The LPC24xx External Memory Controller (EMC) is an ARM PrimeCell™ MultiPort Memory Controller peripheral offering support for asynchronous static memory devices such as RAM, ROM and Flash, as well as dynamic memories such as Single Data Rate SDRAM. The EMC is an Advanced Microcontroller Bus Architecture (AMBA) compliant peripheral. The purpose of this document is to help the developer in the design of a system using SRAM and SDRAM memories.

1. Pin description

Name	Description	Type
A[23:0]	External memory address output. Used for both static and SDRAM memories. SDRAM memories use only bits [14:0].	Output
D[31:0]	External memory data lines. These are inputs when data is read from external memory and outputs when data is written to external memory.	Input/Output
/OE	Low active output enable for static memories.	Output
/BLS[3:0]	Low active byte lane selects. Used for static memories.	Output
/WE	Low active write enable. Used for SDRAM and static memories.	Output
/CS[3:0]	Static memory chip selects. Default active LOW. Used for static memories.	Output
/DYCS[3:0]	SDRAM chip selects. Used for SDRAM memories.	Output
/CAS	Column address strobe. Used for SDRAM memories.	Output
/RAS	Row address strobe. Used for SDRAM memories.	Output
CLKOUT[3:0]	SDRAM clocks. Used for SDRAM memories.	Output
CKEOUT[3:0]	SDRAM clock enables. Used for SDRAM memories. One is allocated for each Chip Select.	Output
DQMOUT[3:0]	Data mask output to SDRAMs. Used for SDRAM memories.	Output

1.1 Pin connect block

The pin connect block allows selected pins of the microcontroller to have more than one function. Configuration registers control the multiplexers to allow connection between the pin and the on chip peripherals.
Peripherals should be connected to the appropriate pins prior to being activated, and prior to any related interrupt(s) being enabled. Activity of any enabled peripheral function that is not mapped to a related pin should be considered undefined.

Pin Function Select Register 4 (PINSEL4 - 0xE002 C010)

PINSEL4	Pin name	Function when 00	Function when 01	Function when 10	Function when 11	Reset value
29:28	P2[14]	GPIO Port 2.14	/CS2	CAP2[0]	SDA1	00
31:30	P2[15]	GPIO Port 2.15	/CS3	CAP2[1]	SCL1	00

Pin Function Select Register 5 (PINSEL5 - 0xE002 C014)

PINSEL5	Pin name	Function when 00	Function when 01	Function when 10	Function when 11	Reset value
1:0	P2[16]	GPIO Port 2.16	/ CAS	Reserved	Reserved	00
3:2	P2[17]	GPIO Port 2.17	/ RAS	Reserved	Reserved	00
5:4	P2[18]	GPIO Port 2.18	CLKOUT0	Reserved	Reserved	00
7:6	P2[19]	GPIO Port 2.19	CLKOUT1	Reserved	Reserved	00
9:8	P2[20]	GPIO Port 2.20	/ DYCS0	Reserved	Reserved	00
11:10	P2[21]	GPIO Port 2.21	/ DYCS1	Reserved	Reserved	00
13:12	P2[22]	GPIO Port 2.22	/ DYCS2	CAP3[0]	SCK0	00
15:14	P2[23]	GPIO Port 2.23	/ DYCS3	CAP3[1]	SSEL0	00
17:16	P2[24]	GPIO Port 2.24	CKEOUT0	Reserved	Reserved	00
19:18	P2[25]	GPIO Port 2.25	CKEOUT1	Reserved	Reserved	00
21:20	P2[26]	GPIO Port 2.26	CKEOUT2	MAT3[0]	MISO0	00
23:22	P2[27]	GPIO Port 2.27	CKEOUT3	MAT3[1]	MOSI0	00
25:24	P2[28]	GPIO Port 2.28	DQMOUT0	Reserved	Reserved	00
27:26	P2[29]	GPIO Port 2.29	DQMOUT1	Reserved	Reserved	00
29:28	P2[30]	GPIO Port 2.30	DQMOUT2	MAT3[2]	SDA2	00
31:30	P2[31]	GPIO Port 2.31	DQMOUT3	MAT3[3]	SCL2	00

Pin Function Select Register 6 (PINSEL6 - 0xE002 C018)

PINSEL6	Pin name	Function when 00	Function when 01	Function when 10	Function when 11	Reset value
1:0	P3[0]	GPIO Port 3.0	D0	Reserved	Reserved	00
3:2	P3[1]	GPIO Port 3.1	D1	Reserved	Reserved	00
5:4	P3[2]	GPIO Port 3.2	D2	Reserved	Reserved	00
7:6	P3[3]	GPIO Port 3.3	D3	Reserved	Reserved	00
9:8	P3[4]	GPIO Port 3.4	D4	Reserved	Reserved	00
11:10	P3[5]	GPIO Port 3.5	D5	Reserved	Reserved	00
13:12	P3[6]	GPIO Port 3.6	D6	Reserved	Reserved	00
15:14	P3[7]	GPIO Port 3.7	D7	Reserved	Reserved	00
17:16	P3[8]	GPIO Port 3.8	D8	Reserved	Reserved	00
19:18	P3[9]	GPIO Port 3.9	D9	Reserved	Reserved	00
21:20	P3[10]	GPIO Port 3.10	D10	Reserved	Reserved	00
23:22	P3[11]	GPIO Port 3.11	D11	Reserved	Reserved	00
25:24	P3[12]	GPIO Port 3.12	D12	Reserved	Reserved	00
27:26	P3[13]	GPIO Port 3.13	D13	Reserved	Reserved	00
29:28	P3[14]	GPIO Port 3.14	D14	Reserved	Reserved	00
31:30	P3[15]	GPIO Port 3.15	D15	Reserved	Reserved	00

Pin Function Select Register 7 (PINSEL7 - 0xE002 C01C)

PINSEL7	Pin name	Function when 00	Function when 01	Function when 10	Function when 11	Reset value
1:0	P3[16]	GPIO Port 3.16	D16	PWM0[1]	TXD1	00
3:2	P3[17]	GPIO Port 3.17	D17	PWM0[2]	RXD1	00
5:4	P3[18]	GPIO Port 3.18	D18	PWM0[3]	CTS1	00
7:6	P3[19]	GPIO Port 3.19	D19	PWM0[4]	DCD1	00
9:8	P3[20]	GPIO Port 3.20	D20	PWM0[5]	DSR1	00
11:10	P3[21]	GPIO Port 3.21	D21	PWM0[6]	DR1	00
13:12	P3[22]	GPIO Port 3.22	D22	PCAP0[0]	RI1	00
15:14	P3[23]	GPIO Port 3.23	D23	CAP0[0]	PCAP1[0]	00
17:16	P3[24]	GPIO Port 3.24	D24	CAP0[1]	PWM1[1]	00
19:18	P3[25]	GPIO Port 3.25	D25	MAT0[0]	PWM1[2]	00
21:20	P3[26]	GPIO Port 3.26	D26	MAT0[1]	PWM1[3]	00
23:22	P3[27]	GPIO Port 3.27	D27	CAP1[0]	PWM1[4]	00
25:24	P3[28]	GPIO Port 3.28	D28	CAP1[1]	PWM1[5]	00
27:26	P3[29]	GPIO Port 3.29	D29	MAT1[0]	PWM1[6]	00
29:28	P3[30]	GPIO Port 3.30	D30	MAT1[1]	RTS1	00
31:30	P3[31]	GPIO Port 3.31	D31	MAT1[2]	Reserved	00

Pin Function Select Register 8 (PINSEL8 - 0xE002 C020)

PINSEL8	Pin name	Function when 00	Function when 01	Function when 10	Function when 11	Reset value
1:0	P4[0]	GPIO Port 4.0	A0	Reserved	Reserved	00
3:2	P4[1]	GPIO Port 4.1	A1	Reserved	Reserved	00
5:4	P4[2]	GPIO Port 4.2	A2	Reserved	Reserved	00
7:6	P4[3]	GPIO Port 4.3	A3	Reserved	Reserved	00
9:8	P4[4]	GPIO Port 4.4	A4	Reserved	Reserved	00
11:10	P4[5]	GPIO Port 4.5	A5	Reserved	Reserved	00
13:12	P4[6]	GPIO Port 4.6	A6	Reserved	Reserved	00
15:14	P4[7]	GPIO Port 4.7	A7	Reserved	Reserved	00
17:16	P4[8]	GPIO Port 4.8	A8	Reserved	Reserved	00
19:18	P4[9]	GPIO Port 4.9	A9	Reserved	Reserved	00
21:20	P4[10]	GPIO Port 4.10	A10	Reserved	Reserved	00
23:22	P4[11]	GPIO Port 4.11	A11	Reserved	Reserved	00
25:24	P4[12]	GPIO Port 4.12	A12	Reserved	Reserved	00
27:26	P4[13]	GPIO Port 4.13	A13	Reserved	Reserved	00
29:28	P4[14]	GPIO Port 4.14	A14	Reserved	Reserved	00
31:30	P4[15]	GPIO Port 4.15	A15	Reserved	Reserved	00

Pin Function Select Register 9 (PINSEL9 - 0xE002 C024)

PINSEL9	Pin name	Function when 00	Function when 01	Function when 10	Function when 11	Reset value
1:0	P4[16]	GPIO Port 4.16	A16	Reserved	Reserved	00
3:2	P4[17]	GPIO Port 4.17	A17	Reserved	Reserved	00
5:4	P4[18]	GPIO Port 4.18	A18	Reserved	Reserved	00
7:6	P4[19]	GPIO Port 4.19	A19	Reserved	Reserved	00
9:8	P4[20]	GPIO Port 4.20	A20	SDA2	SCK1	00
11:10	P4[21]	GPIO Port 4.21	A21	SCL2	SSEL1	00
13:12	P4[22]	GPIO Port 4.22	A22	TXD2	MISO1	00
15:14	P4[23]	GPIO Port 4.23	A23	RXD2	MOSI1	00
17:16	P4[24]	GPIO Port 4.24	/ OE	Reserved	Reserved	00
19:18	P4[25]	GPIO Port 4.25	/ WE	Reserved	Reserved	00
21:20	P4[26]	GPIO Port 4.26	BLS0	Reserved	Reserved	00
23:22	P4[27]	GPIO Port 4.27	BLS1	Reserved	Reserved	00
25:24	P4[28]	GPIO Port 4.28	BLS2	MAT2[0]	TXD3	00
27:26	P4[29]	GPIO Port 4.29	BLS3	MAT2[1]	RXD3	00
29:28	P4[30]	GPIO Port 4.30	/ CS0	Reserved	Reserved	00
31:30	P4[31]	GPIO Port 4.31	/ CS1	Reserved	Reserved	00

