4 SYSTEM DESIGN

OVERVIEW

The KS32C5000(A)/50100, SNASUMG's 16/32-bit RISC microcontroller is cost-effective and high performance microcontroller solution for Ethernet-based system. The integrated on-chip functions of KS32C5000(A)/50100 are

- 8K-byte unified cache/SRAM
- Ethernet controller
- HDLC controller
- UART
- Timer
- I2C
- Programable I/O ports
- Interrupt controller

Therefore, you can use KS32C5000(A)/50100 as amount types of system.

APPLICABLE SYTEM WITH KS32C5000(A)50100

If your product need to be networked, the KS32C5000(A)/50100, SNASUMG's 16/32-bit RISC microcontroller can be reduce your system cost. There are sample system, it can be designed with KS32C5000(A)/50100.

- Managed hub/Switch
- Router and bridge
- ISDN Router /TA
- ADSL Router
- Home/Industry security
- Cable modem
- Digital camera
- Home/Industry security
- Cable modem
- Internet FAX/Internet phone
- Network printer
- Game machine
- Any system with network interface

MEORY INTERFACE DESIGN

ADDRESS BUS GENERATION

The KS32C5000(A)/50100 address bus generation is based on the required data bus width of each memory bank, The internal system address bus is shifted out to an external address bus, ADDR[21:0]. This means that memory control signals such as nRAS[3:0], nCAS[3:0], nECS[3:0],nRCS[5:0], nWBE[3:0] are generated by the system manager according to a pre-configured external memory scheme (see Table 4-1,and Figure 4-1).

Table 4-1. Address Bus Generation Guidlines

Data Bus Width	External Address Pins, ADDR[21:0]	Accessible Memory Size
8-bit (Byte)	A21-A0 (internal)	4M bytes
16-bit (Half-Word)	A22-A1 (internal)	4M half-words
32-bit (Word)	A23-A2 (internal)	4M words

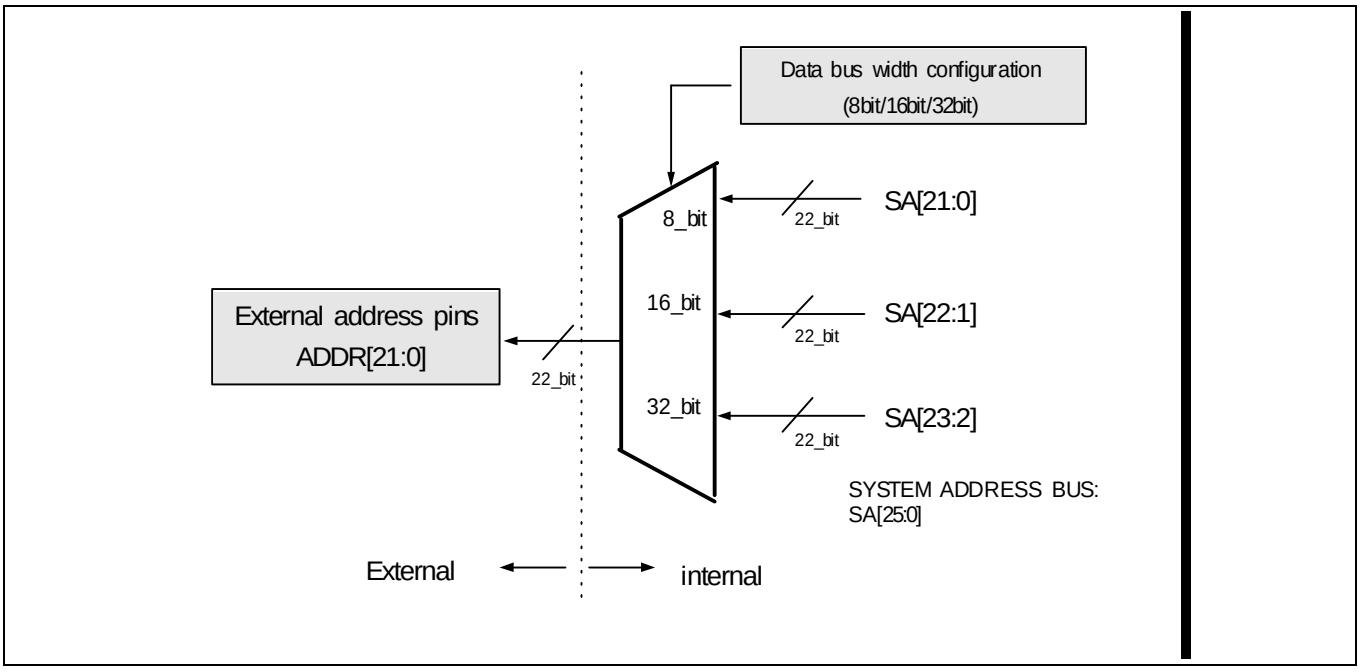


Figure 4-1. External address bus generation(ADDR[21:0])

BOOT ROM DESIGN

When system reset, a KS32C5000(A)/50100A access 0x00000000 address. And KS32C5000(A)/50100 should be configure some system variable after reset. Therefore this special code (BOOT ROM image) should be located on address 0x00000000. A boot ROM can have a various width of data bus, and it is controlled by B0SIZE[1:0] pins.

Table 4-2. Data Bus Width for ROM Bank 0

B0SIZE[1:0]	Data Bus Width
00	Reserved
01	8-bit (byte)
10	16-bit (half-word)
11	32-bit (word)

ONE BYTE BOOT ROM DESIGN

A design with one byte boot ROM is shown in Figure 4-2.

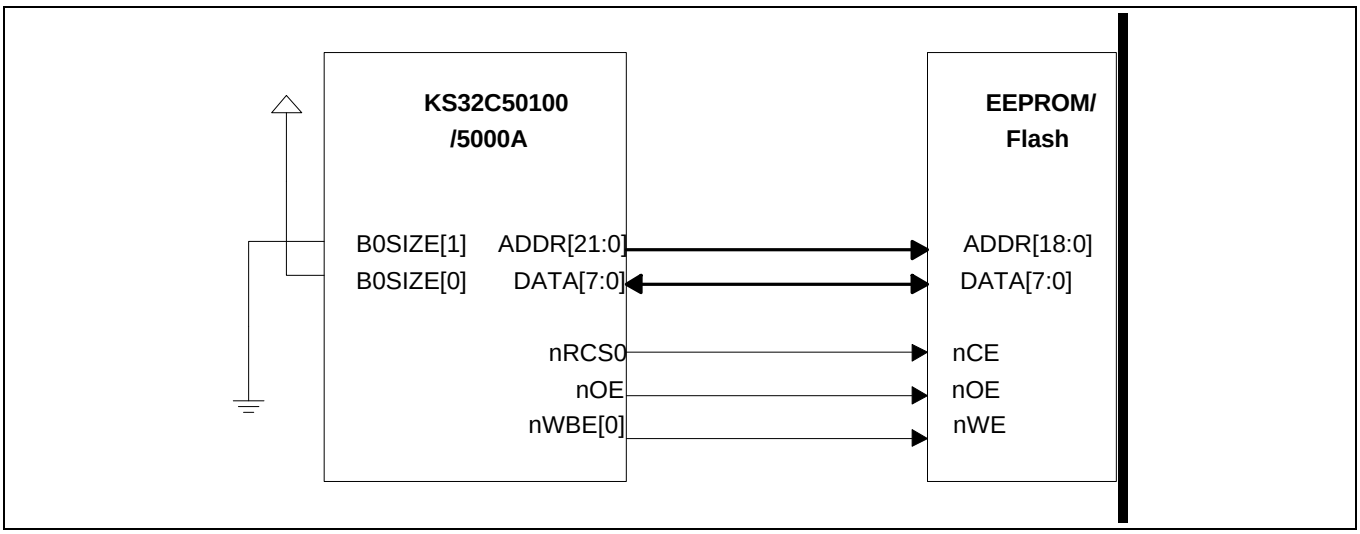


Figure 4-2. One Byte Boot ROM Design

MAKE AND FUSING ONE BYTE ROM IMAGE

When make one byte ROM image, you can use the binary file that maded from compile and link.

HALF-WORD BOOT ROM DESIGN WITH BYTE EEPROM/FLASH

A design with half-word boot ROM with byte EEPROM/Flash is shown in Figure 4-3.

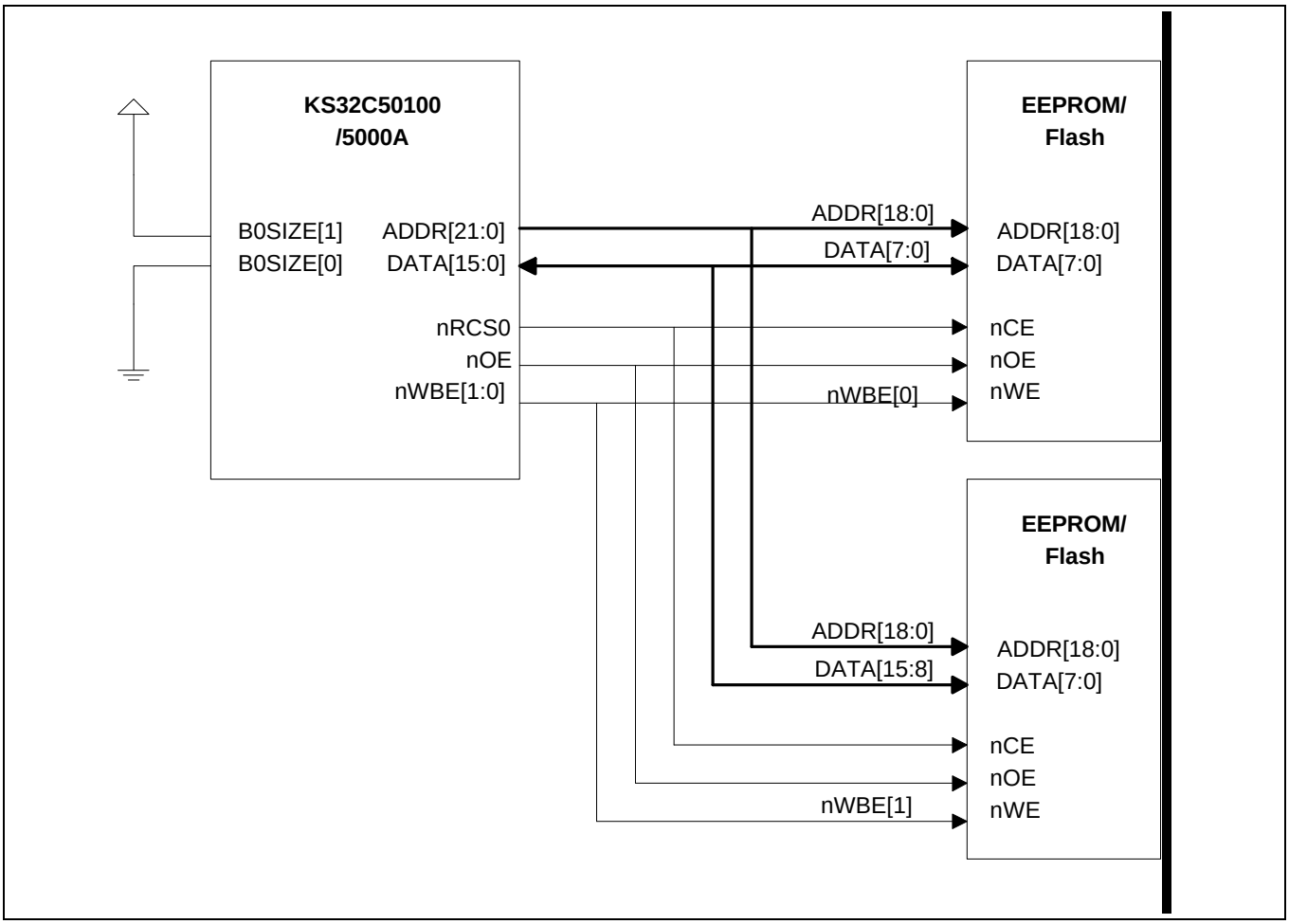


Figure 4-3. The Half-Word Boot ROM Design with Byte EEPROM/Flash

MAKE AND FUSING HALF-WORD ROM IMAGE WITH BYTE EEPROM/FLASH

When make half-word ROM image, you can split two image files, EVEN and ODD. When you use our device KS32C5000(A)/50100 as big-endian mode, then you should swap EVEN and ODD. Because KS32C5000(A)/50100 has little-endian data bus structure.

