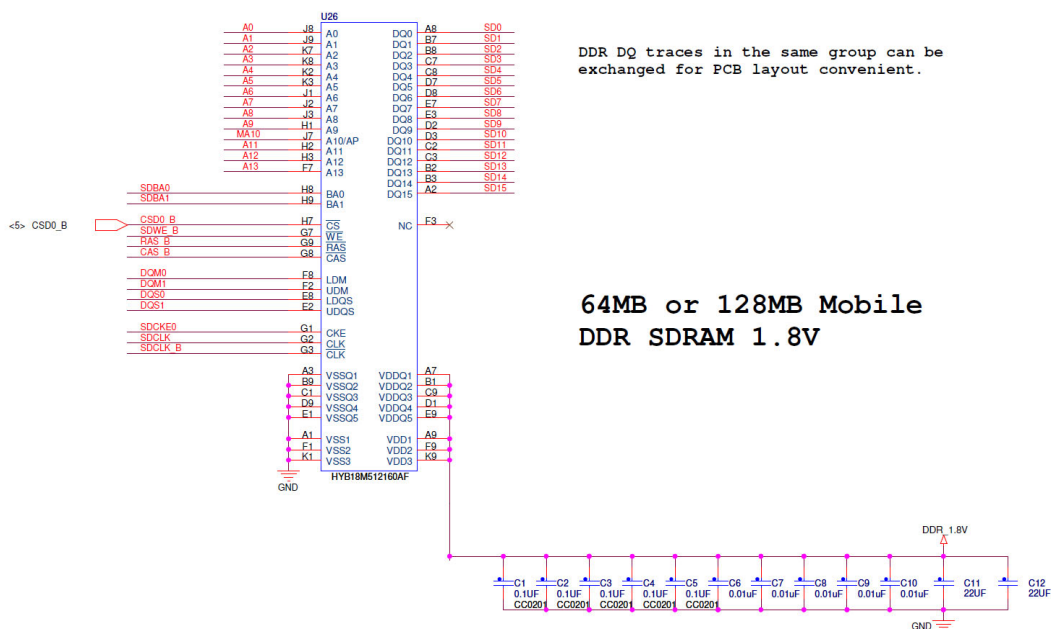


Figure 2. mDDR Memory Connection

2.2 DDR2 Memory

The difference between the DDR2 and the mDDR memories are the ODT and VREF signals. Also, in Figure 3, it is seen that the /DQS signals are tied to ground with a 1 KΩ resistor. Here, the DDR2 device is the K4T51163QE.

Figure 3 shows the DDR 2 memory connection.