2. External static memory interface

2.1.1 General connectivity

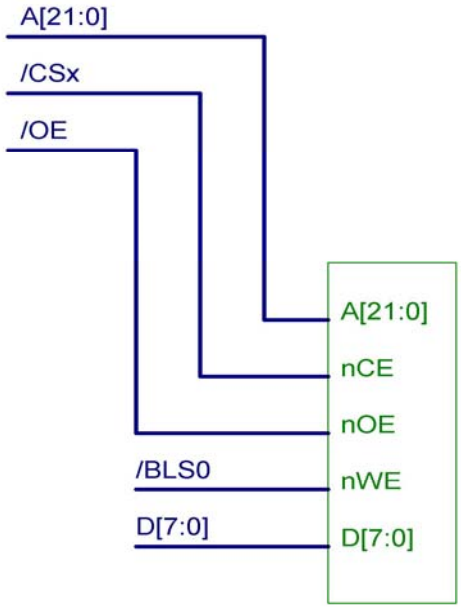

Figure 2-1 Memory bank (8-bit) constructed from 8-bit memory

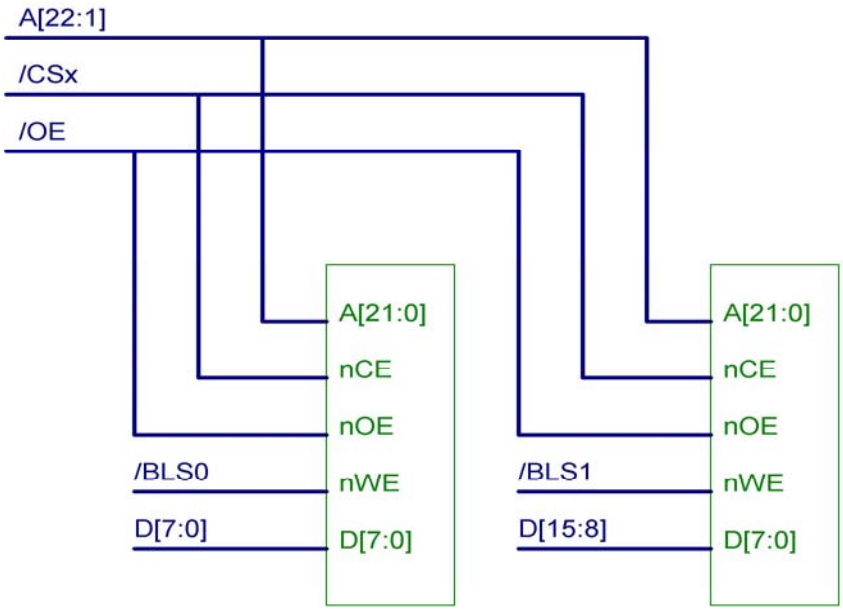

Figure 2-2 Memory bank (16-bit) constructed from 8-bit memory

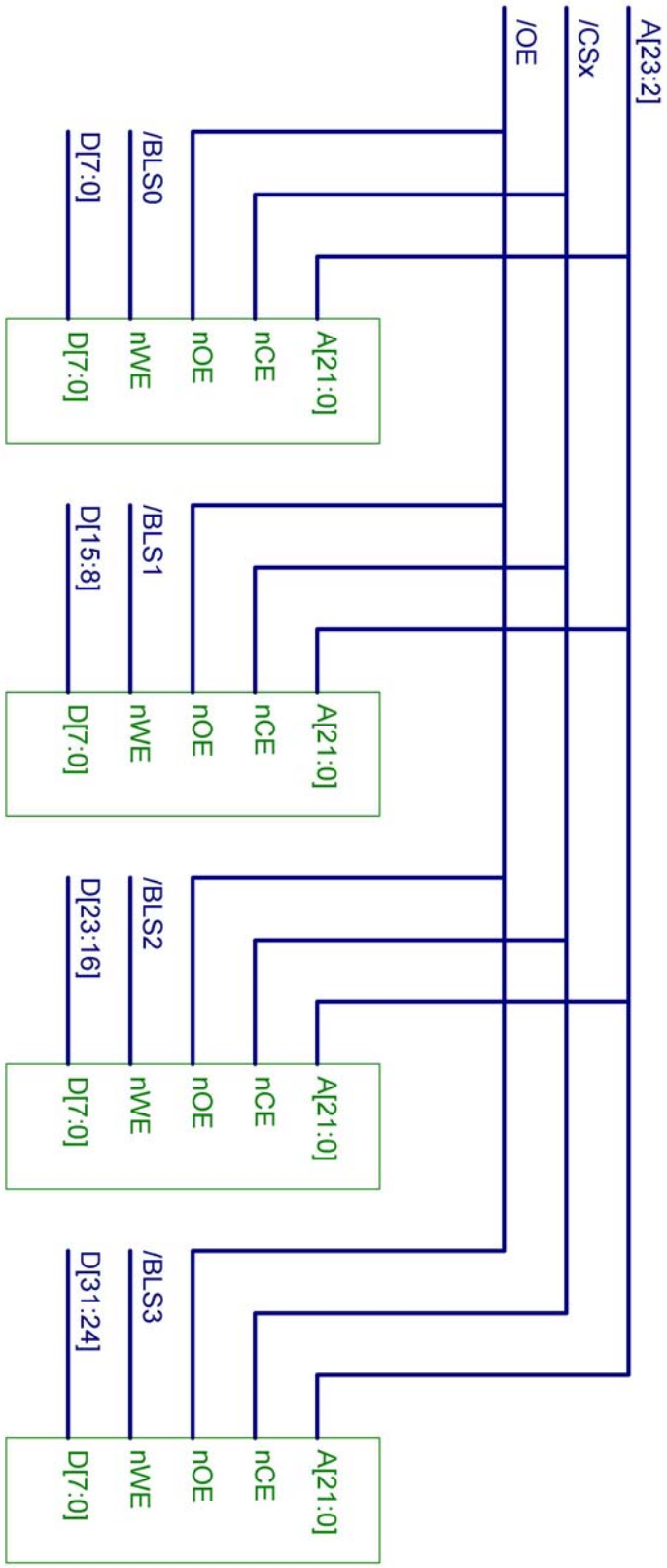

Figure 2-3 Memory bank (32-bit) constructed from 8-bit memory

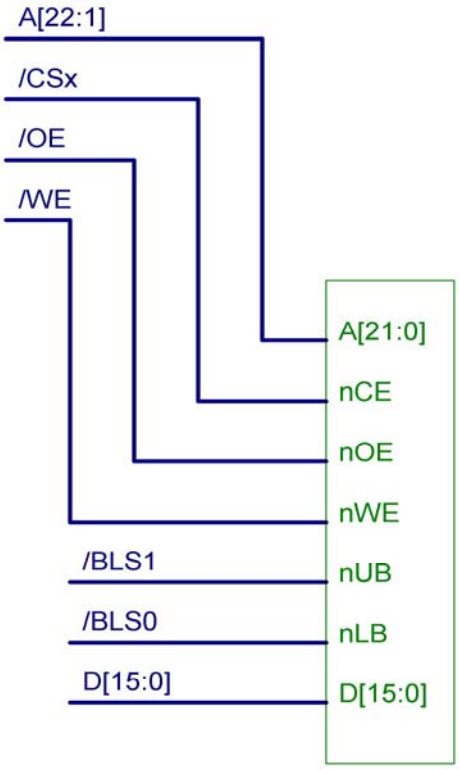

Figure 2-4 Memory bank (16-bit) constructed from 16-bit memory

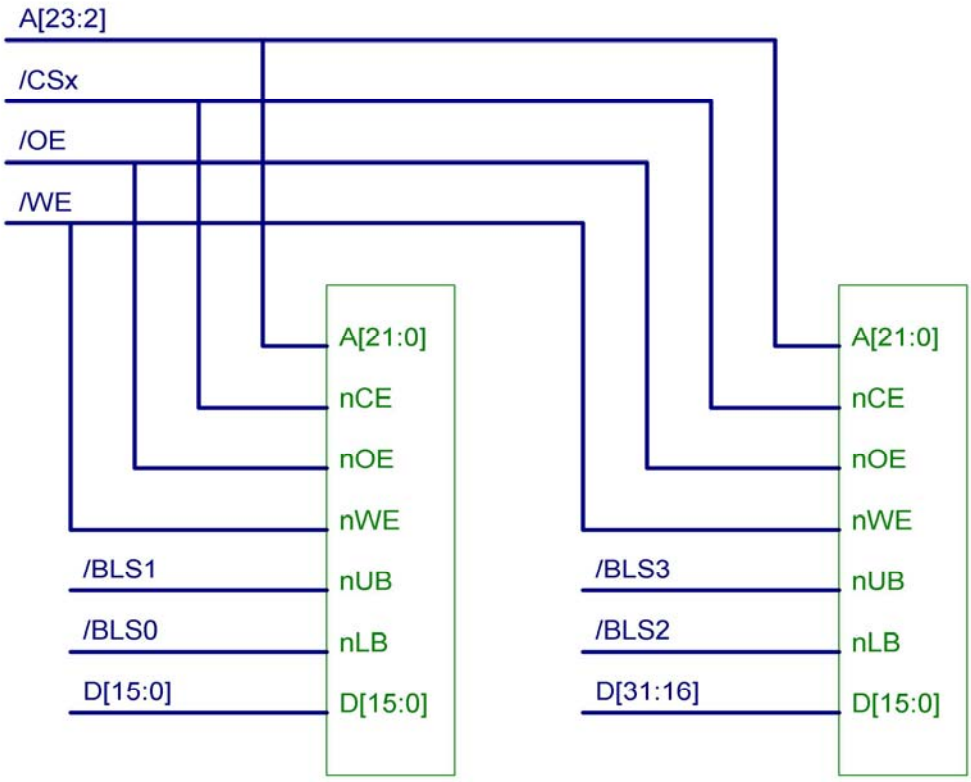

Figure 2-5 Memory bank (32-bit) constructed from 16-bit memory

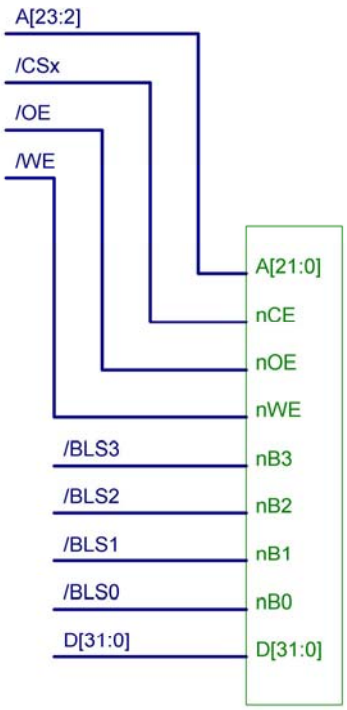

Figure 2-6 Memory bank (32-bit) constructed from 32-bit memory

2.1.2 Typical connectivity

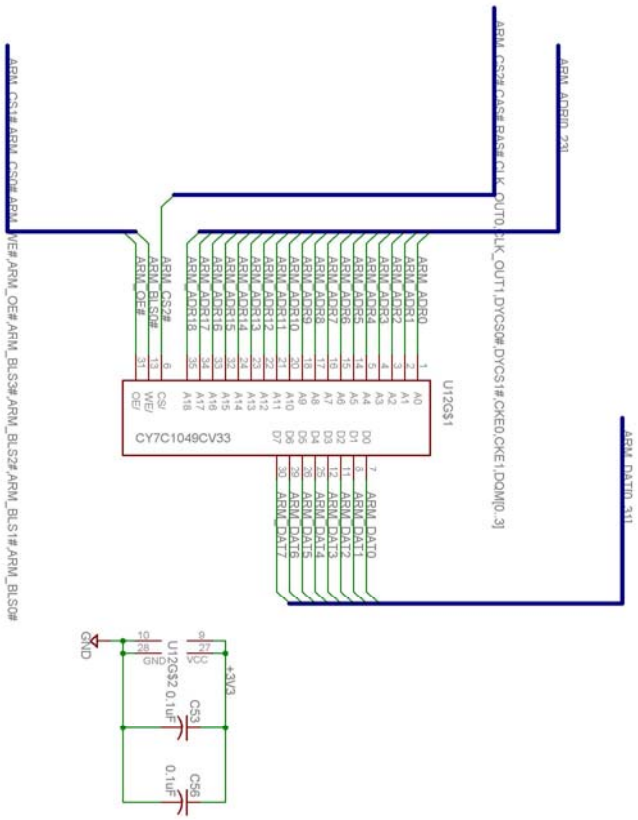

Figure 2-7 Memory bank (8-bit) constructed from 8-bit memory

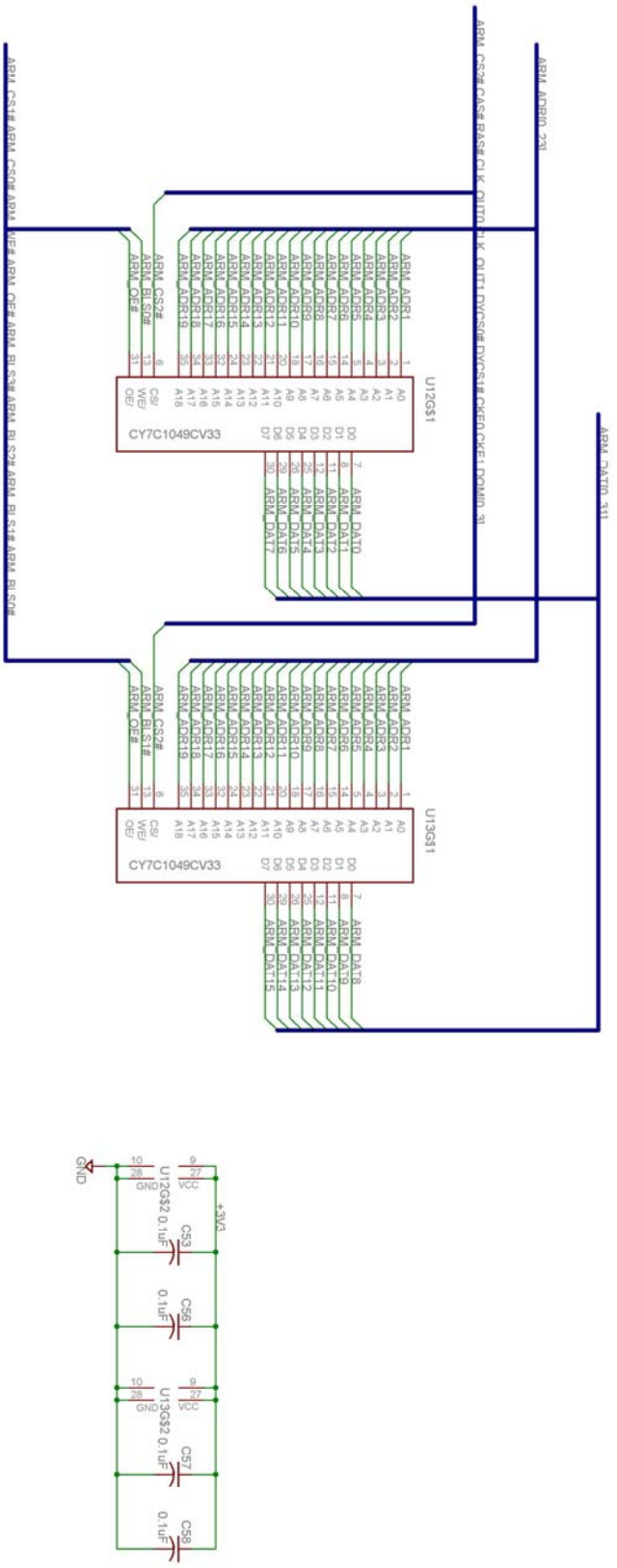

Figure 2-8 Memory bank (16-bit) constructed from 8-bit memory

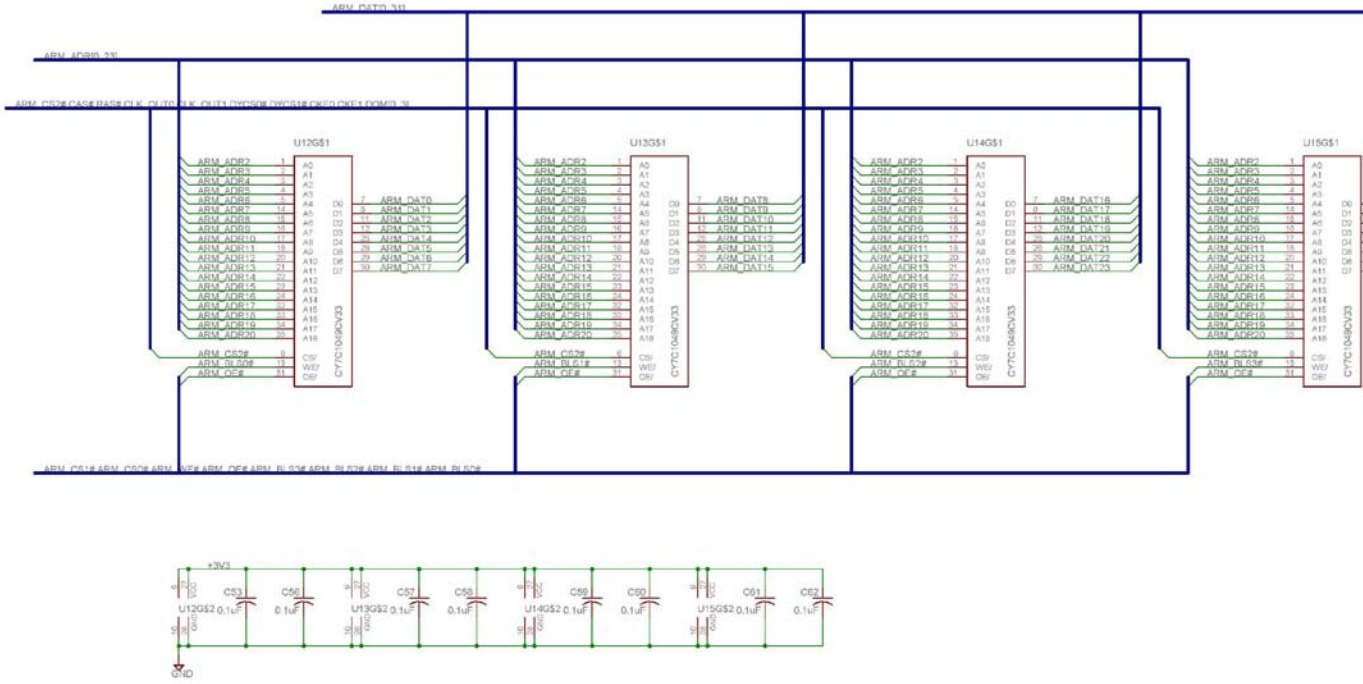

Figure 2-9 Memory bank (32-bit) constructed from 8-bit memory

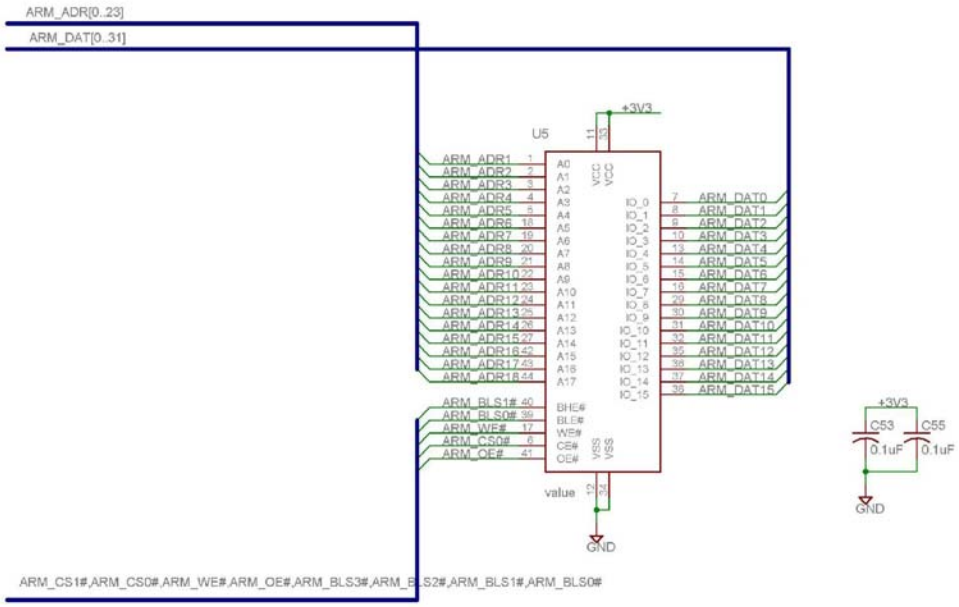

Figure 2-10 Memory bank (16-bit) constructed from 16-bit memory

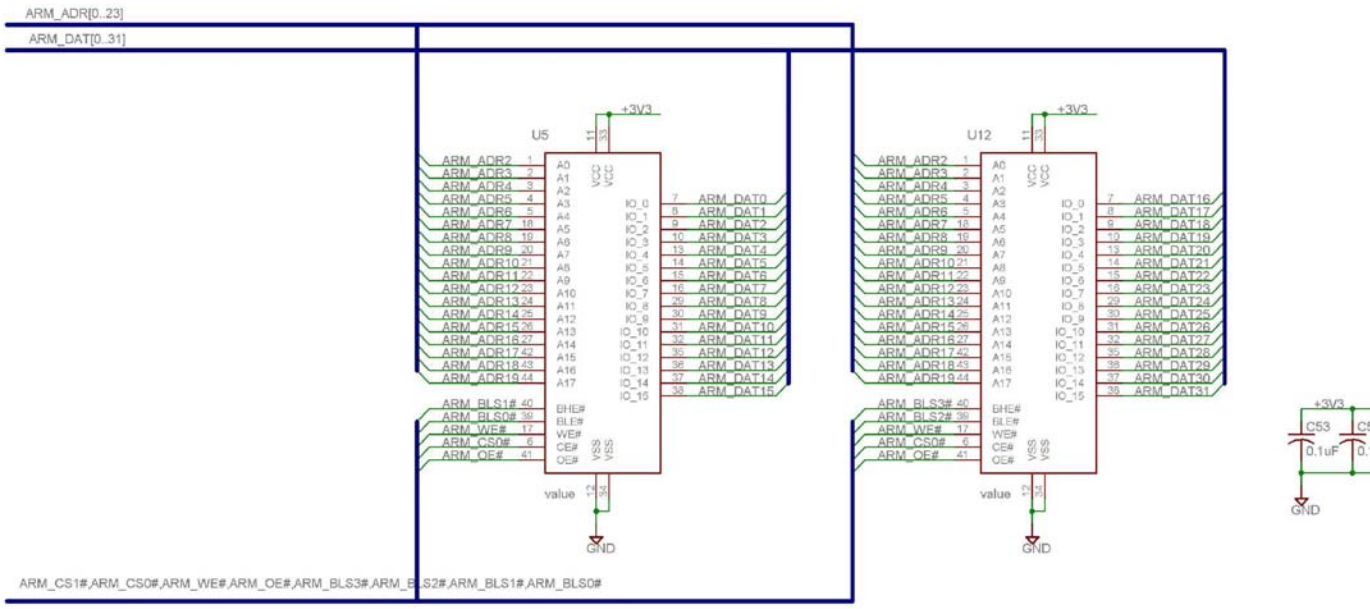

Figure 2-11 Memory bank (32-bit) constructed from 16-bit memory

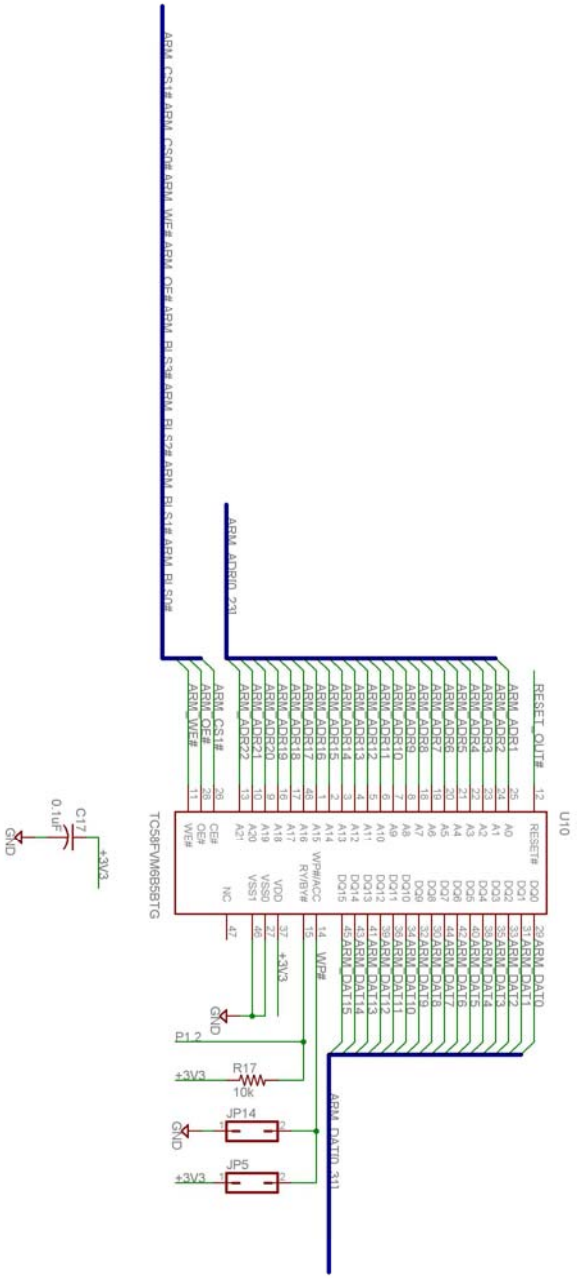

Figure 2-12 Memory bank (16-bit) constructed from 16-bit memory

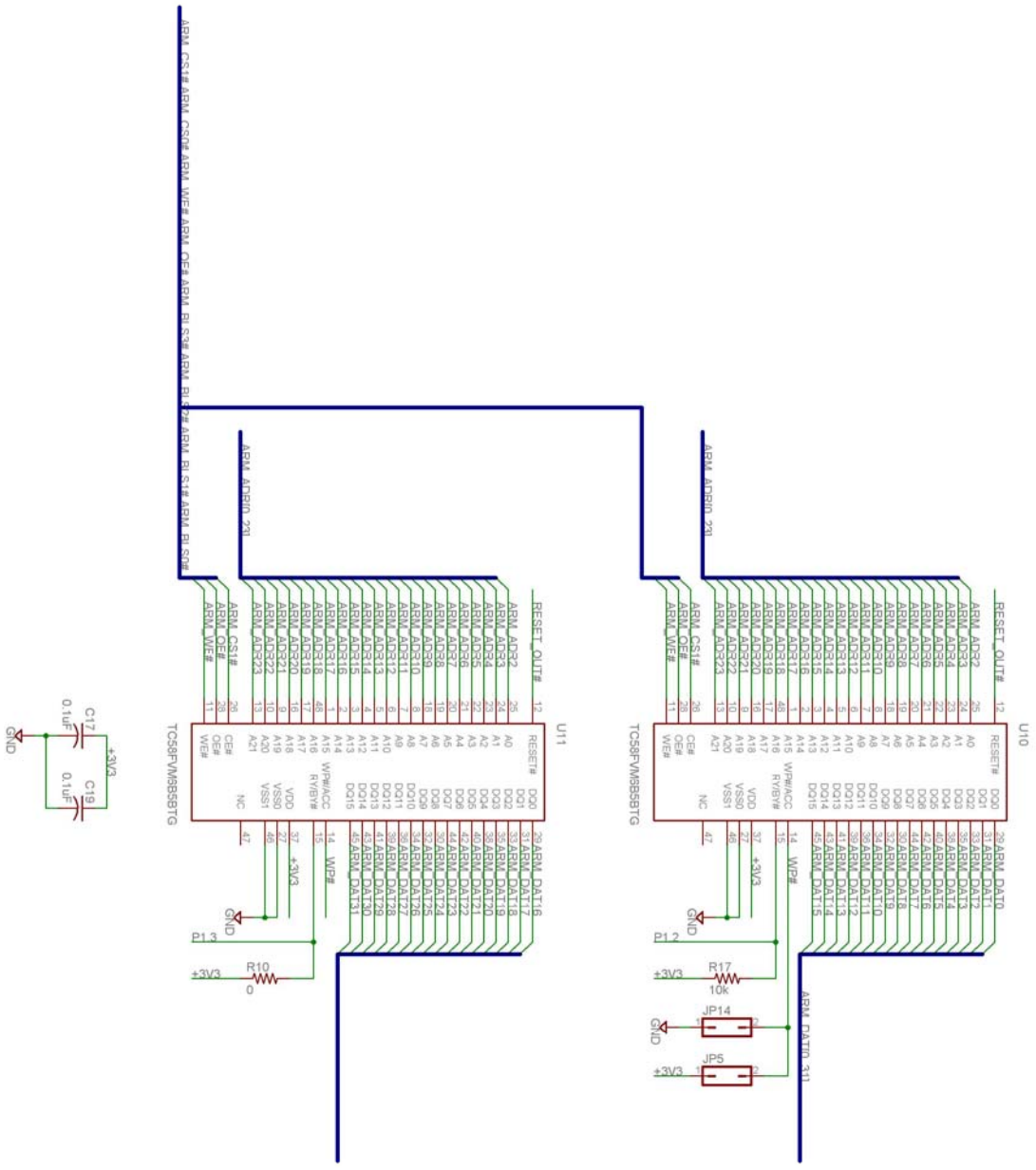

Figure 2-13 Memory bank (32-bit) constructed from 16-bit memory

2.2 Boot from external flash, external SRAM remapped after boot

The system set up is:
- chip select 1 is connected to the external boot flash device
- chip select 0 is connected to the external SRAM to be remapped to 0x00000000 after boot.

The boot sequence is as follows:
- At power-on reset chip select 1 (/CS1) is mirrored into chip select 0 (/CS0) and chip select 4 (/DYCS0).
- When the power-on reset goes inactive, the processor starts booting from 0x00000000 in memory.
- The software programs the optimum delay values in the flash memory so that the boot code can run at full speed.
- The code branches to chip select 1 so that the code can continue executing from the nonremapped memory location.
- The appropriate values are programmed into the EMC to configure chip select 0.
- The address mirroring is disabled by clearing the Address Mirror (M) field in the EMCControl Register.
- The ARM reset and interrupt vectors are copied from flash memory to SRAM that can then be accessed at address 0x00000000.
- More boot, initialization, or application code is executed.

2.3 EMC Initialization

```
/*****************************************************************