For example :

split2 rom.bin
Output even file name : rom.ODD
Out odd file name : rom.EVN

But, when you use our device as little-endian mode, then there is no need swap of EVEN and ODD.

HALF-WORD BOOT ROM DESIGN WITH HALF-WORD EEPROM/FLASH

A design with half-word boot ROM with byte EEPROM/Flash is shown in Figure 4-4.

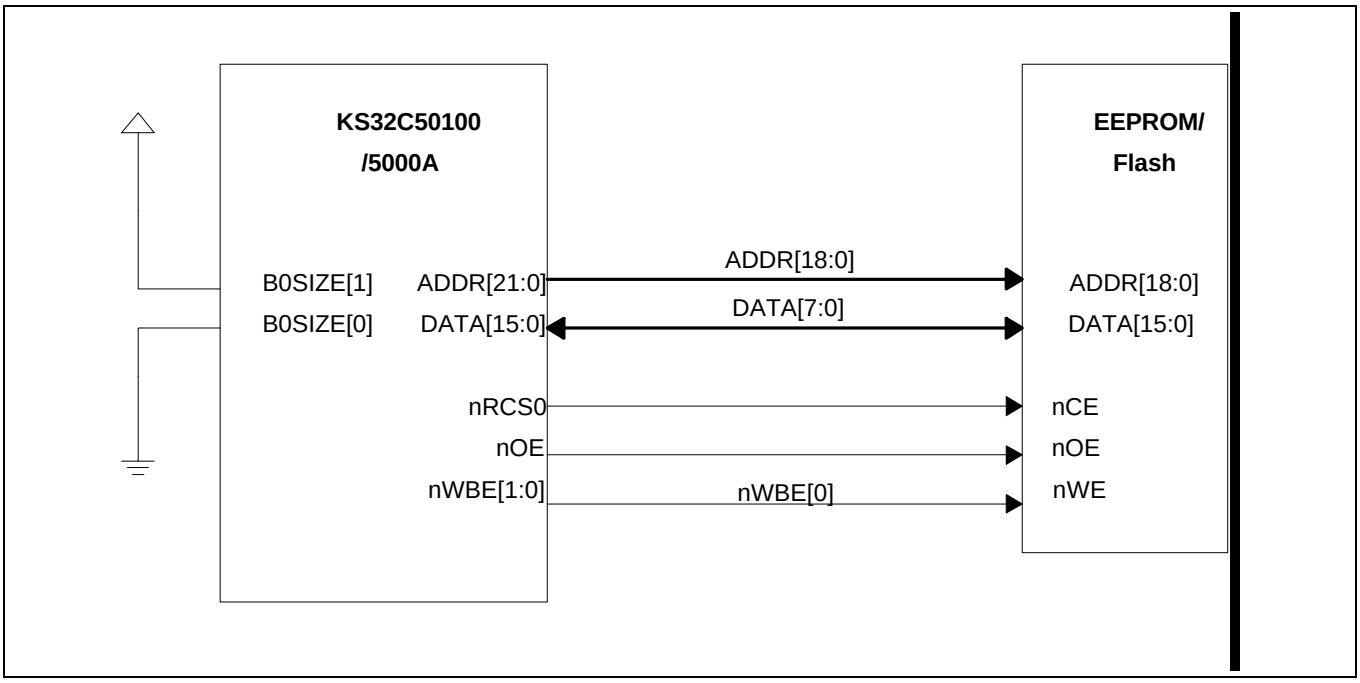


Figure 4-4. The Half-Word Boot ROM Design with Half-Word EEPROM/Flash

MAKE AND FUSING HALF-WORD ROM IMAGE WITH BYTE EEPROM/FLASH

When make half-word ROM image, you should swap EVEN and ODD ROM image on ROM writer. Because KS32C5000(A)/50100 has little-endian data bus structure.

But, when you use our device as little-endian mode, then there is no need swap of EVEN and ODD.

MEMORY BANKS DESIGN AND CONTROL

The KS32C50100 has 6 bank ROM/SRAM banks (ROM0 bank for boot ROM), 4 EDO/Synchronous DRAM banks, and 4 external I/O banks,. The KS32C5000(A) has 6 bank ROM/SRAM banks(ROM0 bank for boot ROM), 4 EDO DRAM banks, and 4 external I/O banks, The system manager on KS32C5000(A)/50100 can control access time, data bus width, and base/end point, for each banks by S/W. The access time and base/end point of ROM/SRAM banks is controlled by ROMCON0-5 control register on system manager. The base point of external I/O banks is controlled by REFEXTCON control register, and access time for each external I/O banks is controlled by EXTCON0-1 control register. And the access time and base/end point of DRAM banks is controlled by DRAMCON0-3 control register.

The care is need for attach the memory and external I/O to KS32C5000(A)/50100's, Even though CPU is configured for the big-endian memory accessing. However, to connect with the external memory, KS32C5000(A)/50100 is different with the other normal big-endian microcontrollers because of its internal byte-swap mechanism. The data connection with external memory should be done as the "little endian" way. That is, the byte 0 of the external memory(that is, the most significant byte of a word)should be connected to the controller's data pin[7:0], byte 1 to data pin[15:8], byte 2 to data pin[23:16], and byte 3 to data pin[31:24].

When you use KS32C50100 with SDRAM, you can enable SDRAM mode by SYSCFG register bit 31.

The data bus width for each ROM/SRAM banks, external I/O banks, and DRAM banks is controlled by EXTDBWTH data bus width register on system manager, except ROM bank 0.

The ROM bank 0 is used for boot ROM bank, therefore bank 0 is controlled by H/W, B0SIZE[1:0] is used for this purpose(see table 3-2).

The control of EXTDBWTH,ROMCON0-5, DRAMCON0-4,REFEXTCON is performed when system reset, by special command, LDMIA and STMIA. sample code for special register configuration is described below.

ROM/SRAM BANKS DESIGN

The ROM/SRAM banks 1-5, can have a various width of data bus, and the bus width is controlled by S/W, A EXTDBWTH special register set. A sample design for ROM/SRAM bank 1-5 is shown in Figure 4-5, Figure 4-6, and Figure 4-7.