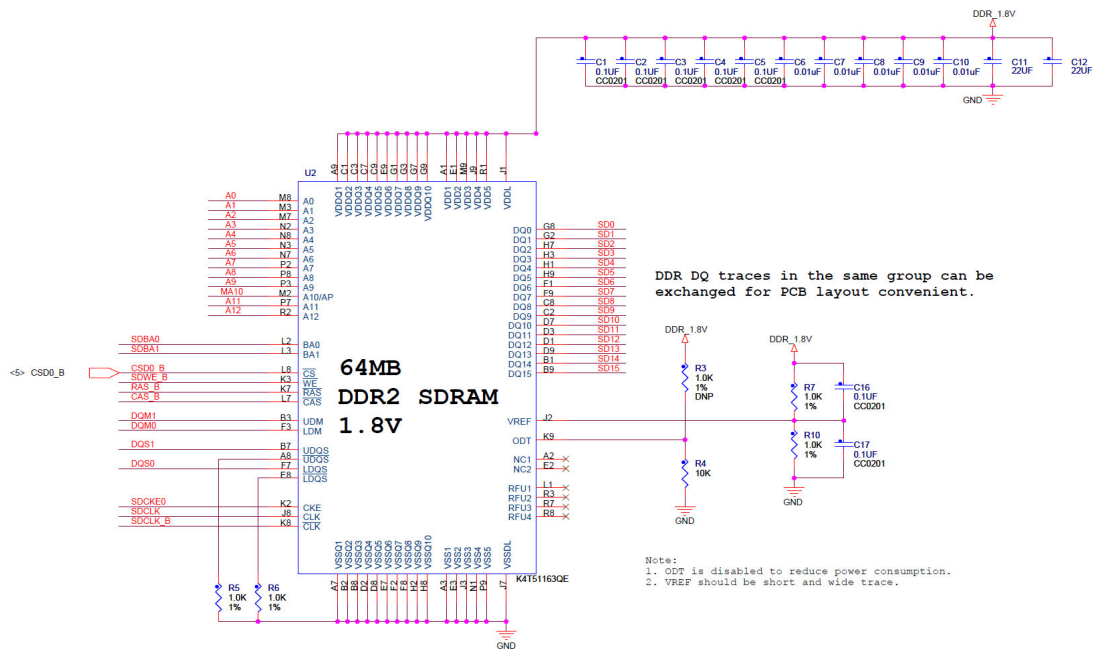


Figure 3. DDR2 Memory Connection

2.3 mDDR and DDR2 JTAG Script Configuration

The following examples show the configuration of mDDR and DDR2 memories for the i.MX25.

The code for configuring mDDR memory is as follows:

```

/*=====
/* Copyright (C) 2008, Freescale Semiconductor, Inc. All Rights Reserved
/* THIS SOURCE CODE IS CONFIDENTIAL AND PROPRIETARY AND MAY NOT
/* BE USED OR DISTRIBUTED WITHOUT THE WRITTEN PERMISSION OF
/* Freescale Semiconductor, Inc.
/*=====
// Initialization script for Senna mDDR PDK
/*=====
// Revision History:
//
// Author (core ID)      Modification      Tracking      Description of Changes
// -----
// Platform Diagnostics  21-Jul-2008      v2            File created from MX31 init file.
// Platform Diagnostics  05-Aug-2008      v2            Added CLKO options
/*=====
wait = on
/*=====
// init_ccm - base freq = 532, configuring post dividers
/*=====
//
//
/*=====
// Configure WEIM
/*=====
//

```

i.MX25 Memory Interface

```

/*=====
// Configure CPLD
/*=====
//
//
/*=====
// Initialization script for 16-bit mDDR on MX25 PDK
/*=====
wait = on
reset
// ESDMISC
// enable mDDR mode (MDDR_EN = 1)
setmem /32 0xB8001010 =0x00000004
// ESDRAMC timings - optimize later
//setmem /32 0xB8001004 =0x006ac73a
// DDR initialization
// enable CS0 precharge command
setmem /32 0xB8001000 =0x92100000
// precharge all dummy write only address matter
setmem /32 0x80000400 =0x12344321
// enable CS0 Auto-Refresh command
setmem /32 0xB8001000 =0xa2100000
// two refresh command dummy write only address matter
setmem /32 0x80000000 =0x12344321
setmem /32 0x80000000 =0x12344321
// enable CS0 Load Mode Register command
setmem /32 0xB8001000 =0xb2100000
// MODE register - CAS=3, BL=8
// dummy write only address matter
setmem /8 0x80000033 =0xda
// Extended MODE register
// dummy write only address matter
setmem /8 0x81000000 =0xff
// Configure EDSRAMC for normal mode
// ROW: 13, COL: 10, DSIZ: 16bit D[15:0], refresh: 8192cycles/64ms
//setmem /32 0xB8001000 =0x82216080
// uncomment to enable power down time-out field PWDT to 64 clocks, most aggressive setting
setmem /32 0xB8001000 =0x82216880
// dummy write to set DQSlow (needed in MX31, not sure about MX25)
// setmem /32 0x80000000 =0x0000
// @@ configure the data abort not to be precise
setreg @CPSR_A=0
// configure AIPS1
setmem /32 0x43F00040 =0x0 // AIPS1_OPACR0_7
setmem /32 0x43F00044 =0x0 // AIPS1_OPACR8_15
setmem /32 0x43F00048 =0x0 // AIPS1_OPACR16_23
setmem /32 0x43F0004C =0x0 // AIPS1_OPACR24_31
setmem /32 0x43F00050 =0x0 // AIPS1_OPACR32_33
setmem /32 0x43F00000 =0x77777777 // AIPS1_MPROT0_7
setmem /32 0x43F00004 =0x77777777 // AIPS1_MPROT8_15
// configure AIPS2 - keep for now, may need to modify based on MX25
setmem /32 0x53F00040 =0x0 // AIPS2_OPACR0_7
setmem /32 0x53F00044 =0x0 // AIPS2_OPACR8_15
setmem /32 0x53F00048 =0x0 // AIPS2_OPACR16_23
setmem /32 0x53F0004C =0x0 // AIPS2_OPACR24_31
setmem /32 0x53F00050 =0x0 // AIPS2_OPACR32_33
setmem /32 0x53F00000 =0x77777777 // AIPS2_MPROT0_7
setmem /32 0x53F00004 =0x77777777 // AIPS2_MPROT8_15