 * Figure 2-10 Memory bank (16-bit) constructed from 16-bit memory *
 * Figure 2-11 Memory bank (32-bit) constructed from 16-bit memory *
 * Cypress CY7C1041CV33-12ZXC                                     *
 * LPC2468 CCLK : 60MHz                                           *
 * LPC2460 CCLK : 72MHz                                           *
 *****************************************************************/
#ifdef SRAM_CS0_CONFIG_32BIT
    EMCSTATICCNFG0 = 0x00000082;
#endif
#ifdef SRAM_CS0_CONFIG_16BIT
    EMCSTATICCNFG0 = 0x00000081;
#endif
    EMCSTATICWAITWEN0 = 0x00000000;  /* (n + 1) -> 1 clock cycles */
    EMCSTATICWAITOEN0 = 0x00000000;  /* (n) -> 0 clock cycles     */
    EMCSTATICWAITRD0 = 0x00000001;   /* (n + 1) -> 2 clock cycles */
    EMCSTATICWAITPG0 = 0x00000000;   /* (n + 1) -> 1 clock cycles */
    EMCSTATICWAITWR0 = 0x00000000;   /* (n + 2) -> 2 clock cycles */
    EMCSTATICWAITTURN0 = 0x00000000; /* (n + 1) -> 1 clock cycles */
```

```
/*****************************************************************
 * Figure 2-12 Memory bank (16-bit) constructed from 16-bit memory *
 * Figure 2-13 Memory bank (32-bit) constructed from 16-bit memory *
 * Toshiba TC58FVM6B5BTG65                                        *
 * LPC2468 CCLK : 60MHz                                           *
 * LPC2460 CCLK : 72MHz                                           *
 *****************************************************************/
#ifdef FLASH_CS1_CONFIG_32BIT
    EMCSTATICCNFG1 = 0x00000082;
#endif
#ifdef FLASH_CS1_CONFIG_16BIT
    EMCSTATICCNFG1 = 0x00000081;
#endif
    EMCSTATICWAITWEN1 = 0x00000000;  /* (n + 1) -> 1 clock cycles */
    EMCSTATICWAITOEN1 = 0x00000000;  /* (n) -> 0 clock cycles     */
    EMCSTATICWAITRD1 = 0x00000004;   /* (n + 1) -> 5 clock cycles */
    EMCSTATICWAITPG1 = 0x00000000;   /* (n + 1) -> 1 clock cycles */
    EMCSTATICWAITWR1 = 0x00000003;   /* (n + 2) -> 5 clock cycles */
    EMCSTATICWAITTURN1 = 0x00000000; /* (n + 1) -> 1 clock cycles */

/*****************************************************************
 * Figure 2-7 Memory bank (8-bit) constructed from 8-bit memory    *
 * Figure 2-8 Memory bank (16-bit) constructed from 8-bit memory   *
 * Figure 2-9 Memory bank (32-bit) constructed from 8-bit memory   *
 * Cypress CY7C1049CV33-15VXC                                     *
 * LPC2468 CCLK : 60MHz                                           *
 * LPC2460 CCLK : 72MHz                                           *
 *****************************************************************/
#ifdef SRAM_CS2_CONFIG_32BIT
    EMCSTATICCNFG2 = 0x00000002;
#endif