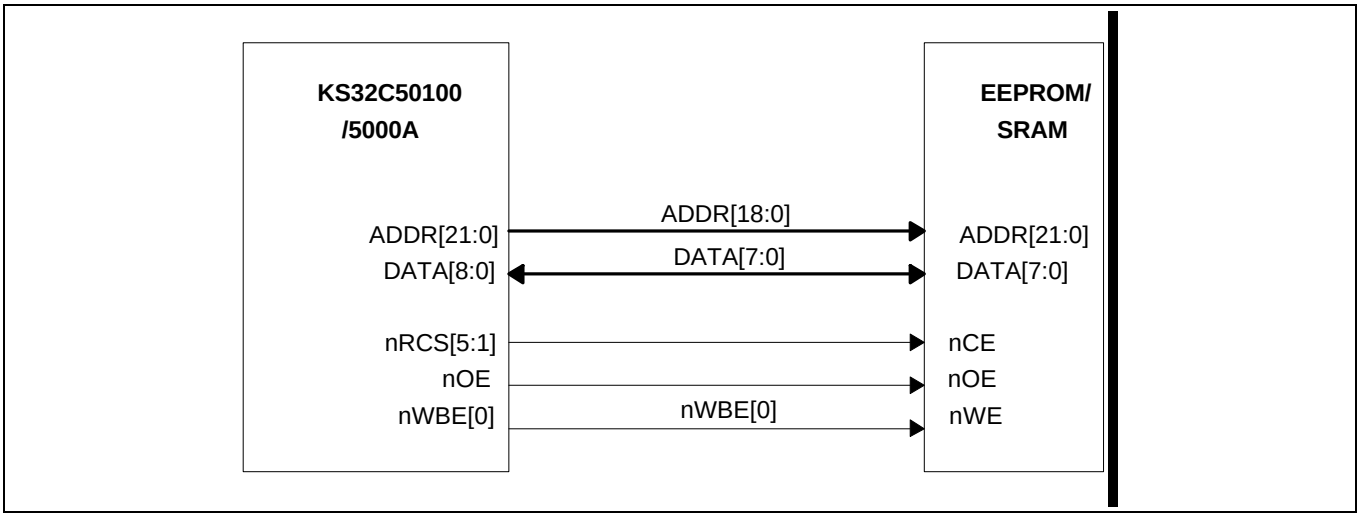


Figure 4-5. One-byte EEPROM/SRAM Banks Design

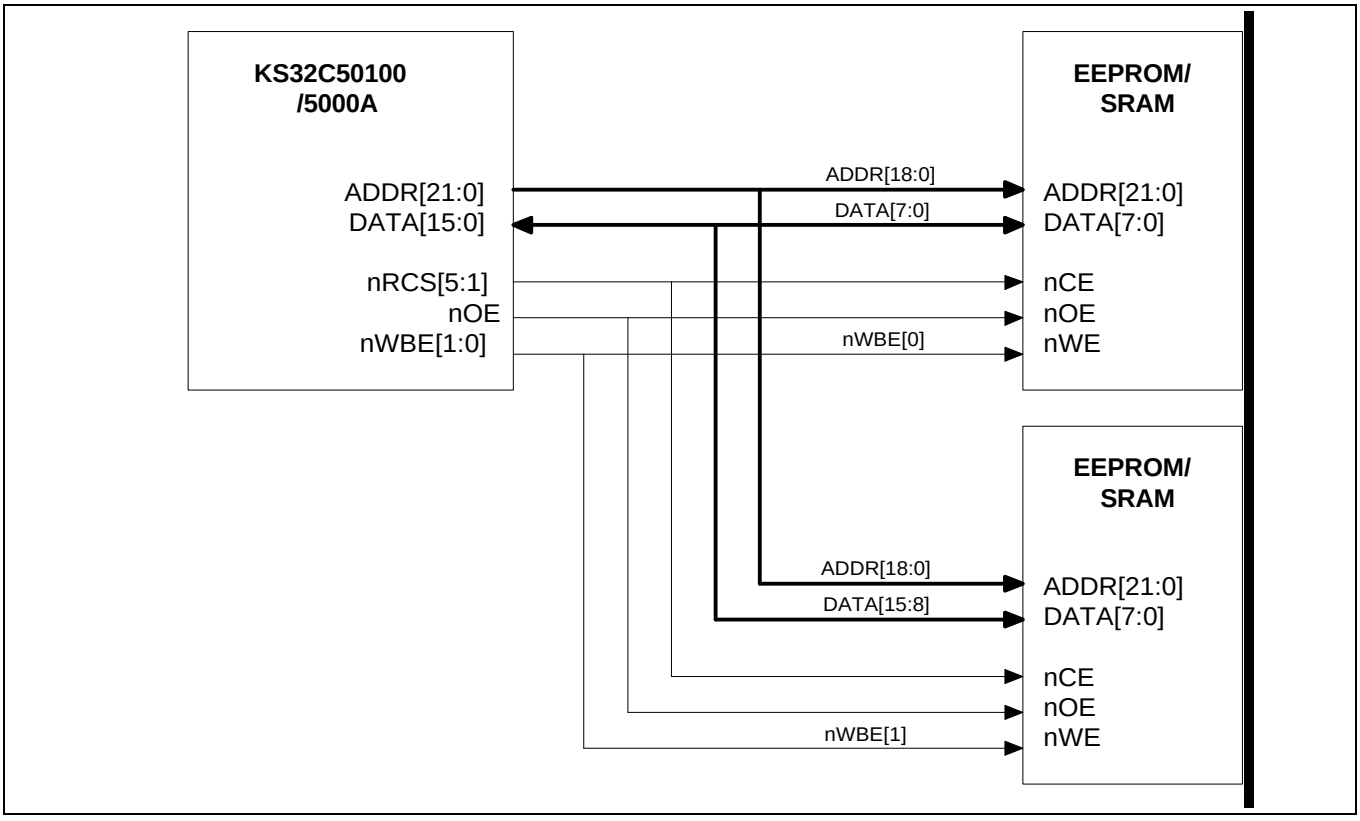


Figure 4-6. Half-word EEPROM/SRAM Banks Design

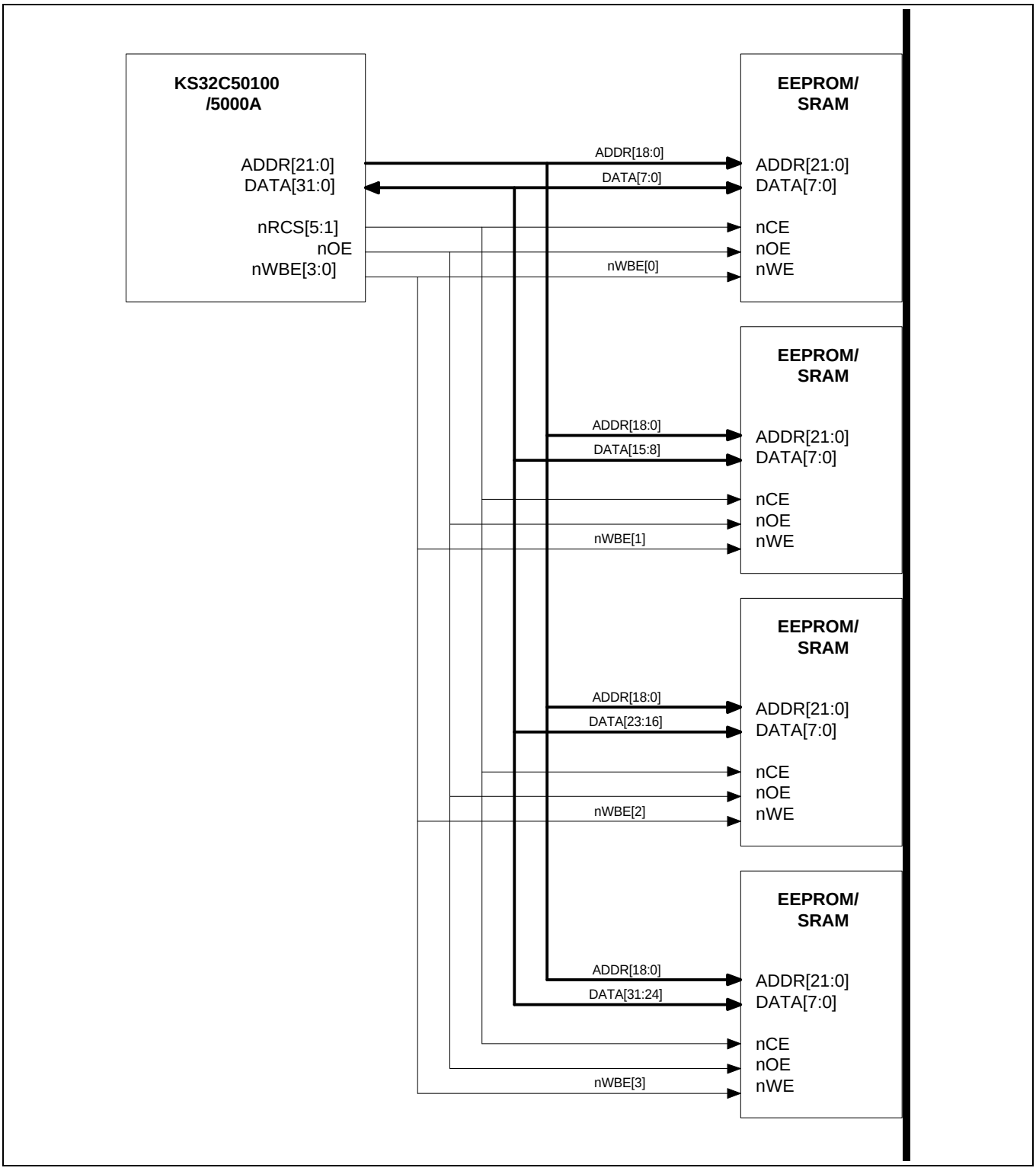


Figure 4-7. Word EEPROM/SRAM banks design

ROM/SRAM BANK 5 DESIGN WITH MULTIPLEXED ADDRESS/DATA BUS

The KS32C5000(A)/50100 supports multiplexed address/data bus for low-cost chips which require multiplexed bus. But, ROM Bank5 muxed bus mode was added to KS32C5000(A)/50100 for the purpose of using special ASIC chip interface not commercial device. So, there are some limits using this mode, The address bus generation scheme of KS32C5000(A)/50100 is shifted address bus according to the data bus width of each memory bank.

The ROM bank 5 restriction is listed in below.

- Data bus width is fixed to 32bit. So, if you want to attach 8-bit device, then you should read all 32-bit data and then process by S/W to get 8-bit real data.
- Multiplexed address signal has same configuration with stand-alone address bus, so it can make exhaust accessable addressing area than real address area. That is, If you configure the data bus width of ROM Bank5 to word in muxed bus mode, then the muxed address will be appeared on the data bus shifted by two bits. Using this mode in spite of the loss address space less then 64Mbyte, you have to use trick as follows

  (ROM Bank5 offset address + access address << n) + byte offset.

  Where, n is 2 in case of 32bit data bus width, 1 in case of 16bits, 0 in case of 8bits. byte offset have to be 0 for byte access, 1 for halfword access, 2 for word access. For example, you want to access half-word 0x123 address, translated address is as follows:

  0x123 << 2 + 3 = 0x48f
- You can use the MCLKO clock output as a target device.

EDO DRAM BANKS DESIGN FOR KS32C5000(A)/50100

The DRAM banks 0-4, can have a various width of data bus, and the bus width is controlled by S/W, A EXTDBWTH special register set. A sample design for DRAM bank 0-3 is shown in Figure 4-8, Figure 4-9, and Figure 4-10.