```

```

// configure CPLD on CS5
setmem /32 0xB8002050 =0x0000D843 // CS5_CSCRU
setmem /32 0xB8002054 =0x22252521 // CS5_CSCRL
setmem /32 0xB8002058 =0x22220A00 // CS5_CSCRA
//*=====
// init_ccm - base freq = 400, configuring post dividers
//*=====
setmem /32 0x53F80008 = 0x20034000 // ARM clk = 399, AHB clk = 133
// CLKO options, uncomment one
// on 3DS, CLKO available on R281
//setmem /32 0x53f80064 = 0x43300000// AHB clk, div-by-4
//setmem /32 0x53f80064 = 0x47200000// ARM clk, div-by-8
//setmem /32 0x53f80064 = 0x40000000// CLK32K straight//
readfile, raw, gui "X:\redboot\diag-ecos-rel\REDBOOT\bin\mx25_3stack_redboot.bin"=0x87F00000
setreg @R15=0x87F00000
    
```

The code for configuring DDR2 memory is as follows:

```

;AIPS settings
setmem /32 0x43f00040 =0x00000000
setmem /32 0x43f00044 =0x00000000
setmem /32 0x43f00048 =0x00000000
setmem /32 0x43f0004C =0x00000000
setmem /32 0x43f00050 =0x00000000
setmem /32 0x43f00000 =0x77777777
setmem /32 0x43f00004 =0x77777777
setmem /32 0x53f00040 =0x00000000
setmem /32 0x53f00044 =0x00000000
setmem /32 0x53f00048 =0x00000000
setmem /32 0x53f0004C =0x00000000
setmem /32 0x53f00050 =0x00000000
setmem /32 0x53f00000 =0x77777777
setmem /32 0x53f00004 =0x77777777
;cs0
setmem /32 0xB8002008 = 0x00010000
setmem /32 0xB8002004 = 0x00210511
setmem /32 0xB8002000 = 0x00000200
;cs5
setmem /32 0xB8002054 = 0x00000001
setmem /32 0xB8002050 = 0x00000200
;cs1
setmem /32 0xB8002018 = 0x00010000
setmem /32 0xB8002010 = 0x00000200
setmem /32 0xB8002014 = 0x00210511
; ; ;
setmem /32 0x53f80008 =0x2003C000
setmem /32 0x43fac454 = 0x1000
;cs4
setmem /32 0xB8002040 = 0x0000DCF6
setmem /32 0xB8002044 = 0x444A4541
setmem /32 0xB8002048 = 0x44443302
; Set ODT = 0
setmem /32 0xB400000C =0x0400
;setmem /32 0xB8002008 =0x0
;setmem /32 0xB8002004 =0x00200501
;setmem /32 0xB8002000 =0x00000800
setmem /32 0xd8002000 =0x0000CC03
setmem /32 0xd8002004 =0xa0330D01
    
```