#ifdef SRAM_CS2_CONFIG_16BIT
    EMCSTATICCNFG2 = 0x00000001;
#endif
#ifdef SRAM_CS2_CONFIG_8BIT
    EMCSTATICCNFG2 = 0x00000000;
#endif
    EMCSTATICWAITWEN2 = 0x00000000;  /* (n + 1) -> 1 clock cycles */
    EMCSTATICWAITOEN2 = 0x00000000;  /* (n) -> 0 clock cycles     */
    EMCSTATICWAITRD2 = 0x00000001;   /* (n + 1) -> 2 clock cycles */
    EMCSTATICWAITPG2 = 0x00000000;   /* (n + 1) -> 1 clock cycles */
    EMCSTATICWAITWR2 = 0x00000000;   /* (n + 2) -> 2 clock cycles */
    EMCSTATICWAITTURN2 = 0x00000000; /* (n + 1) -> 1 clock cycles */
```

2.4 EMC timing diagram

2.4.1 Zero wait state burst read timing diagram

Timing parameter	Value
WAITRD	0
WAITOEN	0
WAITPAGE	NA
WAITWR	NA
WAITWEN	NA
WAITTURN	NA

Cycle	Description
T0	AHB address provided to memory controller.
T0-T1	AHB transaction processing.
T1-T2	Arbitration of AHB memory ports.
T2-T3	Raise EBI bus request.
T3-T4	Register EBI grant.
T4-T5	Static memory read 0 address, chip select, and control signals submitted to static memory.
T5-T6	Static memory read 1 address, chip select, and control signals submitted to static memory. Read data 0 returned from the static memory. Read data 0 is provided to the AHB.
T6-T7	Static memory read 2 address, chip select, and control signals submitted to static memory. Read data 1 returned from the static memory. Read data 1 is provided to the AHB.
T7-T8	Static memory read 3 address, chip select, and control signals submitted to static memory. Read data 2 returned from the static memory. Read data 2 is provided to the AHB.
T8-T9	Read data 3 returned from the static memory. Read data 3 is provided to the AHB.

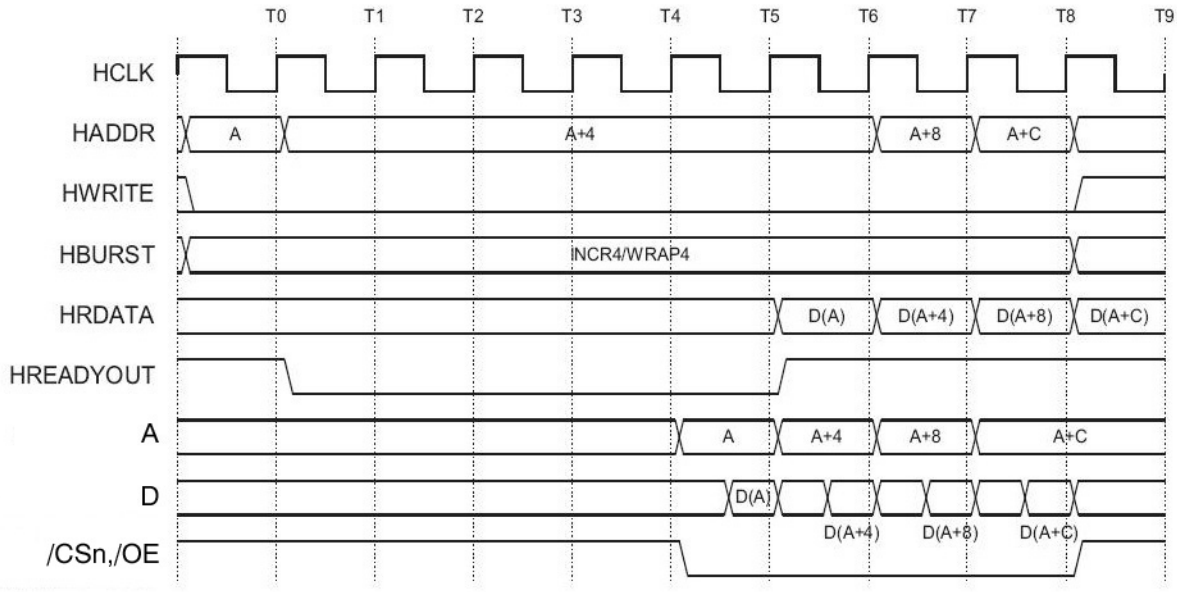

Figure 2-14 Zero wait state burst read timing diagram

2.4.2 Two wait state burst read timing diagram

Timing parameter	Value
WAITRD	2
WAITOEN	0
WAITPAGE	NA
WAITWR	NA
WAITWEN	NA
WAITTURN	NA

Cycle	Description
T0	AHB address provided to memory controller.
T0-T1	AHB transaction processing.
T1-T2	Arbitration of AHB memory ports.
T2-T3	Raise EBI bus request.
T3-T4	Register EBI grant.
T4-T5	Static memory address, chip select, and control signals submitted to static memory.
T5-T6	Read wait state 1.
T6-T7	Read wait state 2.
T7-T8	Read data 0 returned from the static memory. Read data 0 is provided to the AHB. Static memory transfer 1, address, chip select, and control signals submitted to static memory.
T8-T9	Read wait state 1.
T9-T10	Read wait state 2.
T10-T11	Read data 1 returned from the static memory. Read data1 is provided to the AHB. Static memory transfer 2, address, chip select, and control signals submitted to static memory.
T11-T12	Read wait state 1.