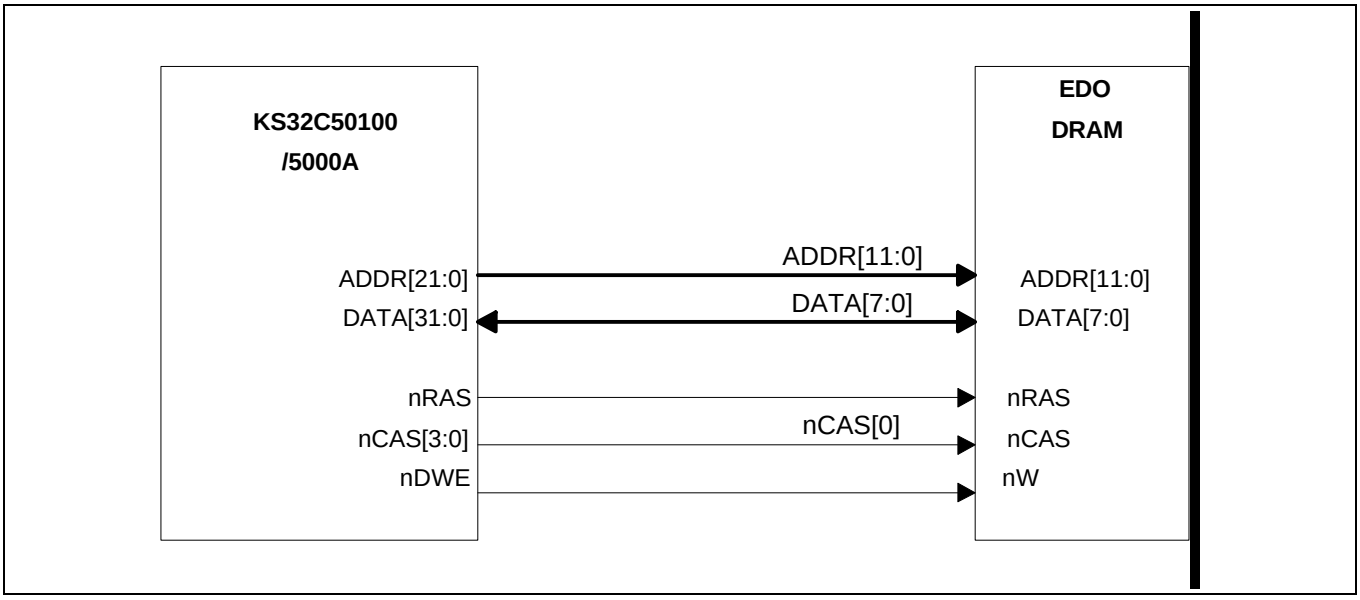


Figure 4-8. 1-byte EDO/Normal DRAM Banks Design

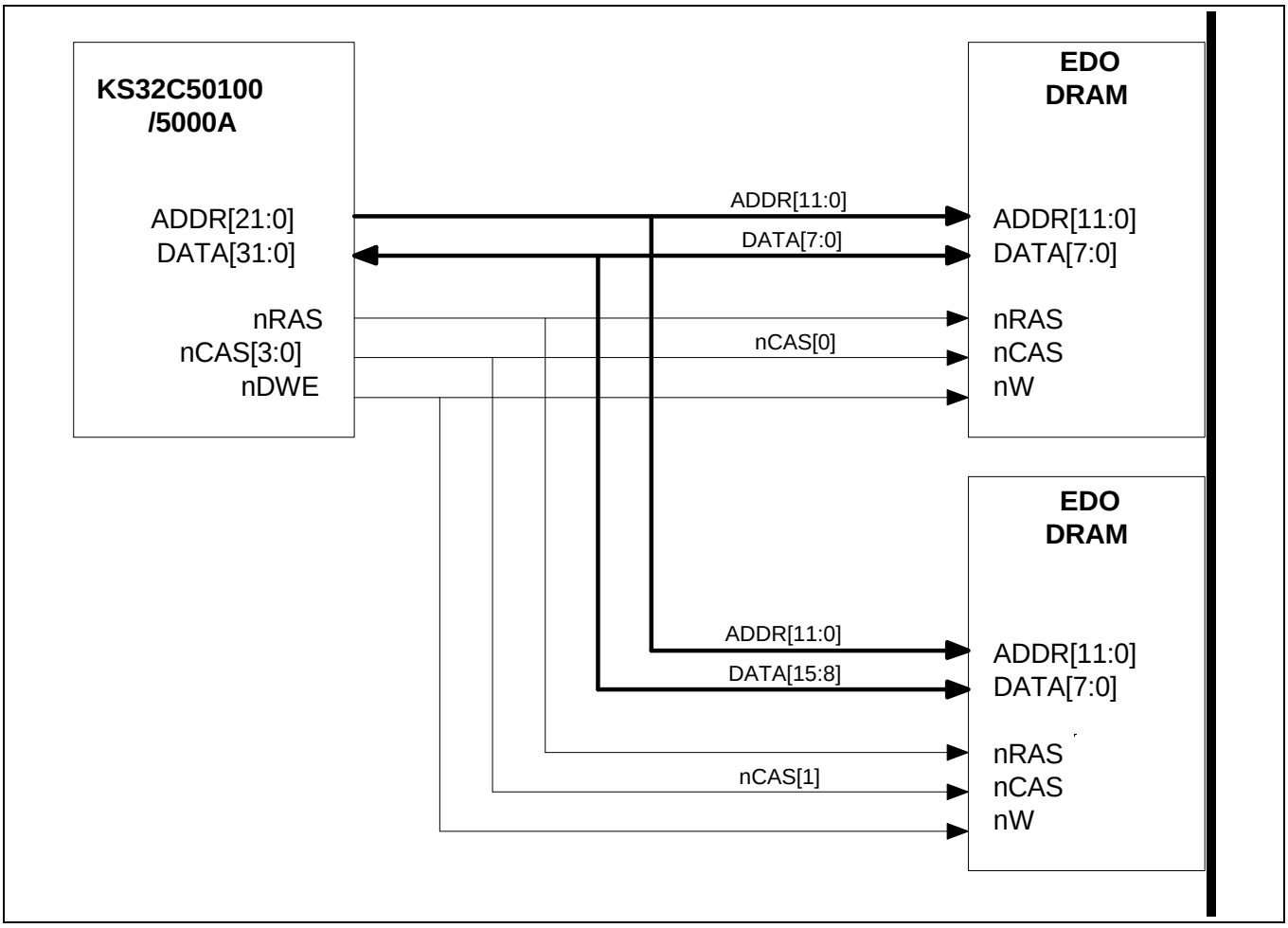


Figure 4-9. Half-word EDO/Normal DRAM Banks Design

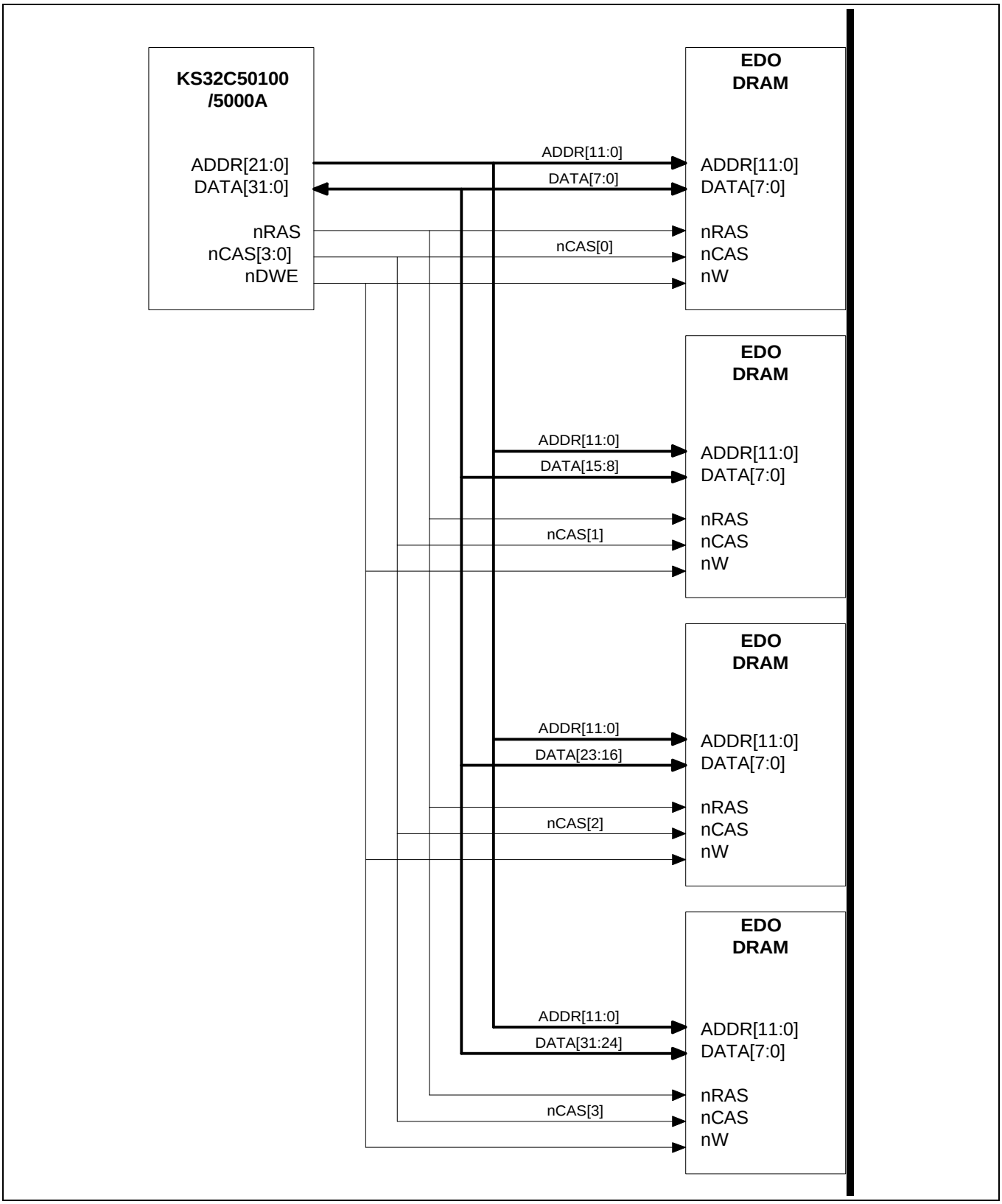


Figure 4-10. Word EDO/Normal DRAM Banks Design

SYNCHRONOUS DRAM BANKS DESIGN FOR KS32C50100

The KS32C50100 Synchronous DRAM interface features are as follows :

- Maximum column address of SDRAM: 11bit
- CAS latency: 2 cycle
- Burst lengyh support: only 1 burst mode(single mode) support
- Burst type: sequential
- Support auto refresh

KS32C50100 can support below Samsung' s SDRAM configuration for each bank.

- 16M SDRAM
  - ✓ KM44S4020C: 2M x 4bit with 2 banks
  - ✓ KM48S2020C: 1M x 8bit with 2 banks
  - ✓ KM416S1020C: 512K x 16bit x 2 banks
- 64M 2 banks SDRAM
  - ✓ KM44S16020B: 8M x 4bit with 2 banks
  - ✓ KM48S8020B: 4M x 8bit with 2 banks
  - ✓ KM416S4020B: 2M x 16bit x 2 banks
- 64M 4 banks SDRAM
  - ✓ KM44S16030B/C: 4M x 4bit with 4 banks
  - ✓ KM48S8030B/C: 2M x 8bit with 4 banks
  - ✓ KM416S4030B/C: 1M x 16bit x 4 banks
- 2Mx32 4 banks SDRAM
  - ✓ KM432S2030B: 512K x 32bit with 4 banks

When you desgn with SDRAM, you should enable MCLKO pin for SDRAM sync. Clock. And Address 10 is used for bank address (BA) select address for SDRAM.

The required SDRAM interface pin is CKE, SDCLK, nSDCS[3:0], nSDCAS, nSDRAS, DQM[3:0], ADDR[10]/AP. The sample design with SDRAM is shown in Figure 4-11, and Figure 4-12.

When you design with 5 V device with 3.3 V SDRAM, then you need to protect the 3.3 V SDRAM from 5 V device. That is, data bus should be has 10 ohm damping resister from 5 V device.