i.MX25 Memory Interface

```

setmem /32 0xd8002008 =0x00220800
;DDR2 initial begin
;setmem /32 0xB8001004=0x007ffcff
setmem /32 0xB8001004=0x0076E83a
; ESD_MISC
;setmem /32 0xB8001010=0x0000020C
; setmem /32 0xB8001018 =0x0000004C
setmem /32 0xB8001010=0x00000204
; ESD_ESDCTL0
DE_SMODE_SP_ROW_00_COL_00_DSIZ_SREFR_0_PWDT_0_FP_BL_0__PRCT
; ESD_ESDCTL0 32'b1_001_0_010_00_01_00_00_000_0_00_0_0_0_0_00000
; enable CS0 precharge command
setmem /32 0xB8001000=0x92210000
; precharge all dummy write only address matter
setmem /32 0x80000f00=0x12344321
; ESD_ESDCTL0: select Load-Mode-Register mode
setmem /32 0xB8001000 =0xB2210000
; DDR2: Load reg EMR2
setmem /8 0x82000000 =0xda
; DDR2: Load reg EMR3
setmem /8 0x83000000 =0xda
; DDR2: Load reg EMR1 -- enable DLL
setmem /8 0x81000400 =0xda
; DDR2: Load reg MR -- reset DLL
setmem /8 0x80000333 =0xda
; ESD_ESDCTL0: select Precharge-All mode
setmem /32 0xB8001000 =0x92210000
; DDR2: Precharge-All
setmem /8 0x80000400 =0x12345678
; ESD_ESDCTL0: select Manual-Refresh mode
setmem /32 0xB8001000 =0xA2210000
; DDR2: Manual-Refresh 2 times
setmem /32 0x80000000 =0x87654321
setmem /32 0x80000000 =0x87654321
; ESD_ESDCTL0: select Load-Mode-Register mode
setmem /32 0xB8001000 =0xB2210000
; DDR2: Load reg MR -- CL=3, BL=8, end DLL reset
setmem /8 0x80000233 =0xda
; DDR2: Load reg EMR1 -- OCD default
setmem /8 0x81000780 =0xda
; DDR2: Load reg EMR1 -- OCD exit
setmem /8 0x81000400 =0xda
; ESD_ESDCTL0
DE_SMODE_SP_ROW_00_COL_00_DSIZ_SREFR_0_PWDT_0_FP_BL_0__PRCT
; ESD_ESDCTL0 32'b1_000_0_010_00_10_00_10_011_0_00_0_0_0_0_000000
; @; normal mode row=010//col=10//dzize=10//self ref=011//PWDT =00//BL =0//prct =000000
setmem /32 0xB8001000=0x82216080
;-----
; Init IOMUXC_SW_PAD_CTL_GRP_DDRTYPE_GRP(1-5)
;-----
setmem /32 0x43FAC454 =0x00001000
;comment clock gating open
setmem /32 0x53F8000C =0xffffffff
setmem /32 0x53F80010 =0xffffffff
setmem /32 0x53F80014 =0xffffffff
readfile, raw, gui

```



```
x:\redboot\diag-ecos-rel\REDBOOT\bin\mx25_T01_1_3stack_redboot.bin"=0x83F00000
setreg @R15=0x83F00000
```

3 mDDR and DDR2 Routing Guidelines

Routing is one of the critical items of this interface. The following routing guidelines are very restrictive, but they result in a better and more robust design. The difficulty in meeting these guidelines depends on the stack up.

The DDR routing is performed in the following ways:

- Routing all the signals of the same length
- Routing all the signals by byte-group

The first approach of routing all the signals of the same length is more difficult, but this approach serves as a better and easy way for analysis purpose.

Table 1 shows the first approach for DDR routing.

Table 1. Option 1 for DDR Routing

Signals	Length	Considerations
Address and Bank	Clock length	Match the signals ± 20 mils
Data and Buffer	Clock length	
Control signals	Clock length	
Clock	Lcritical	Match the signals of clocks signals ± 5 mils

The second approach of routing all the signals by byte-group requires a better control of the signals of each group, and this approach is a little more difficult for analysis and constraint settings.

Table 2 shows the second approach for DDR routing.