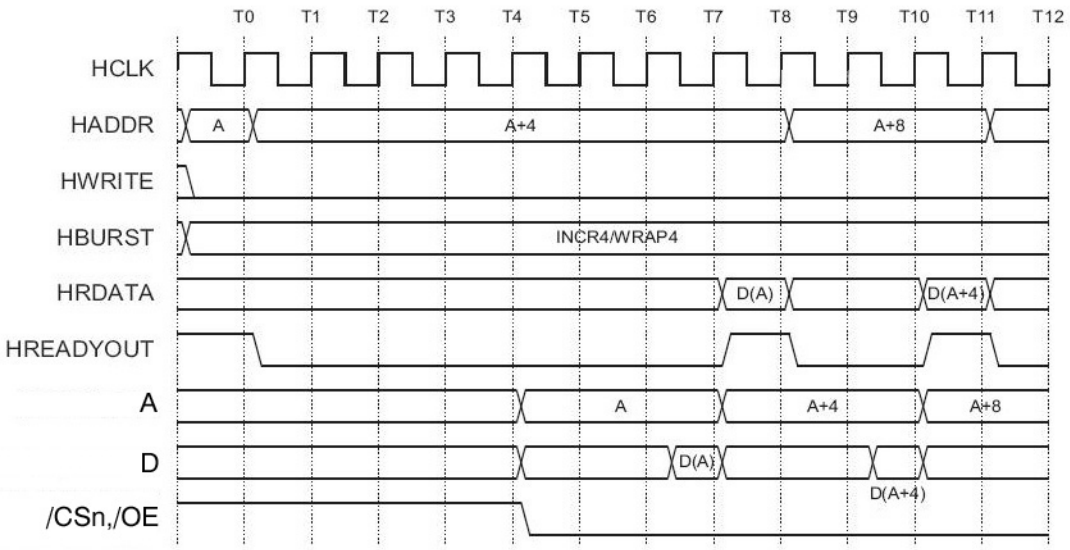

Figure 2-15 Two wait state burst read timing diagram

2.4.3 Two wait state page mode read timing diagram

Timing parameter	Value
WAITRD	2
WAITOEN	0
WAITPAGE	1
WAITWR	NA
WAITWEN	NA
WAITTURN	NA

Cycle	Description
T0	AHB address provided to memory controller.
T0-T1	AHB transaction processing.
T1-T2	Arbitration of AHB memory ports.
T2-T3	Raise EBI bus request.
T3-T4	Register EBI grant.
T4-T5	Static memory transfer 0, address, chip select, and control signals submitted to static memory.
T5-T6	Read wait state 1.
T6-T7	Read wait state 2.
T7-T8	Read data 0 returned from the static memory. Read data 0 is provided to the AHB. Static memory transfer 1, address, chip select, and control signals submitted to static memory.
T8-T9	Read page mode wait state1.
T9-T10	Read data 1 returned from the static memory. Read data1 is provided to the AHB. Static memory transfer 2, address, chip select, and control signals submitted to static memory.
T10-T11	Read page mode wait state1.
T11-T12	Read data 2 returned from the static memory. Read data 2 is provided to the AHB. Static memory transfer 3, address, chip select, and control signals submitted to static memory.

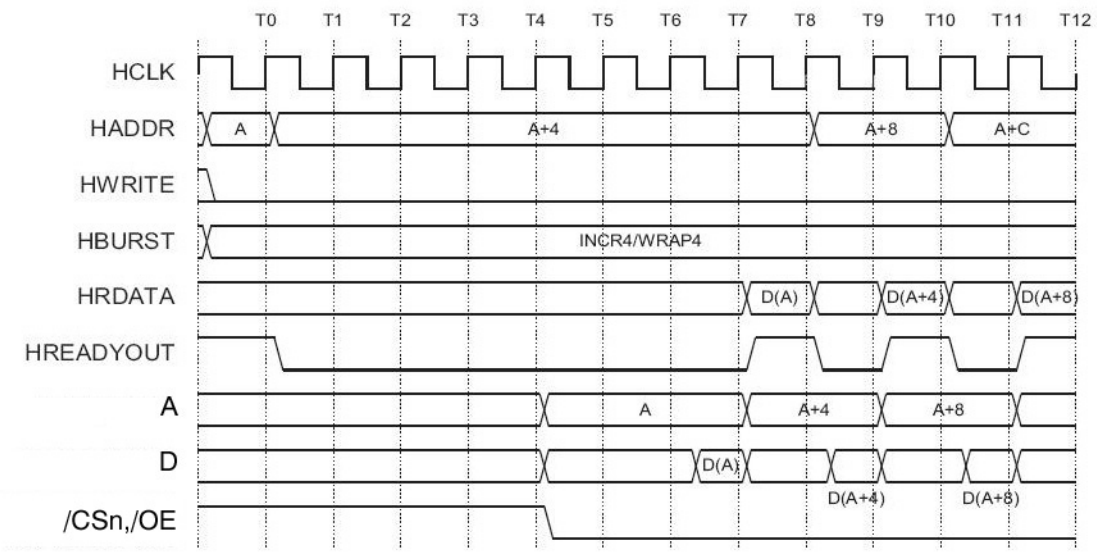

Figure 2-16 Two wait state page mode read timing diagram

2.4.4 Zero wait state write timing diagram

Timing parameter	Value
WAITRD	NA
WAITOEN	NA
WAITPAGE	NA
WAITWR	0
WAITWEN	0
WAITTURN	NA

Cycle	Description
T0	AHB address provided to memory controller.
T0-T1	AHB transaction processing.
T1-T2	Arbitration of AHB memory ports.
T2-T3	Raise EBI bus request.
T3-T4	Register EBI grant.
T4-T5	Static memory transfer 0, address, chip select, and control signals submitted to static memory. Write data is read from the AHB memory port. Write enable inactive.
T5-T6	Write enable taken active. Write data submitted to static memory. Static memory writes the data.
T6-T7	Static memory writes the data. Write enable taken inactive.
T7-T8	Static memory control signals taken inactive.

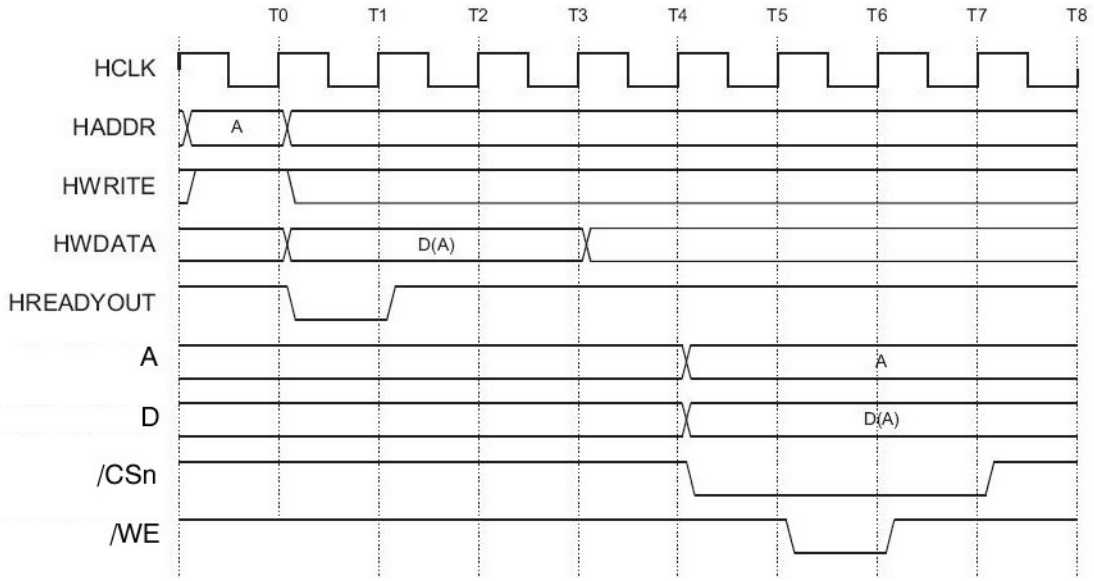

Figure 2-17 Zero wait state write timing diagram

2.4.5 Two wait state write timing diagram

Timing parameter	Value
WAITRD	NA
WAITOEN	NA
WAITPAGE	NA
WAITWR	2
WAITWEN	0
WAITTURN	NA

Cycle	Description
T0	AHB address provided to memory controller.
T0-T1	AHB transaction processing.
T1-T2	Arbitration of AHB memory ports.
T2-T3	Raise EBI bus request.
T3-T4	Register EBI grant.
T4-T5	Static memory transfer 0, address, chip select, and control signals submitted to static memory. Write data is read from the AHB memory port. Write enable inactive.
T5-T6	Write enable taken active. Write data submitted to static memory.
T6-T7	Wait state 1.
T7-T8	Wait state 2.
T8-T9	Static memory writes the data. Write enable taken inactive.
T9-T10	Static memory control signals taken inactive.

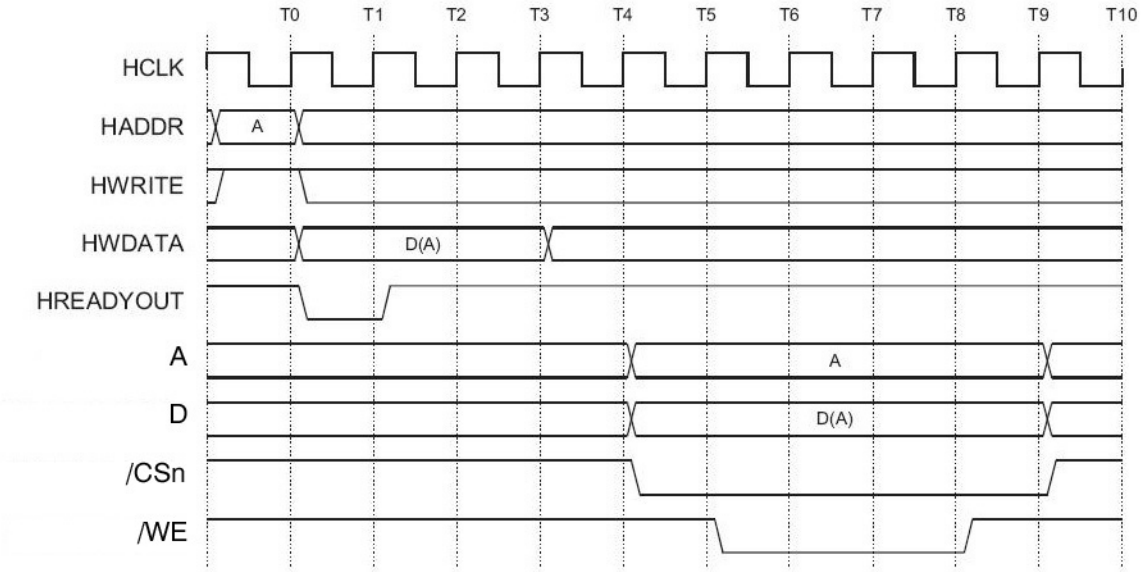

Figure 2-18 Two wait state write timing diagram

2.5 Bus turnaround

The EMC can be configured for each memory bank to use external bus turnaround cycles between read and write memory accesses. The WAITTURN field can be programmed for 1 to 16 bus turnaround wait states. This is to avoid bus contention on the external memory data bus. Bus turnaround cycles are generated between external bus transfers as follows:
- read to read (different memory banks)
- read to write (same memory bank)
- read to write (different memory banks).

2.6 Power consumption

2.6.1 LPC2468 with one 16-bit SRAM

Symbol	Parameter	Conditions	Typ	Unit
$I_{DD(3V3)}$	supply current	$V_{DD(3V3)}$ =3.3V; T_{amb} = 25°C; [1]	16.8	mA
$I_{DD(DCDC)(3V3)}$	DC-DC coverter supply current	$V_{DD(DCDC)(3V3)}$ =3.3V; T_{amb} = 25°C; [1]	73.3	mA
$I_{DD(SRAM)}$	supply current	$V_{DD(SRAM)}$ =3.3V; T_{amb} = 25°C; [1]	14.0	mA

[1] LPC2468 CCLK = 60MHz; PCLK = ¼ CCLK;
Peripherals enabled are default after reset;
Code executed from internal flash writes various patterns to the SRAM and verifies;
Figure 2-10 memory bank (16-bit) constructed from 16-bit memory;

2.6.2 LPC2468 with two 16-bit SRAM

Symbol	Parameter	Conditions	Typ	Unit
$I_{DD(3V3)}$	supply current	$V_{DD(3V3)}$ =3.3V; T_{amb} = 25°C; [2]	16.8	mA
$I_{DD(DCDC)(3V3)}$	DC-DC coverter supply current	$V_{DD(DCDC)(3V3)}$ =3.3V; T_{amb} = 25°C; [2]	73.3	mA
$I_{DD(SRAM)}$	supply current	$V_{DD(SRAM)}$ =3.3V; T_{amb} = 25°C; [2]	27.2	mA

[2] LPC2468 CCLK = 60MHz; PCLK = ¼ CCLK;
Peripherals enabled are default after reset;
Code executed from internal flash writes various patterns to the SRAM and verifies;
Figure 2-11 memory bank (32-bit) constructed from 16-bit memory;

2.6.3 LPC2468 with one 8-bit SRAM

Symbol	Parameter	Conditions	Typ	Unit
$I_{DD(3V3)}$	supply current	$V_{DD(3V3)}$ =3.3V; T_{amb} = 25°C; [3]	14.4	mA
$I_{DD(DCDC)(3V3)}$	DC-DC coverter supply current	$V_{DD(DCDC)(3V3)}$ =3.3V; T_{amb} = 25°C; [3]	73.3	mA
$I_{DD(SRAM)}$	supply current	$V_{DD(SRAM)}$ =3.3V; T_{amb} = 25°C; [3]	24.8	mA