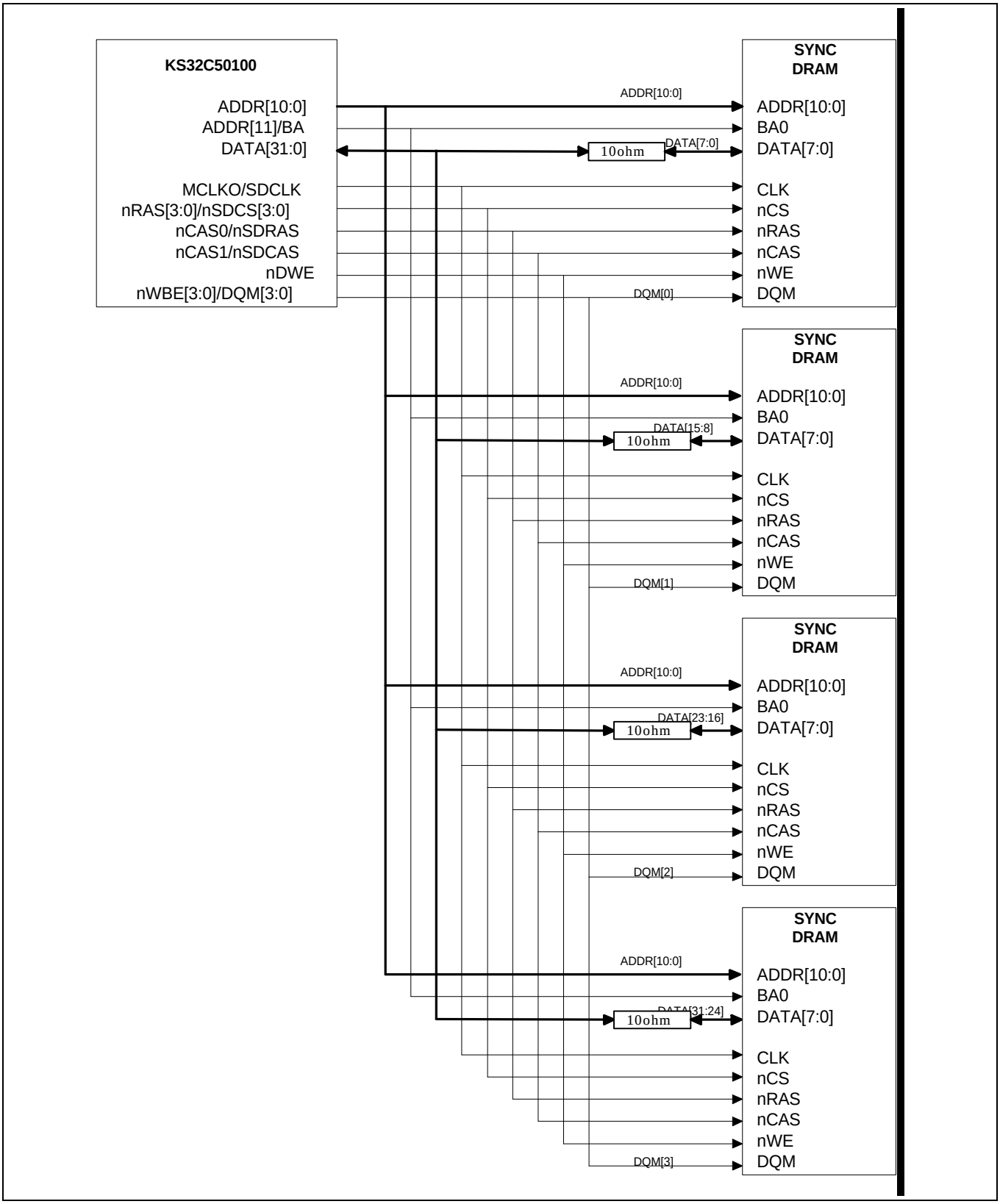


Figure 4-11. SDRAM Design with 1-byte Components

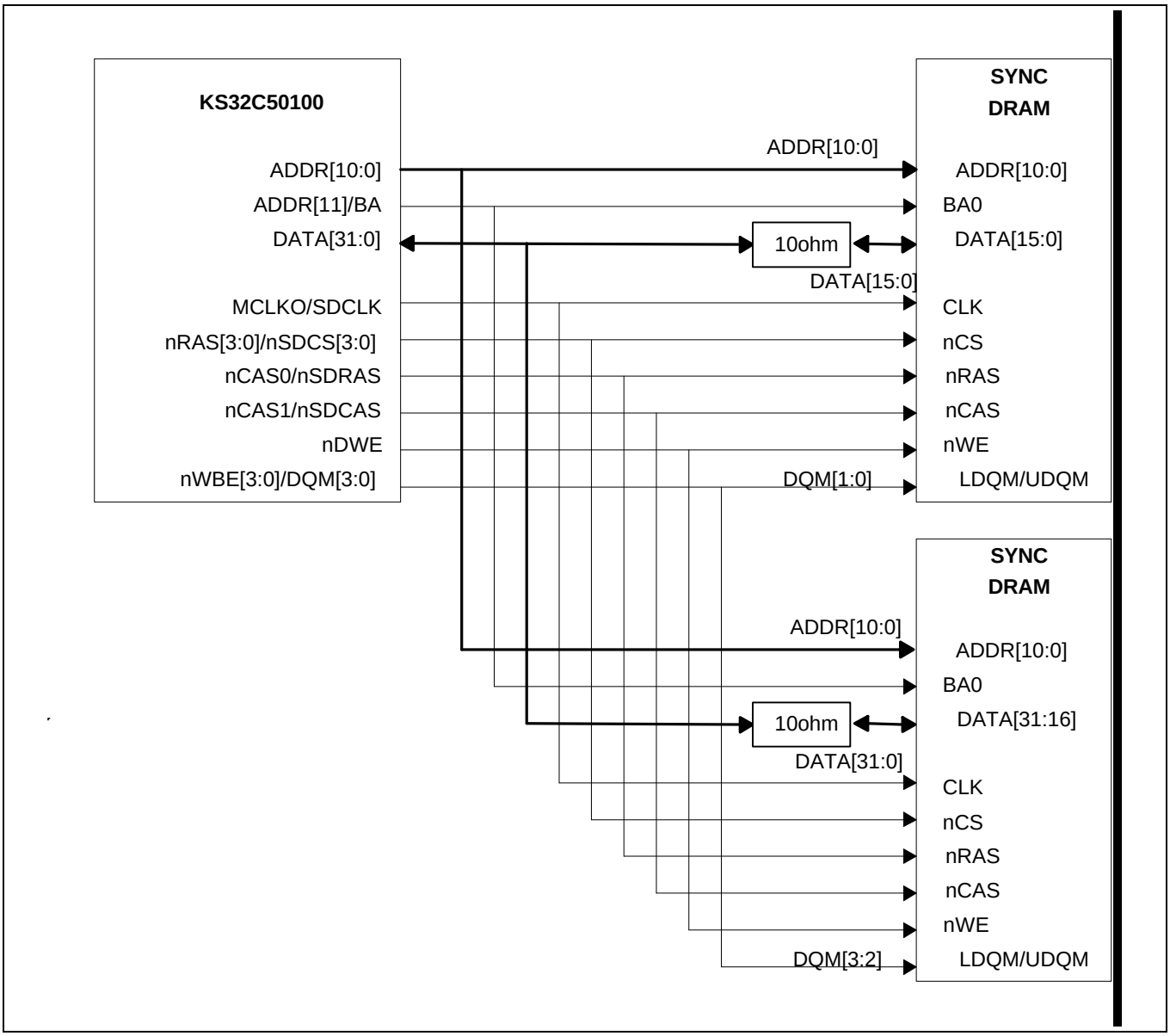


Figure 4-12. SDRAM Design with Half-word Components

EXTERNAL I/O BANKS DESIGN

The external I/O banks 0-3, can have a various width of data bus, and the bus width is controlled by S/W, A EXTDBWTH special register set. A design with external I/O banks 0-3 to memory-like device is similar to ROM/SRAM banks 1-5.

For a very slow device, the KS32C5000(A)/50100 provide external wait request signal (nEWAIT). To use this request signal, you set the tCOS and tCOH should not be zero. and nEWAIT signal should be assert, as soon as possible, the nECS chip select signal is asserted. For detail timing for use of nEWAIT, you can see KS32C5000(A)/50100 USER'S MANUAL, System Manager block description.

The sample design timing diagram with nECS and nEWAIT is deficted in below Figure 4-13.

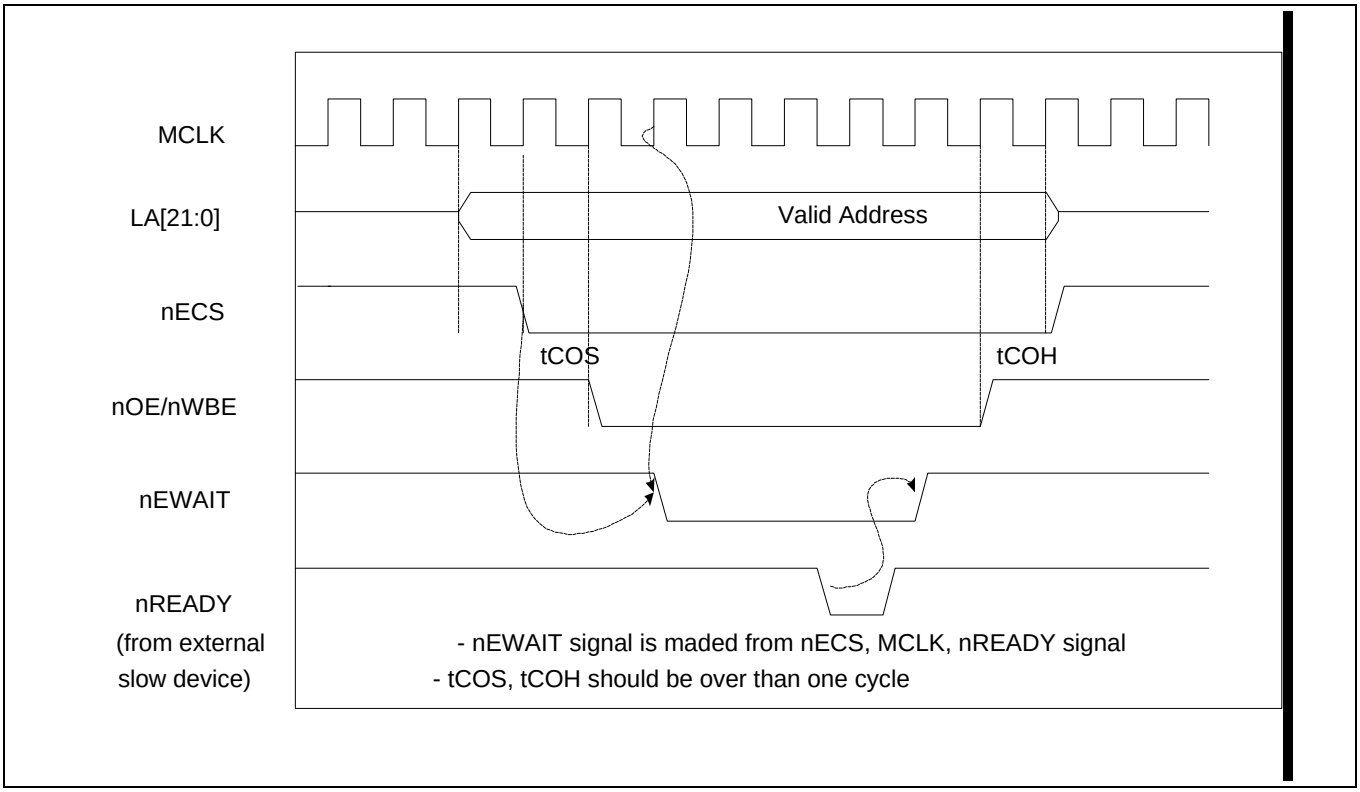


Figure 4-13. External Slow Device Design with nEWAIT Signal

1. Sample Design with External I/O banks: General UART chipset interface

In this design, we use external I/O bank 0, and one byte data bus width.

The external I/O access cycle should be configured as proper value, by software.