Table 2. Option 2 for DDR Routing

Signals	Length	Considerations
Address and Bank	Clock length	Match the signals ± 20 mils
Byte Group 1 DQ0-DQ7, DQS0, DQM0	The max byte group 1 length Clock length	Match the signals of each byte-group ± 20 mils The difference between the byte-groups must be ± 50 mils
Byte Group 2 DQ8-DQ15, DQS1, DQM1	The max byte group 2 length Clock length	
Byte Group 3 DQ16-DQ23, DQS2, DQM2	The max byte group 3 length Clock length	
Byte Group 4 DQ24-DQ31, DQS3, DQM3	The max byte group 4 length Clock length	
Control signals	Clock length	Match the signals ± 20 mils
Clock	Lcritical	Match the signals of clocks signals ± 5 mils

4 mDDR and DDR2 Layout

This section shows the PDK (Platform Development Kit) implementation of the routing of both the mDDR and DDR2 signals.

4.1 mDDR Signal

For the mDDR option, the stack up has ten layers. The layers used for the mDDR routing are as follows:

Figure 4 shows the top routing.

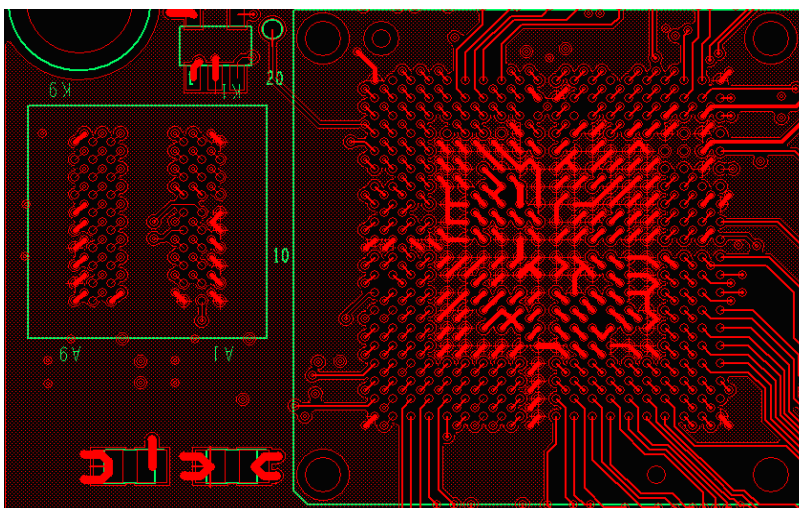


Figure 4. Top Routing

Figure 5 shows the internal 1 routing.

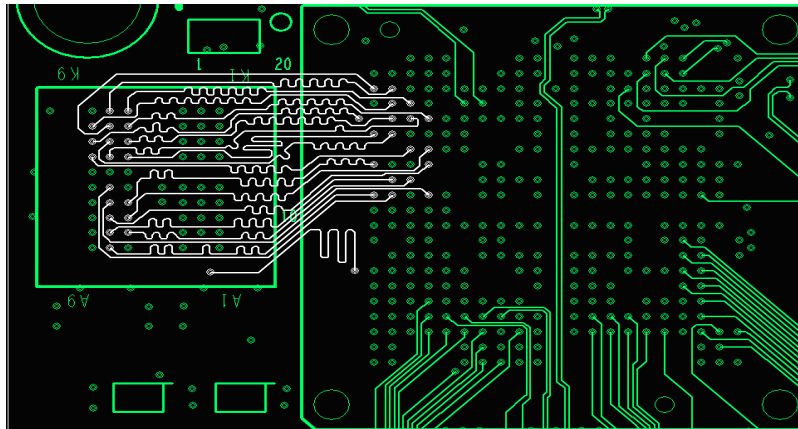


Figure 5. Internal 1 Routing

Figure 6 shows the internal 2 routing.