[3] LPC2468 CCLK = 60MHz; PCLK = ¼ CCLK;
Peripherals enabled are default after reset;
Code executed from internal flash writes various patterns to the SRAM and verifies.
Figure 2-7 memory bank (8-bit) constructed from 8-bit memory;

2.6.4 LPC2468 with two 8-bit SRAM

Symbol	Parameter	Conditions	Typ	Unit
$I_{DD(3V3)}$	supply current	$V_{DD(3V3)}$ =3.3V; T_{amb} = 25°C; [4]	16.8	mA
$I_{DD(DCDC)(3V3)}$	DC-DC coverter supply current	$V_{DD(DCDC)(3V3)}$ =3.3V; T_{amb} = 25°C; [4]	73.3	mA
$I_{DD(SRAM)}$	supply current	$V_{DD(SRAM)}$ =3.3V; T_{amb} = 25°C; [4]	30.0	mA

[4] LPC2468 CCLK = 60MHz; PCLK = ¼ CCLK;
Peripherals enabled are default after reset;
Code executed from internal flash writes various patterns to the SRAM and verifies.
Figure 2-8 memory bank (16-bit) constructed from 8-bit memory;

2.6.5 LPC2468 with four 8-bit SRAM

Symbol	Parameter	Conditions	Typ	Unit
$I_{DD(3V3)}$	supply current	$V_{DD(3V3)}$ =3.3V; T_{amb} = 25°C; [5]	16.8	mA
$I_{DD(DCDC)(3V3)}$	DC-DC coverter supply current	$V_{DD(DCDC)(3V3)}$ =3.3V; T_{amb} = 25°C; [5]	73.3	mA
$I_{DD(SRAM)}$	supply current	$V_{DD(SRAM)}$ =3.3V; T_{amb} = 25°C; [5]	55.6	mA

[5] LPC2468 CCLK = 60MHz; PCLK = ¼ CCLK;
Peripherals enabled are default after reset;
Code executed from internal flash writes various patterns to the SRAM and verifies.
Figure 2-9 memory bank (32-bit) constructed from 8-bit memory;

2.6.6 LPC2460 with one 16-bit flash and one 16-bit SRAM

Symbol	Parameter	Conditions	Typ	Unit
$I_{DD(3V3)}$	supply current	$V_{DD(3V3)}$ =3.3V; T_{amb} = 25°C; [6]	9.2	mA
$I_{DD(DCDC)(3V3)}$	DC-DC coverter supply current	$V_{DD(DCDC)(3V3)}$ =3.3V; T_{amb} = 25°C; [6]	64.4	mA
$I_{DD(FLASH)}$	supply current	$V_{DD(FLASH)}$ =3.3V; T_{amb} = 25°C; [6]	46.7	mA
$I_{DD(SRAM)}$	supply current	$V_{DD(SRAM)}$ =3.3V; T_{amb} = 25°C; [6]	6.0	mA

[6] LPC2460 CCLK = 72MHz; PCLK = ¼ CCLK;
Peripherals enabled are default after reset;
Code executed from external flash writes various patterns to the SRAM and verifies.
Figure 2-12 memory bank (16-bit) constructed from 16-bit memory;
Figure 2-10 memory bank (16-bit) constructed from 16-bit memory;

2.6.7 LPC2460 with one 16-bit flash and two 16-bit SRAM

Symbol	Parameter	Conditions	Typ	Unit
$I_{DD(3V3)}$	supply current	$V_{DD(3V3)}$ =3.3V; T_{amb} = 25°C; [7]	9.2	mA
$I_{DD(DCDC)(3V3)}$	DC-DC coverter supply current	$V_{DD(DCDC)(3V3)}$ =3.3V; T_{amb} = 25°C; [7]	64.4	mA
$I_{DD(FLASH)}$	supply current	$V_{DD(FLASH)}$ =3.3V; T_{amb} = 25°C; [7]	46.7	mA
$I_{DD(SRAM)}$	supply current	$V_{DD(SRAM)}$ =3.3V; T_{amb} = 25°C; [7]	10.0	mA

[7] LPC2460 CCLK = 72MHz; PCLK = ¼ CCLK;
Peripherals enabled are default after reset;
Code executed from external flash writes various patterns to the SRAM and verifies.
Figure 2-12 memory bank (16-bit) constructed from 16-bit memory;
Figure 2-11 memory bank (32-bit) constructed from 16-bit memory;

2.6.8 LPC2460 with one 16-bit flash and one 8-bit SRAM

Symbol	Parameter	Conditions	Typ	Unit
$I_{DD(3V3)}$	supply current	$V_{DD(3V3)}$ =3.3V; T_{amb} = 25°C; [8]	9.2	mA
$I_{DD(DCDC)(3V3)}$	DC-DC coverter supply current	$V_{DD(DCDC)(3V3)}$ =3.3V; T_{amb} = 25°C; [8]	64.4	mA
$I_{DD(FLASH)}$	supply current	$V_{DD(FLASH)}$ =3.3V; T_{amb} = 25°C; [8]	46.7	mA
$I_{DD(SRAM)}$	supply current	$V_{DD(SRAM)}$ =3.3V; T_{amb} = 25°C; [8]	6.8	mA

[8] LPC2460 CCLK = 72MHz; PCLK = ¼ CCLK;
Peripherals enabled are default after reset;
Code executed from external flash writes various patterns to the SRAM and verifies.
Figure 2-12 memory bank (16-bit) constructed from 16-bit memory;
Figure 2-7 Memory bank (8-bit) constructed from 8-bit memory;

2.6.9 LPC2460 with one 16-bit flash and two 8-bit SRAM

Symbol	Parameter	Conditions	Typ	Unit
$I_{DD(3V3)}$	supply current	$V_{DD(3V3)}$ =3.3V; T_{amb} = 25°C; [9]	9.2	mA
$I_{DD(DCDC)(3V3)}$	DC-DC coverter supply current	$V_{DD(DCDC)(3V3)}$ =3.3V; T_{amb} = 25°C; [9]	64.4	mA
$I_{DD(FLASH)}$	supply current	$V_{DD(FLASH)}$ =3.3V; T_{amb} = 25°C; [9]	46.7	mA
$I_{DD(SRAM)}$	supply current	$V_{DD(SRAM)}$ =3.3V; T_{amb} = 25°C; [9]	11.2	mA

[9] LPC2460 CCLK = 72MHz; PCLK = ¼ CCLK;
Peripherals enabled are default after reset;
Code executed from external flash writes various patterns to the SRAM and verifies.
Figure 2-12 memory bank (16-bit) constructed from 16-bit memory;
Figure 2-8 Memory bank (16-bit) constructed from 8-bit memory;

2.6.10 LPC2460 with one 16-bit flash and four 8-bit SRAM

Symbol	Parameter	Conditions	Typ	Unit
$I_{DD(3V3)}$	supply current	$V_{DD(3V3)}$ =3.3V; T_{amb} = 25°C; [10]	9.2	mA
$I_{DD(DCDC)(3V3)}$	DC-DC coverter supply current	$V_{DD(DCDC)(3V3)}$ =3.3V; T_{amb} = 25°C; [10]	64.4	mA
$I_{DD(FLASH)}$	supply current	$V_{DD(FLASH)}$ =3.3V; T_{amb} = 25°C; [10]	46.7	mA
$I_{DD(SRAM)}$	supply current	$V_{DD(SRAM)}$ =3.3V; T_{amb} = 25°C; [10]	20.8	mA

[10] LPC2460 CCLK = 72MHz; PCLK = ¼ CCLK;
Peripherals enabled are default after reset;
Code executed from external flash writes various patterns to the SRAM and verifies.
Figure 2-12 memory bank (16-bit) constructed from 16-bit memory;
Figure 2-9 Memory bank (32-bit) constructed from 8-bit memory;