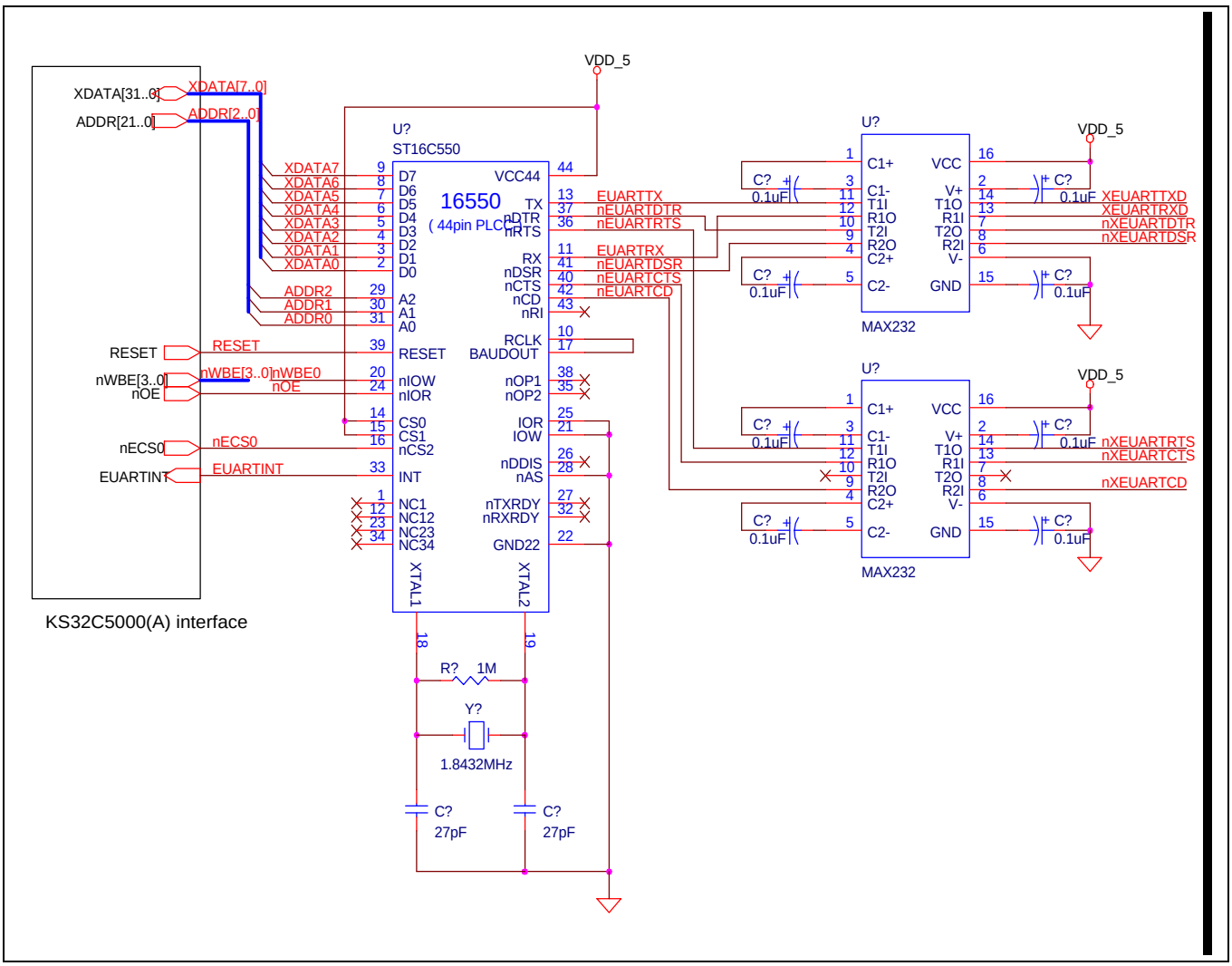


Figure 4-14. External UART Chipset Interface Design

2. Sample Design with External I/O banks: Super-I/O

In this design, we use external I/O bank 0, and one byte data bus width.

The external I/O access cycle should be configured as proper value, by software. And you can refer datasheet of Super-I/O device for more detail operation.

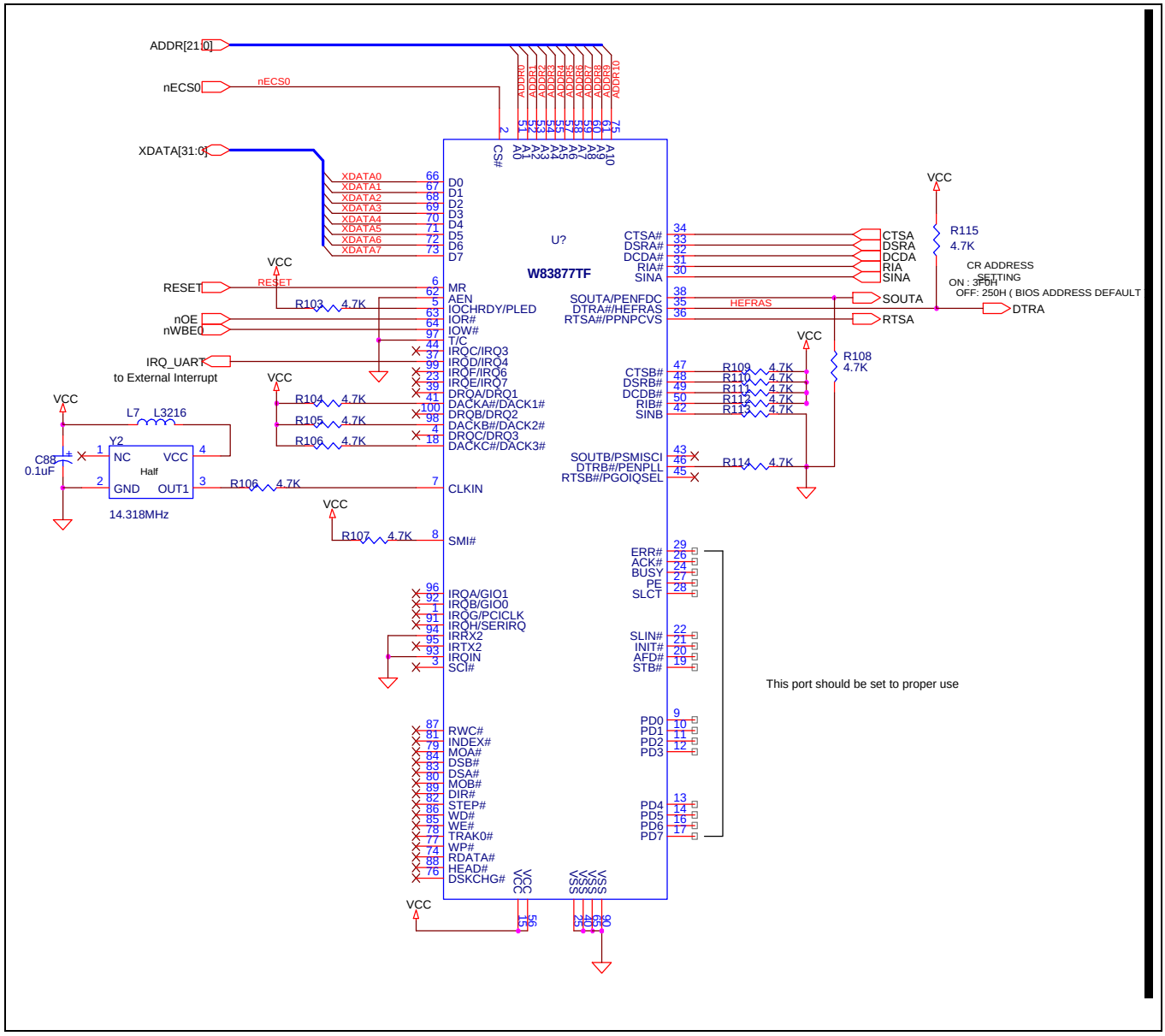


Figure 4-15. Super-I/O Interface Design

3. Sample Design with External I/O banks: ISA bus interface design

The ISA bus has 16 bit data bus width and I/O, memory operation with 8MHz master clock, so we need some kinds of EPLD or CPLD for design ISA bus interface. The access cycle and data bus width should be setted to proper value for generate ISA interface signal.