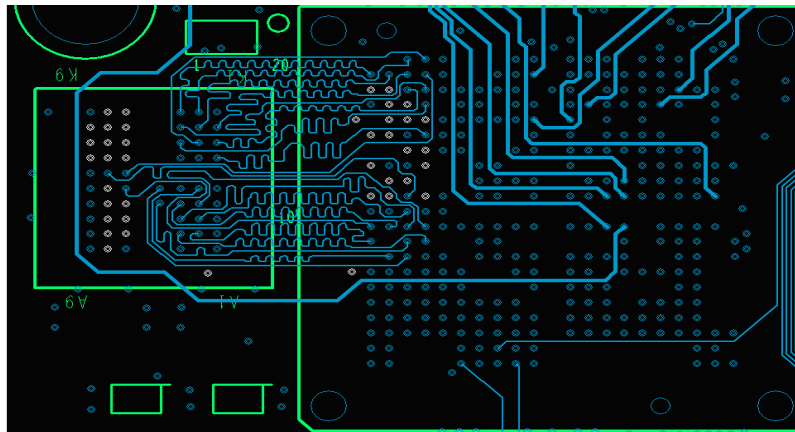


Figure 6. Internal 2 Routing

Figure 7 shows the bottom routing.

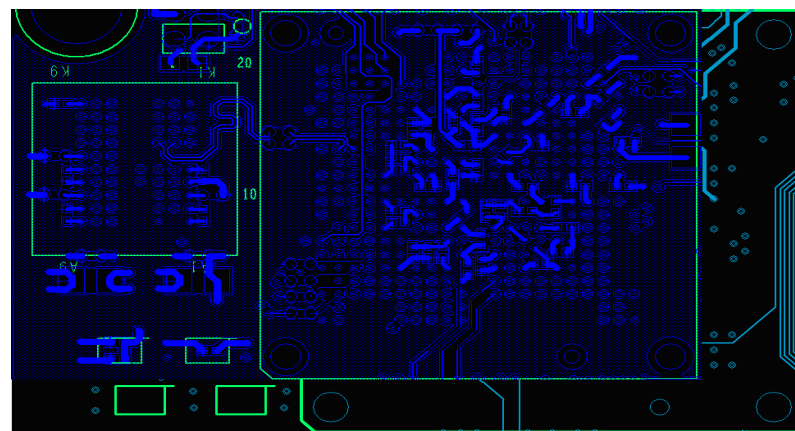


Figure 7. Bottom Routing

4.2 DDR2 Signal

For the DDR2 option the stack up also has ten layers. The layers used for the DDR2 routing are as follows:

Figure 8 shows the top routing.

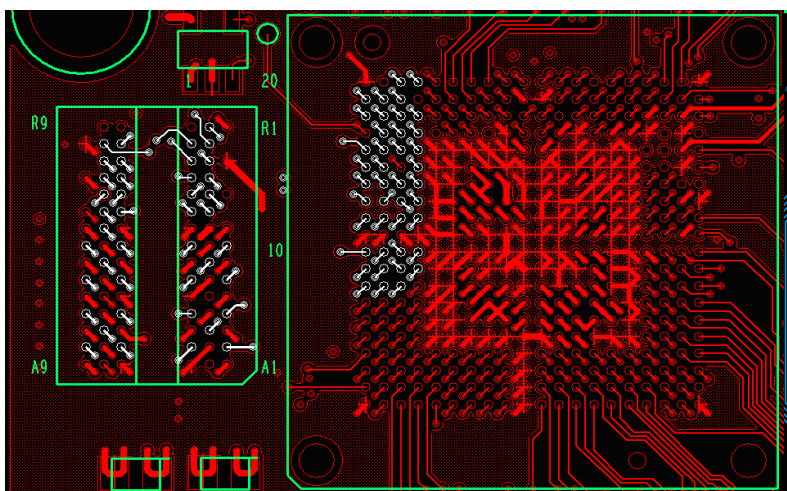


Figure 8. Top Routing

Figure 9 shows the internal 1 routing.