3. External dynamic memory interface

3.1 Typical connectivity

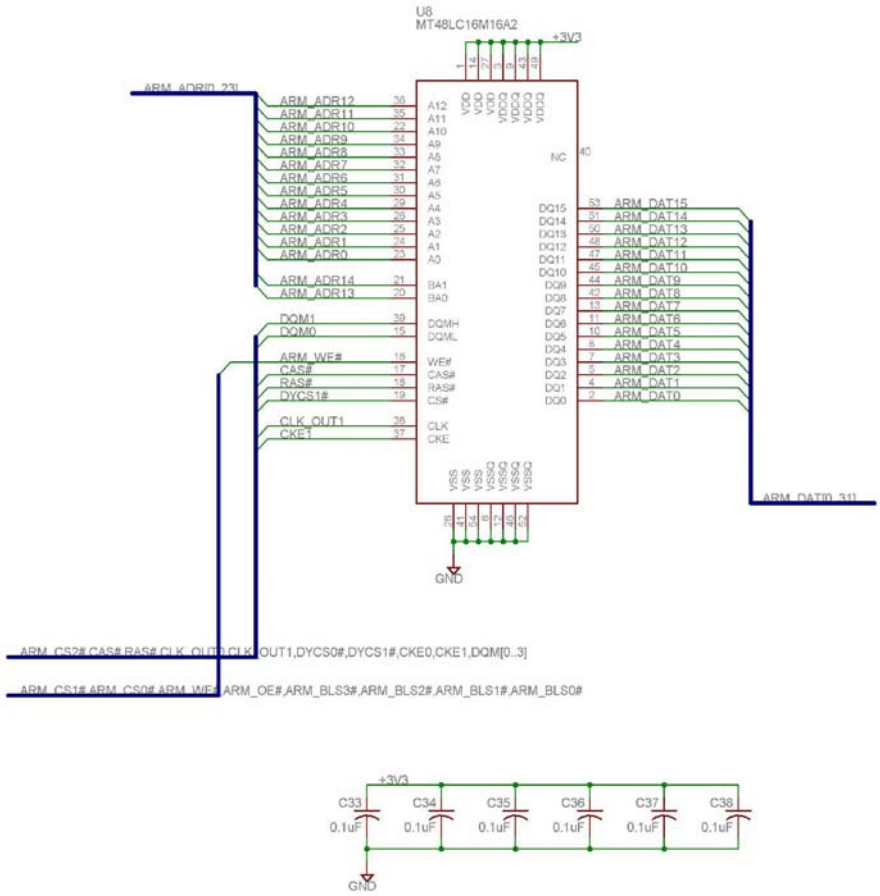

Figure 3-1 Memory bank (16-bit) constructed from 16-bit memory

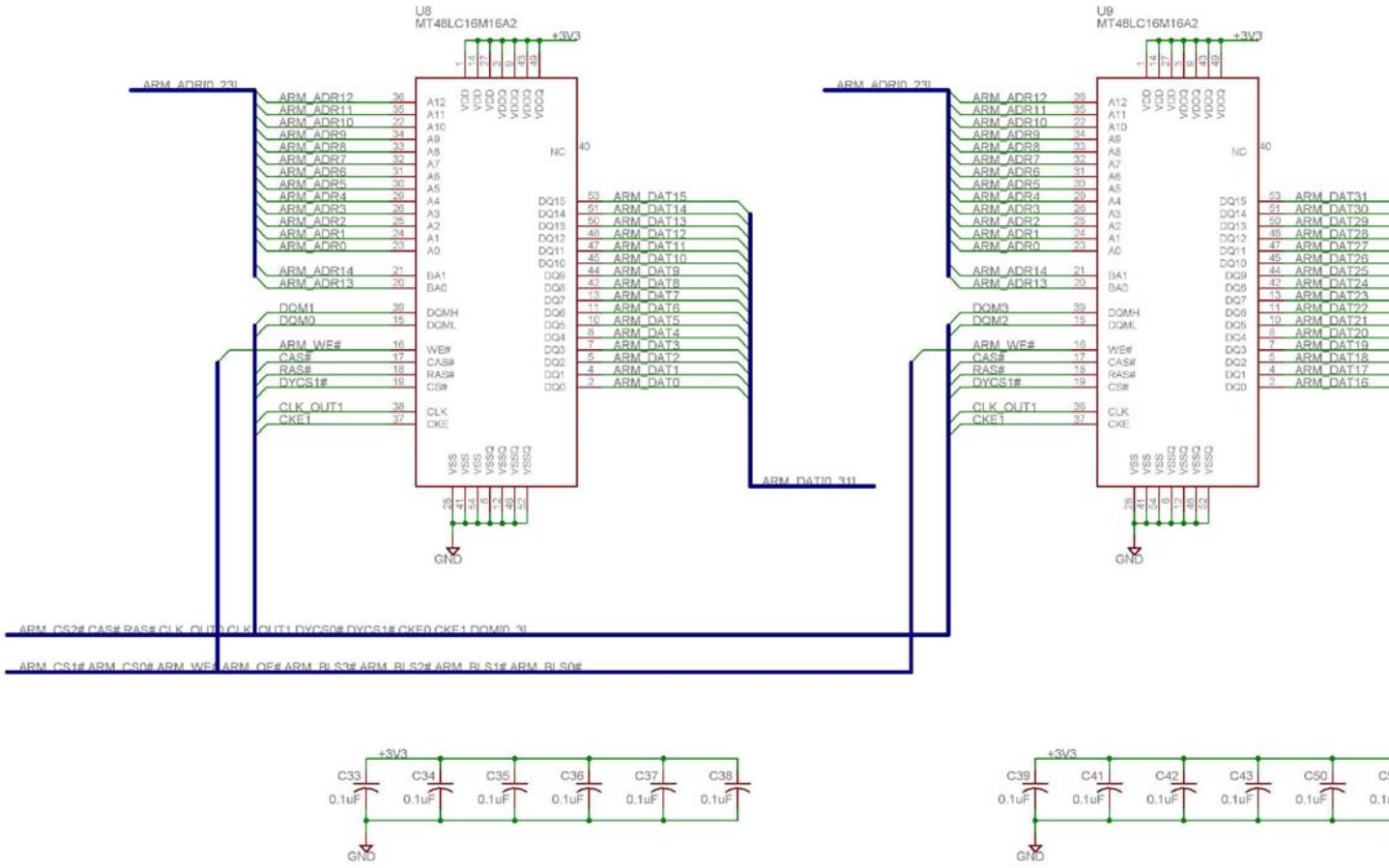

Figure 3-2 Memory bank (32-bit) constructed from 16-bit memory

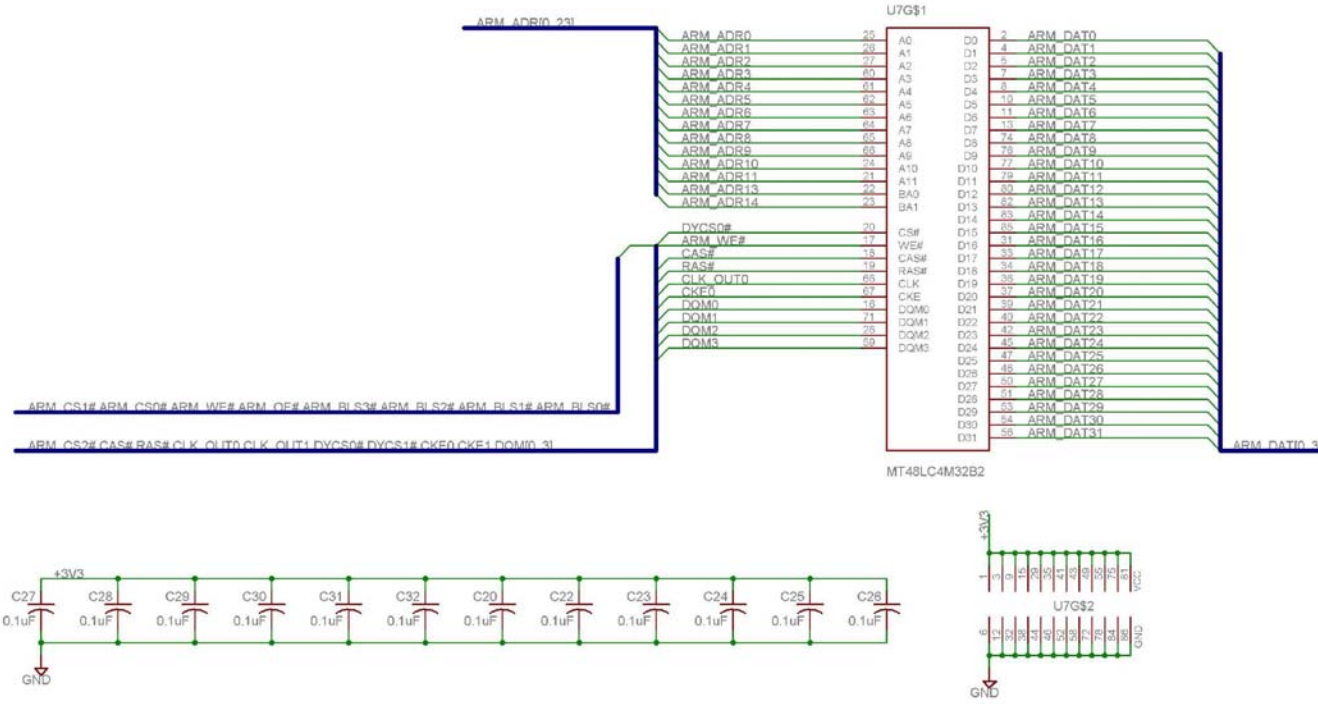

Figure 3-3 Memory bank (32-bit) constructed from 32-bit memory

3.2. SDRAM Address Line Usage

3.2.1 The address mapping for RBC (Row-Bank-Column) SDR-SDRAM with a 32-bit data bus.

Memory type	Row:A[12:0]	Column:A[12:0]	BA1:A[14]	BA0:A[13]
16MB, 1M x 16	[21:11]	[9:2]	-	10
16MB, 2M x 8	[22:12]	[10:2]	11	-
64MB, 2M x 32	[22:12]	[9:2]	11	10
64MB, 4M x 16	[23:12]	[9:2]	11	10

64MB, 8M x 8	[24:13]	[10:2]	11	12
128MB, 4M x 32	[23:12]	[9:2]	11	10
128MB, 8M x 16	[24:13]	[10:2]	11	12
128MB, 16M x 8	[25:14]	[11:2]	13	12
256MB, 8M x 32	[24:12]	[9:2]	11	10
256MB, 16M x 16	[25:13]	[10:2]	11	12
256MB, 32M x 8	[26:14]	[11:2]	13	12
512MB, 32M x 16	[26:14]	[11:2]	13	12
512MB, 64M x 8	[27:15]	[12:2]	13	14

3.2.2 The address mapping for BRC (Bank-Row-Column) SDR-SDRAM with a 32-bit data bus.

Memory type	Row:A[12:0]	Column: A[12:0]	BA1:A[14]	BA0:A[13]
16MB, 1M x 16	[20:10]	[9:2]	21	-
16MB, 2M x 8	[21:11]	[10:2]	-	22
64MB, 2M x 32	[20:10]	[9:2]	21	22
64MB, 4M x 16	[21:10]	[9:2]	23	22
64MB, 8M x 8	[22:11]	[10:2]	23	24
128MB, 4M x 32	[21:10]	[9:2]	23	22
128MB, 8M x 16	[22:11]	[10:2]	23	24
128MB, 16M x 8	[23:12]	[11:2]	25	24
256MB, 8M x 32	[22:10]	[9:2]	23	24
256MB, 16M x 16	[23:11]	[10:2]	25	24
256MB, 32M x 8	[24:12]	[11:2]	25	26
512MB, 32M x 16	[24:12]	[11:2]	25	26
512MB, 64M x 8	[25:13]	[12:2]	27	26

3.2.3 The address mapping for 32-bit wide data bus RBC (Row-Bank-Column) SDR-SDRAM with a 16-bit data bus.

Memory type	Row:A[12:0]	Column:A[12:0]	BA1:A[14]	BA0:A[13]
16MB, 1M x 16	[20:10]	[8:2] a	9	-
16MB, 2M x 8	[21:11]	[9:2] a	-	10
64MB, 4M x 16	[22:11]	[8:2] a	9	10
64MB, 8M x 8	[23:12]	[9:2] a	11	10
128MB, 8M x 16	[23:12]	[9:2] a	11	10
128MB, 16M x 8	[24:13]	[10:2] a	11	12
256MB, 16M x 16	[24:12]	[9:2] a	11	10

256MB, 32M x 8	[25:13]	[10:2] a	11	12
512MB, 32M x 16	[25:13]	[10:2] a	11	12
512MB, 64M x 8	[26:14]	[11:2] a	13	12

3.2.3 The address mapping for 16-bit wide data bus BRC (Bank-Row-Column) SDR-SDRAM with a 16-bit data bus.

Memory type	Row:A[12:0]	Column:A[12:0]	BA1:A[14]	BA0:A[13]
16MB, 1M x 16	[19:9]	[8:2] a	-	20
16MB, 2M x 8	[20:10]	[9:2] a	21	-
64MB, 4M x 16	[20:9]	[8:2] a	21	22
64MB, 8M x 8	[21:10]	[9:2] a	23	22
128MB, 8M x 16	[21:10]	[9:2] a	23	22
128MB, 16M x 8	[22:11]	[10:2] a	23	24
256MB, 16M x 16	[22:10]	[9:2] a	23	24
256MB, 32M x 8	[23:11]	[10:2] a	25	24
512MB, 32M x 16	[23:11]	[10:2] a	25	24
512MB, 64M x 8	[24:12]	[11:2] a	25	26

a A[1] and A[0] are controlled by the SDRAM controller. The SDRAM controller always transfers 32 bits of data at a time. For chip selects with a 16-bit wide data bus the SDRAM controller performs two transfers, a column transfer with the lowest bit set to 0 and a column transfer with the lowest bit set to 1.

3.3. Generic SDRAM initialization

Software must initialize the EMC and each of the dynamic memories connected to the controller. Check the dynamic memory data sheet for the start up procedure. A generic example initialization sequence is:

- Wait 100ms after the power is applied and the clocks have stabilized.
- Set the SDRAM Initialization (I) value to NOP in the EMCDynamicControl Register. This automatically issues a NOP command to the SDRAM memories.
- Wait 200ms.

- Set the SDRAM Initialization (I) value to PALL in the EMCDynamicControl Register. This automatically issues a precharge all instruction (PRE-ALL) to the SDRAM memories. This precharges all banks and places the device into the all banks idle state.
- Perform a number of refresh cycles by writing 1 into the EMCDynamicRefresh Register. This provides a memory refresh every 16 AHB clock cycles.
- Wait until eight SDRAM refresh cycles have occurred (128 AHB clock cycles).
- Program the operational value into the EMCDynamicRefresh Register.
- Program the operational value into the EMCDynamicRasCas Register.
- Program the operational values into the EMCDynamicConfig Register. The buffers must be disabled during initialization.
- Set the SDRAM initialization value (I) to MODE in the EMCDynamicControl Register.
- Program the SDRAM memories mode register. The mode register enables the following parameters to be programmed:
• burst length
• burst type
• CAS latency
• operating mode
• write burst mode.
A read transaction from the SDRAM memory programs the mode register. The address of the transfer contains the value to be programmed. The mapping from the address bus to the SDRAM memories address lines depends on the address mapping value selected in the EMCDynamicConfig Register. The row address bits contain the value to be programmed. The bank select signals BA0 and BA1 must both be 0 to program the mode register.
- Set the SDRAM initialization value (I) to NORMAL in the EMCDynamicControl Register.
- Enable the buffers in EMCDynamicConfig Register. The SDRAM is now ready for normal operation.