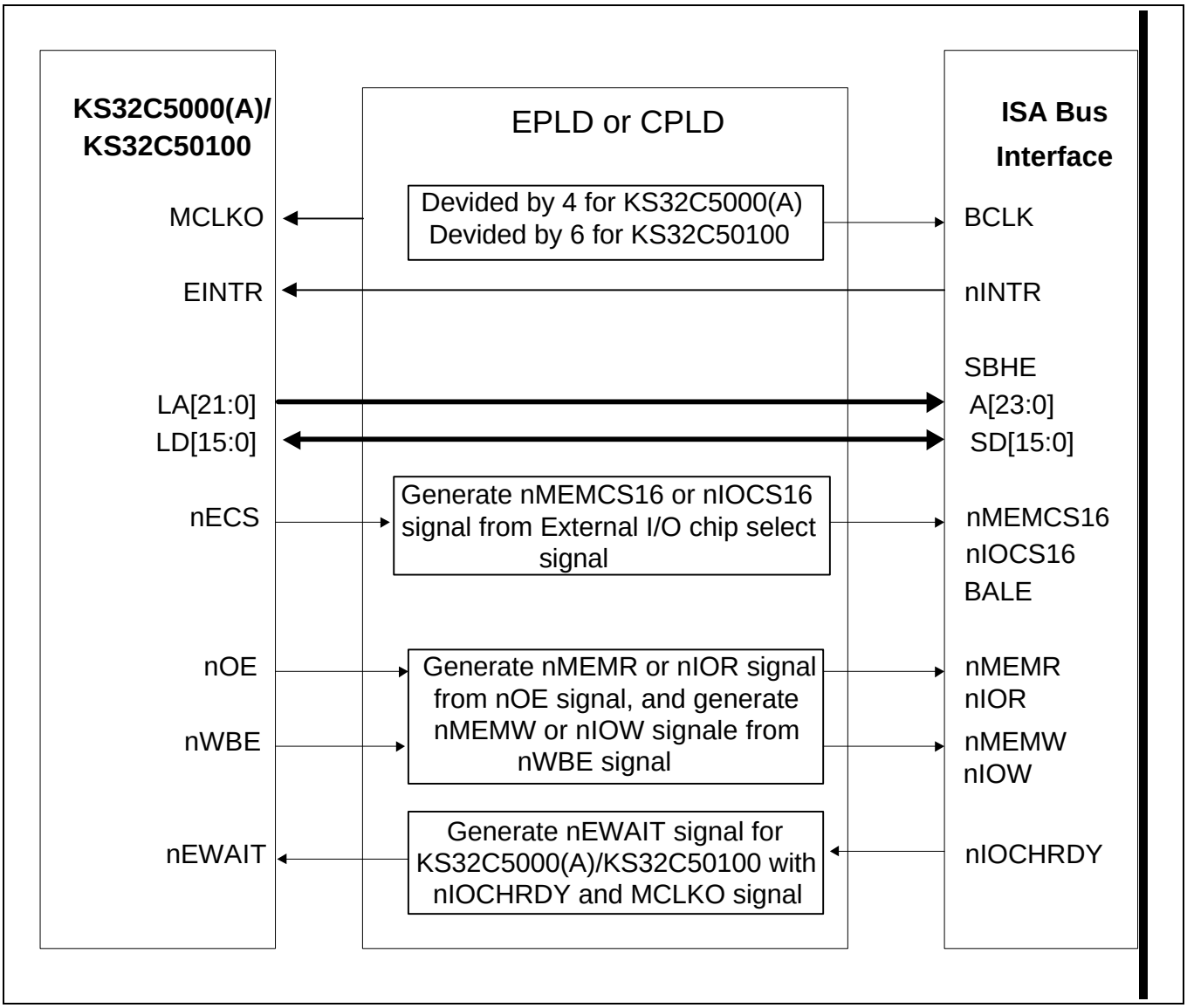


Figure 4-16. ISA Bus Interface Design

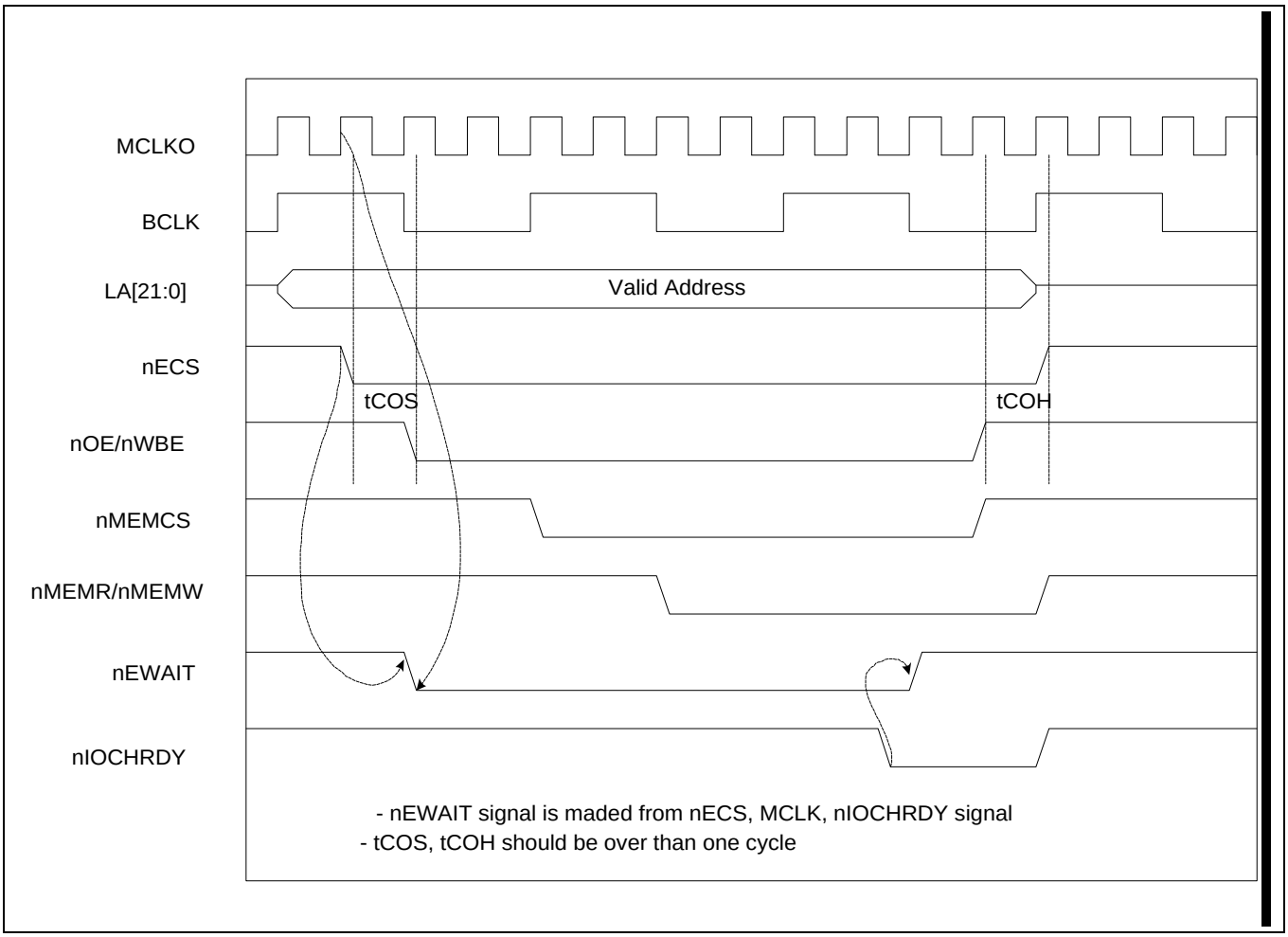


Figure 4-17. ISA Bus Interface Timing Diagram

4. Sample Design with External I/O banks and ROM/SRAM bank: PCI bus interface design

The PCI bus is 32bit synchronous bus, This block diagram is KS32C5000(A)/KS32C50100 with PCI host bridge controller,

The PCI bus general features are

- 32bit Address/Data bus
- High bandwidth (up to 132MB/sec@33MHz)
- Full multi-master capability
- Auto-configuration of devices

The PCI host bridge features are

- Generate configuration cycle
- Generate Interrupt acknowledge cycle
- Arbitration function for all PCI master

When we access PCI bus as PCI host bridge, we have three memory space,

- Memory Space
- I/O Space
- PCI Configuration Space

This three memory space make some consideration, When we design PCI host bridge interface with Samsung' s KS32C5000(A)/KS32C50100, we should be careful, Because KS32C5000(A)/KS32C50100 has total 64MB address range.

This application block diagram for PCI host bridge interface is designed with PLX9080. The Direct Master cycle is more easy to implement than Direct Slave cycle. But you should be careful when you generate address for PCI bus.

The sample block diagram is deficted in Figure 4-18.

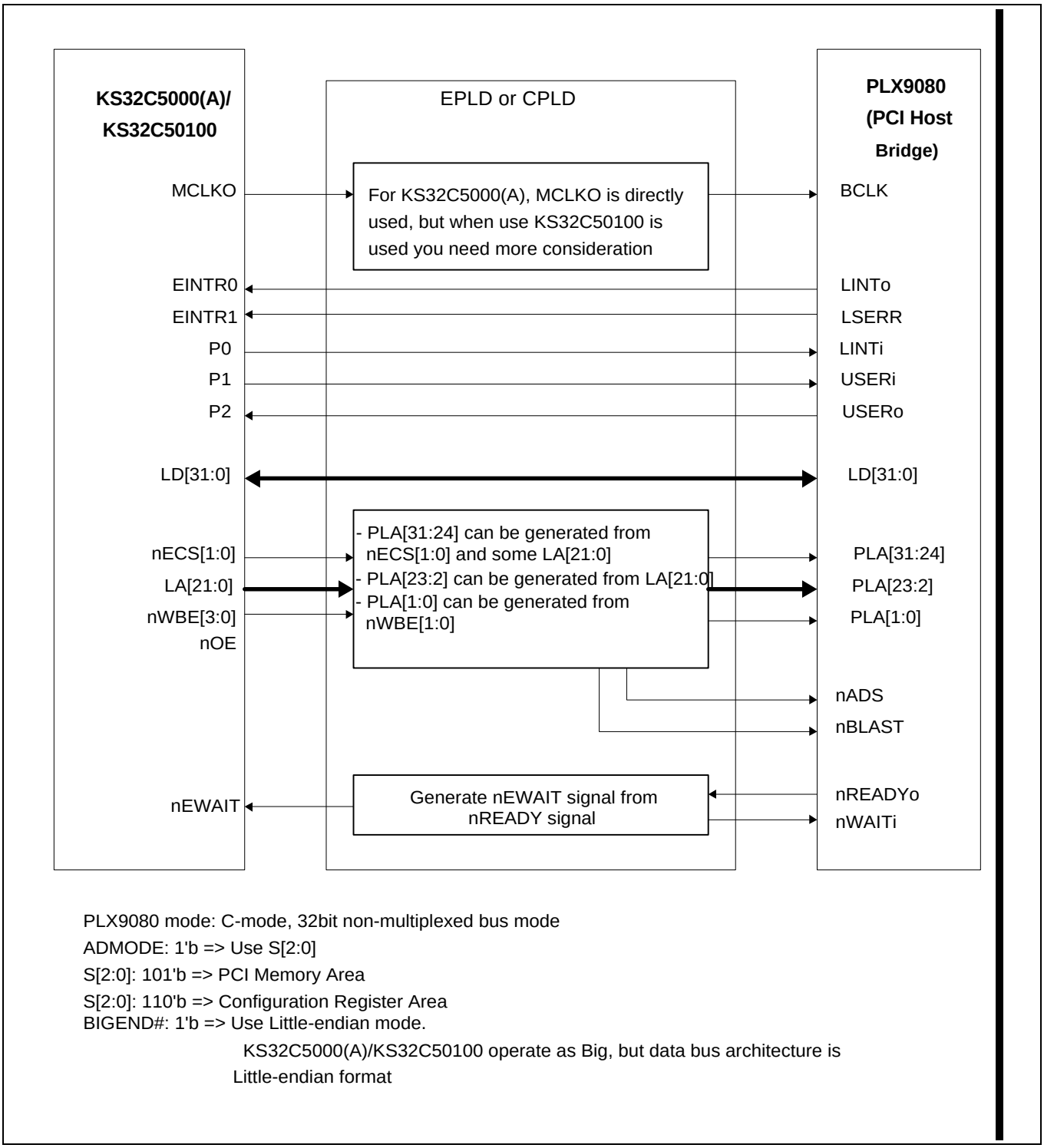


Figure 4-18. PCI bus Direct Master block diagram

The detail timing diagram for Direct master single write/read operation is described in Figure 4-19, Figure 4-20.