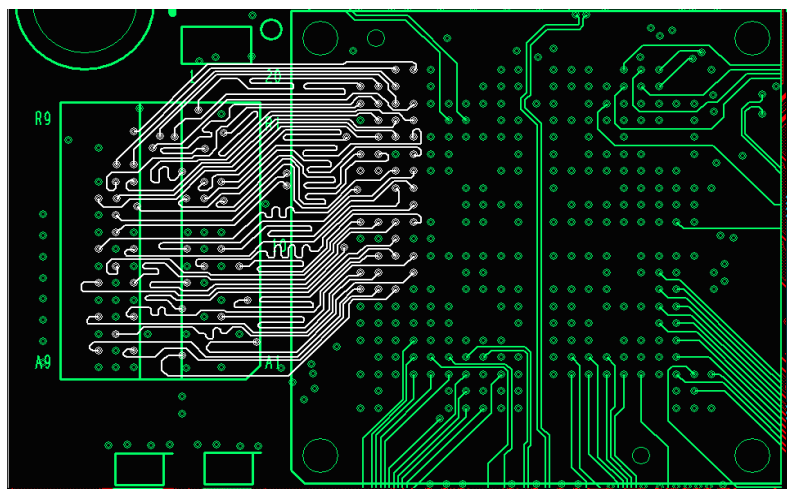


Figure 9. Internal 1 Routing

Figure 10 shows the bottom routing.

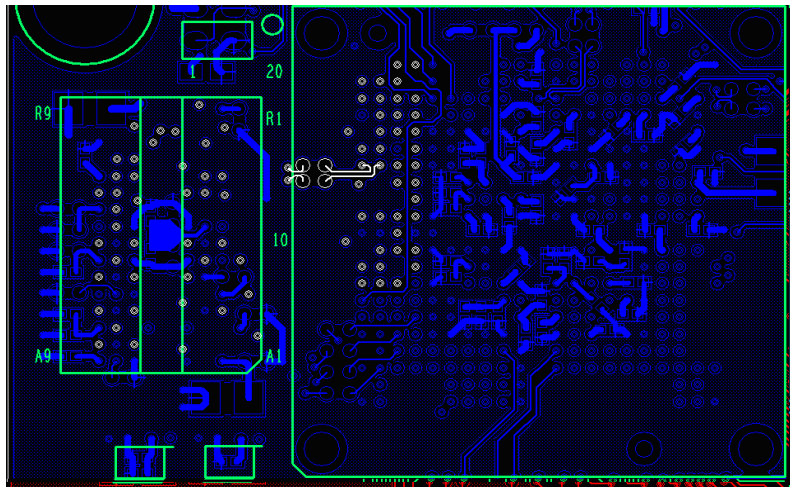


Figure 10. Bottom Routing

The stack up used for the mDDR and DDR2 options are as follows:

- TOP
- GND1
- INT 1
- INT 2
- PWR1
- PWR2
- INT3
- INT4
- GND2
- Bottom

Finally, it is required to have point-to-point connections and impedance for the signals. The impedance must be 50 Ω for singled ended and 100 Ω for differential pairs.

5 Revision History

Table 3 provides a revision history for this application note.

Table 3. Document Revision History

Rev. Number	Date	Substantive Change(s)
0	03/2010	Initial release

THIS PAGE INTENTIONALLY LEFT BLANK

THIS PAGE INTENTIONALLY LEFT BLANK

How to Reach Us:

Home Page:

www.freescale.com

Web Support:

<http://www.freescale.com/support>

USA/Europe or Locations Not Listed:

Freescale Semiconductor, Inc.
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
1-800-521-6274 or
+1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:

Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:

Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku
Tokyo 153-0064
Japan
0120 191014 or
+81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:

Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
support.asia@freescale.com

For Literature Requests Only:

Freescale Semiconductor
Literature Distribution Center
1-800 441-2447 or
+1-303-675-2140
Fax: +1-303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters which may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

Freescale, the Freescale logo, CodeWarrior, ColdFire, PowerQUICC, StarCore, and Symphony are trademarks of Freescale Semiconductor, Inc. Reg. U.S. Pat. & Tm. Off. CoreNet, QorIQ, QUICC Engine, and VortiQa are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. ARM is the registered trademark of ARM Limited.

© 2010 Freescale Semiconductor, Inc.