3.3.1 Typical SDRAM initialization

```
/*****************************************************************
 * Figure 3-1 Memory bank (16-bit) constructed from 16-bit memory  *
 * Figure 3-2 Memory bank (32-bit) constructed from 16-bit memory  *
 * Micron MT48LC16M16A2TG-75                                       *
 * Figure 3-3 Memory bank (32-bit) constructed from 32-bit memory  *
 * Micron MT48LC4M32B2P-7                                          *
 * LPC2468 CCLK : 60MHz                                            *
 * LPC2460 CCLK : 72MHz                                            *
```

```
 ************************************************************************/

    EMCCONTROL = 0x00000001;
    /* Enabled, Normal memory map, Normal mode */
    EMCCONFIG = 0x00000000;
    EMCDYNAMICCFG0 = 0x00005500;
    /* 128MB, 4Mx32, 4 banks, row=12, column=8 */
#ifdef SDRAM_CS1_CONFIG_32BIT
    EMCDYNAMICCFG1 = 0x00005680;
    /* 256MB, 16Mx16, 4 banks, row=13, column=9 */
#endif
#ifdef SDRAM_CS1_CONFIG_16BIT
    EMCDYNAMICCFG1 = 0x00001680;
    /* 256MB, 16Mx16, 4 banks, row=13, column=9 */
#endif

    EMCDYNAMICRASCAS0 = 0x00000202;
    /* 2 RAS, 2 CAS latency */
    EMCDYNAMICRASCAS1 = 0x00000202;
    /* 2 RAS, 2 CAS latency */

    EMCDINAMICRDCFG = 0x00000001;
    /* Command delayed strategy, using EMCCLKDELAY */
    EMCDYNAMICRP = 0x00000001;
    /* (n + 1) -> 2 clock cycles */
    EMCDYNAMICRAS = 0x00000003;
    /* (n + 1) -> 4 clock cycles */
    EMCDYNAMICSREX = 0x00000005;
    /* (n + 1) -> 6 clock cycles */
    EMCDYNAMICAPR = 0x00000001;
    /* (n + 1) -> 2 clock cycles */
    EMCDYNAMICDAL = 0x00000005;
    /* (n) -> 5 clock cycles */
    EMCDYNAMICWR = 0x00000001;
    /* (n + 1) -> 2 clock cycles */
    EMCDYNAMICRC = 0x00000005;
    /* (n + 1) -> 6 clock cycles */
    EMCDYNAMICRFC = 0x00000005;
    /* (n + 1) -> 6 clock cycles */

    EMCDYNAMICXSR = 0x00000005;
    /* (n + 1) -> 6 clock cycles */
    EMCDYNAMICRRD = 0x00000001;
    /* (n + 1) -> 2 clock cycles */
    EMCDYNAMICMRD = 0x00000001;
    /* (n + 1) -> 2 clock cycles */
    EMCDINAMICCTRL = 0x00000183;
    /* Issue NOP command */
    delayMs(100);
    EMCDINAMICCTRL = 0x00000103;
```

```
    /* Issue PALL command */
    EMCDINAMICRFR = 0x00000002;
    /* (n * 16) -> 32 clock cycles */
    delayMs(100);
    EMCDINAMICRFR = 0x00000023;
    /* (n * 16) -> 560 clock cycles */
    /* -> 7.7778uS at 72MHz <= 7.8125uS (64ms / 8192 row) */
    EMCDYNAMICRASCAS0 = 0x00000202;
    /* 2 RAS, 2 CAS latency */
    EMCDYNAMICRASCAS1 = 0x00000202;
    /* 2 RAS, 2 CAS latency */
    EMCDYNAMICCFG0 = 0x00005500;
    /* 128MB, 4Mx32, 4 banks, row=12, column=8 */
#ifdef SDRAM_CS1_CONFIG_32BIT
    EMCDYNAMICCFG1 = 0x00005680;
    /* 256MB, 16Mx16, 4 banks, row=13, column=9 */
#endif
#ifdef SDRAM_CS1_CONFIG_16BIT
    EMCDYNAMICCFG1 = 0x00001680;
    /* 256MB, 16Mx16, 4 banks, row=13, column=9 */
#endif
    EMCDINAMICCTRL = 0x00000083;
    /* Issue MODE command */
    dwtemp = *((volatile INT32U *)(SDRAM_CS0_BASE | 0x00008800));
    /* 4 burst, 2 CAS latency */
#ifdef SDRAM_CS1_CONFIG_32BIT
    dwtemp = *((volatile INT32U *)(SDRAM_CS1_BASE | 0x00011000));
    /* 4 burst, 2 CAS latency */
#endif
#ifdef SDRAM_CS1_CONFIG_16BIT
    wtemp = *((volatile INT16U *)(SDRAM_CS1_BASE | 0x00008C00));
    /* 8 burst, 2 CAS latency */
#endif
    EMCDINAMICCTRL = 0x00000000;
    /* Issue NORMAL command */
    EMCDYNAMICCFG0 = 0x00085500;
    /* 128MB, 4Mx32, 4 banks, row=12, column=8 */
#ifdef SDRAM_CS1_CONFIG_32BIT
    EMCDYNAMICCFG1 = 0x00085680;
    /* 256MB, 16Mx16, 4 banks, row=13, column=9 */
#endif
#ifdef SDRAM_CS1_CONFIG_16BIT
    EMCDYNAMICCFG1 = 0x00081680;
    /* 256MB, 16Mx16, 4 banks, row=13, column=9 */
#endif
```

3.4 Transaction latency formulae

The latency formulas use the following parameters:

PrechargeLatency: SDRAM memory device precharge latency. This parameter is required if the read transaction is to a row that is different to the one already open.

ActivateLatency: SDRAM memory devices activate latency (RAS to CAS latency). This parameter is required if the read transaction is to a precharged bank.

CASLatency: SDRAM memory device CAS latency. This parameter is required for all read transactions.

BurstLength: AHB burst length. This parameter is not required for write transactions.

AlignedIncr: Aligned incrementing AHB burst. If the AHB burst is an aligned INCR, INCR4, INCR8, or INCR16 burst, this is true (1). Otherwise this is false (0).

Arbitration: Arbitration delay. For unaligned INCR4, INCR8, INCR16, WRAP4, WRAP8, and WRAP16 AHB bursts this parameter is false (0).

n: Number of bursts submitted to the memory controller.

3.4.1 Read transfer latency

3.4.1.1 Single read transfer latency formula

ReadLatency = 7 + PrechargeLatency + ActivateLatency + CASLatency - (2 x AlignedIncr) + BurstLength

3.4.1.2 Back-to-back read transfer latency formula

ReadLatency = n(7 + PrechargeLatency + ActivateLatency + CASLatency – [2 x AlignedIncr + BurstLength]) + (n-1) x Arbitration

3.4.2 Write transfer latency

3.4.2.1 Single write transfer latency formula

MemoryWriteLatency = (1 + BurstLength) + (4 + PrechargeLatency + ActivateLatency + BurstLength)

3.4.2.2 Back-to-back write transfer latency formula

3.4.2.2.1 Back-to-back write transfer memory latency formula if the writes are not in-page

MemoryWriteLatency = n(2 + BurstLength) - 1 + n(4 + PrechargeLatency + ActivateLatency + BurstLength)

3.4.2.2.2 Back-to-back write transfer memory latency formula if the writes are in-page

MemoryWriteLatency = n(2 + BurstLength) -1 + (4 + PrechargeLatency + ActivateLatency + n x BurstLength)

3.5 SDRAM Signal Routing Considerations

The following are general guidelines for designing an SDRAM interface with LPC24xx.

- Layout for the SDRAM should begin by placing the SDRAM devices as close as possible to the processor.
- Keep the SDRAM clock (CLKOUTx) and the SDRAM control lines as short as possible.
- Keep the address and data lines as short as possible.
- The series resistor (if any) placement is to be located near the processor. The need and specific value of series termination resistors on the signals is best determined by simulation using IBIS models and the specific design PCB layout.
- To support maximum speeds, reasonable SDRAM loading constraints must be followed. The user must consider all the devices connected on the different buses to calculate the system load.
- Use sufficient decoupling scheme for memory devices. It is recommended to use low ESR 0.01 µF and 0.1 µF decoupling capacitors in parallel. An additional 0.001 µF decoupling capacitor is recommended to minimize ground bounce and to filter high frequency noise.

3.6 Power consumption

3.6.1 LPC2468 with one 32-bit SDRAM

Symbol	Parameter	Conditions	Typ	Unit
$I_{DD(3V3)}$	supply current	$V_{DD(3V3)}$ =3.3V; T_{amb} = 25°C; [1]	24.0	mA
$I_{DD(DCDC)(3V3)}$	DC-DC coverter	$V_{DD(DCDC)(3V3)}$ =3.3V; T_{amb} = 25°C;	73.3	mA

	supply current	[1]		
$I_{DD(SDRAM)}$	supply current	$V_{DD(SDRAM)}$ =3.3V; T_{amb} = 25°C; [1]	35.6	mA

[1] LPC2468 CCLK = 60MHz; PCLK = ¼ CCLK;
Peripherals enabled are default after reset;
Code executed from internal flash writes various patterns to the SDRAM and verifies;
Figure 3-3 memory bank (32-bit) constructed from 32-bit memory;

3.6.2 LPC2468 with one 16-bit SDRAM

Symbol	Parameter	Conditions	Typ	Unit
$I_{DD(3V3)}$	supply current	$V_{DD(3V3)}$ =3.3V; T_{amb} = 25°C; [2]	21.2	mA
$I_{DD(DCDC)(3V3)}$	DC-DC coverter supply current	$V_{DD(DCDC)(3V3)}$ =3.3V; T_{amb} = 25°C; [2]	73.3	mA
$I_{DD(SDRAM)}$	supply current	$V_{DD(SDRAM)}$ =3.3V; T_{amb} = 25°C; [2]	26.4	mA

[2] LPC2468 CCLK = 60MHz; PCLK = ¼ CCLK;
Peripherals enabled are default after reset;
Code executed from internal flash writes various patterns to the SDRAM and verifies;
Figure 3-1 memory bank (16-bit) constructed from 16-bit memory;

3.6.3 LPC2468 with two 16-bit SDRAM

Symbol	Parameter	Conditions	Typ	Unit
$I_{DD(3V3)}$	supply current	$V_{DD(3V3)}$ =3.3V; T_{amb} = 25°C; [3]	24.0	mA
$I_{DD(DCDC)(3V3)}$	DC-DC coverter supply current	$V_{DD(DCDC)(3V3)}$ =3.3V; T_{amb} = 25°C; [3]	73.3	mA
$I_{DD(SDRAM)}$	supply current	$V_{DD(SDRAM)}$ =3.3V; T_{amb} = 25°C; [3]	35.6	mA

[3] LPC2468 CCLK = 60MHz; PCLK = ¼ CCLK;
Peripherals enabled are default after reset;
Code executed from internal flash writes various patterns to the SDRAM and verifies;
Figure 3-2 memory bank (32-bit) constructed from 16-bit memory;

3.6.4 LPC2460 with one 16-bit flash and one 32-bit SDRAM

Symbol	Parameter	Conditions	Typ	Unit
$I_{DD(3V3)}$	supply current	$V_{DD(3V3)}$ =3.3V; T_{amb} = 25°C; [4]	9.2	mA
$I_{DD(DCDC)(3V3)}$	DC-DC coverter supply current	$V_{DD(DCDC)(3V3)}$ =3.3V; T_{amb} = 25°C; [4]	64.4	mA
$I_{DD(FLASH)}$	supply current	$V_{DD(FLASH)}$ =3.3V; T_{amb} = 25°C;	46.7	mA

		[4]		
$I_{DD(SDRAM)}$	supply current	$V_{DD(SDRAM)}$ =3.3V; T_{amb} = 25°C; [4]	18.8	mA

[4] LPC2460 CCLK = 72MHz; PCLK = ¼ CCLK;
Peripherals enabled are default after reset;
Code executed from internal flash writes various patterns to the SDRAM and verifies;
Figure 2-12 memory bank (16-bit) constructed from 16-bit memory;
Figure 3-3 memory bank (32-bit) constructed from 32-bit memory;

3.6.5 LPC2460 with one 16-bit flash and one 16-bit SDRAM

Symbol	Parameter	Conditions	Typ	Unit
$I_{DD(3V3)}$	supply current	$V_{DD(3V3)}$ =3.3V; T_{amb} = 25°C; [5]	9.2	mA
$I_{DD(DCDC)(3V3)}$	DC-DC coverter supply current	$V_{DD(DCDC)(3V3)}$ =3.3V; T_{amb} = 25°C; [5]	64.4	mA
$I_{DD(FLASH)}$	supply current	$V_{DD(FLASH)}$ =3.3V; T_{amb} = 25°C; [5]	46.7	mA
$I_{DD(SDRAM)}$	supply current	$V_{DD(SDRAM)}$ =3.3V; T_{amb} = 25°C; [5]	10.4	mA

[5] LPC2460 CCLK = 72MHz; PCLK = ¼ CCLK;
Peripherals enabled are default after reset;
Code executed from internal flash writes various patterns to the SDRAM and verifies;
Figure 2-12 memory bank (16-bit) constructed from 16-bit memory;
Figure 3-1 memory bank (16-bit) constructed from 16-bit memory;

3.6.6 LPC2460 with one 16-bit flash and two 16-bit SDRAM

Symbol	Parameter	Conditions	Typ	Unit
$I_{DD(3V3)}$	supply current	$V_{DD(3V3)}$ =3.3V; T_{amb} = 25°C; [6]	9.2	mA
$I_{DD(DCDC)(3V3)}$	DC-DC coverter supply current	$V_{DD(DCDC)(3V3)}$ =3.3V; T_{amb} = 25°C; [6]	64.4	mA
$I_{DD(FLASH)}$	supply current	$V_{DD(FLASH)}$ =3.3V; T_{amb} = 25°C;	46.7	mA

		[6]		
$I_{DD(SDRAM)}$	supply current	$V_{DD(SDRAM)}$ =3.3V; T_{amb} = 25°C; [6]	18.8	mA

[6] LPC2460 CCLK = 72MHz; PCLK = ¼ CCLK;
Peripherals enabled are default after reset;
Code executed from internal flash writes various patterns to the SDRAM and verifies;
Figure 2-12 memory bank (16-bit) constructed from 16-bit memory;
Figure 3-2 memory bank (32-bit) constructed from 16-bit memory;