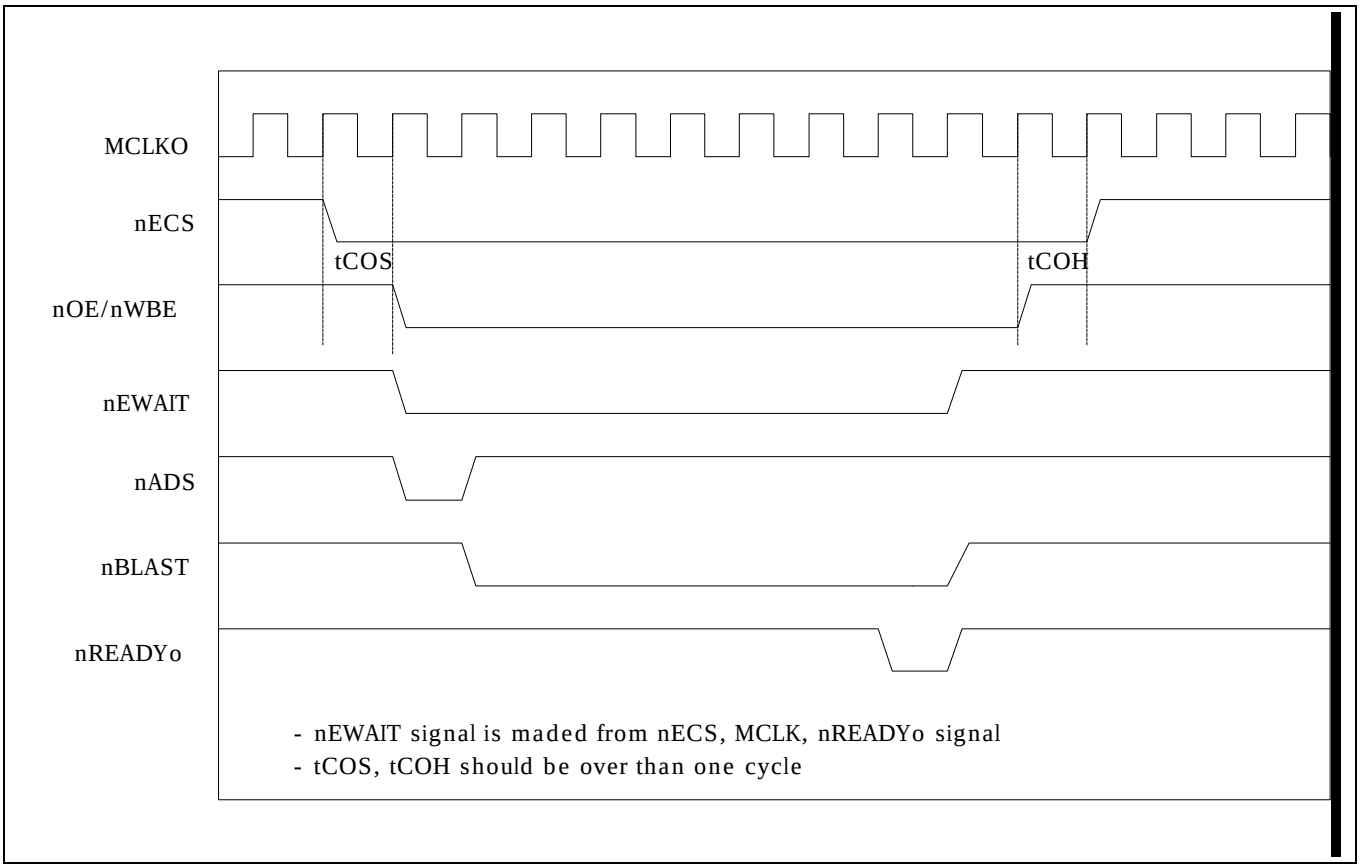


Figure 4-19. PCI bus Direct Master Write timing

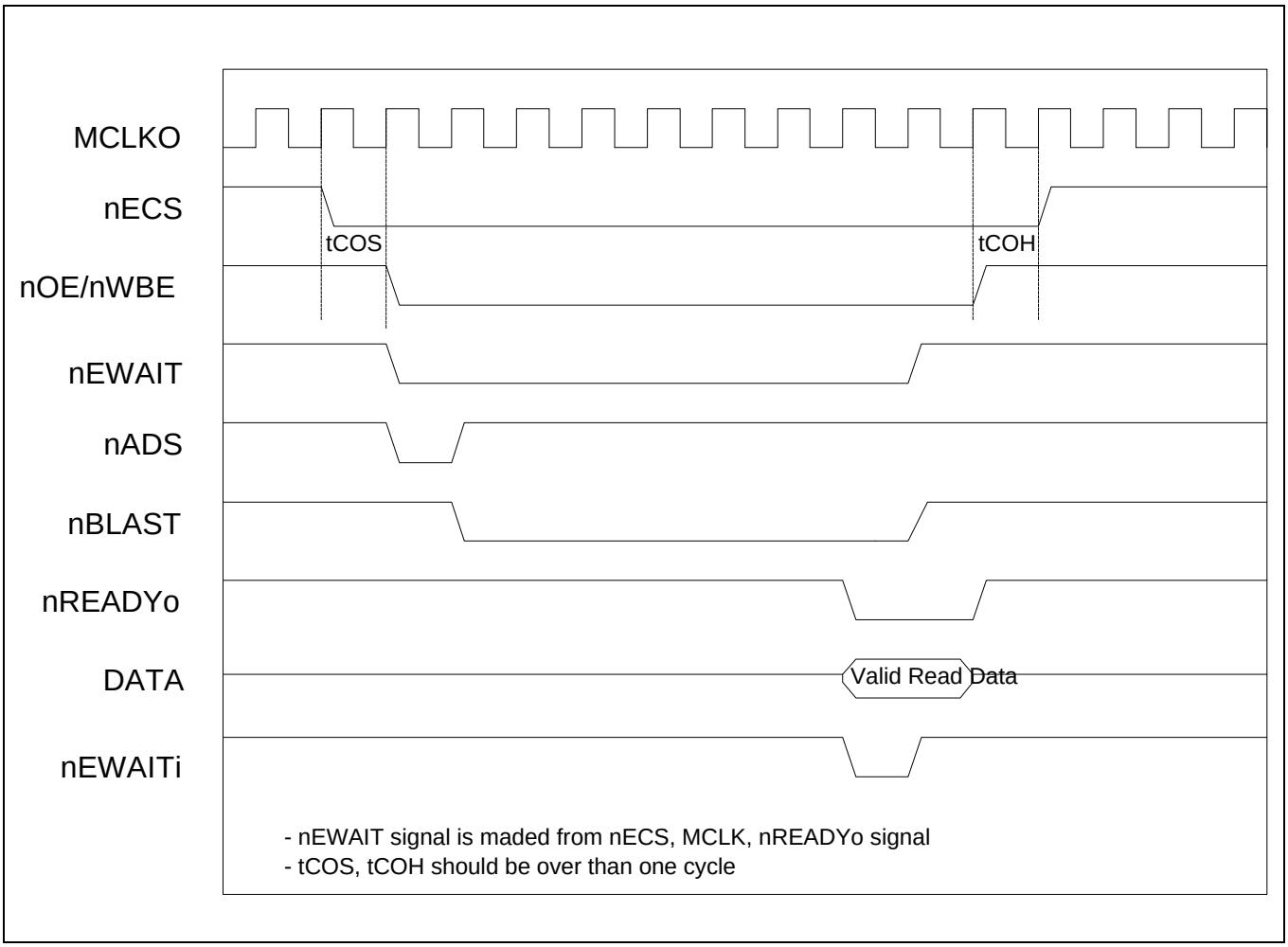


Figure 4-20. PCI bus Direct Master Read Timing

ETHERNET INTERFACE

Interface Overview

The KS32C5000(A)/KS32C50100 ethernet media access controller operates at ether 100 Mbps or 10 Mbps in halh-duplex or full-duplex mode. And the ethernet controller on the KS32C5000(A)/KS32C50100 supports both 10M/100M media independent interface (MII) and old style 10M 7-wire interface.

The KS32C5000(A)/KS32C50100 ethernet interface require physical line interface to connect a network. PHY, a physical line interface chip, is used for interface in a 10M or 100M network. PHY usually supports mediag independent interface(MII) or serial interface.

10M/100Mbps MII Interface

MII (Media Independent Interface) has control, data, and clock signal for the communication between ethernet mediag access controller (MAC) and PHY. And has a serial communication port for manage each PHY, called for Station Management.

The standard MII interface between MAC and PHY is shown in Figure 4-21.

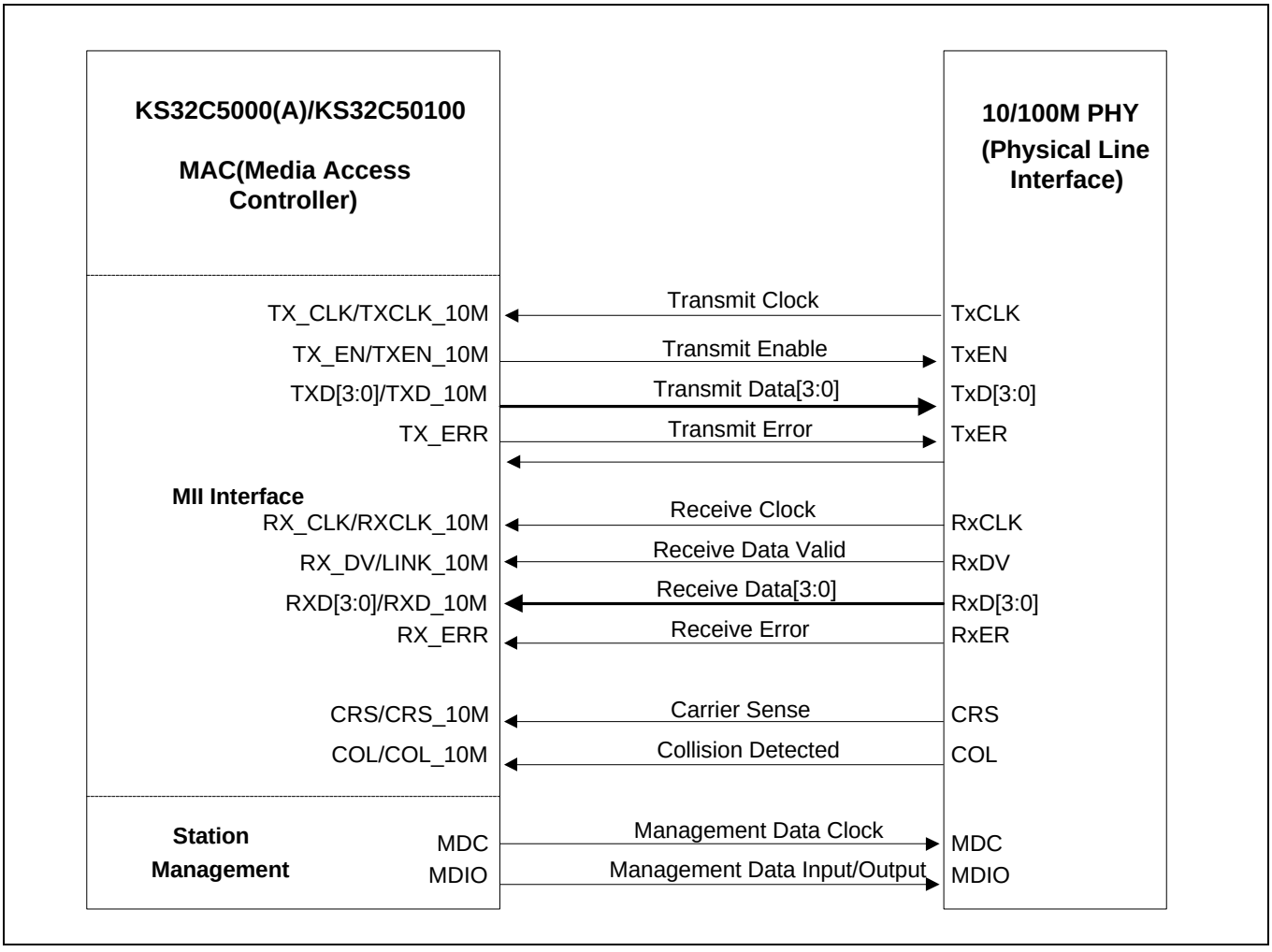


Figure 4-21. MII Connection with PHY

1. Sample Design with 10/100Mbps MII Interface PHY: ICS1890

You can refer ICS1890 datasheet for the detail specification. The configuration of ICS1890 is decided by your system usage.

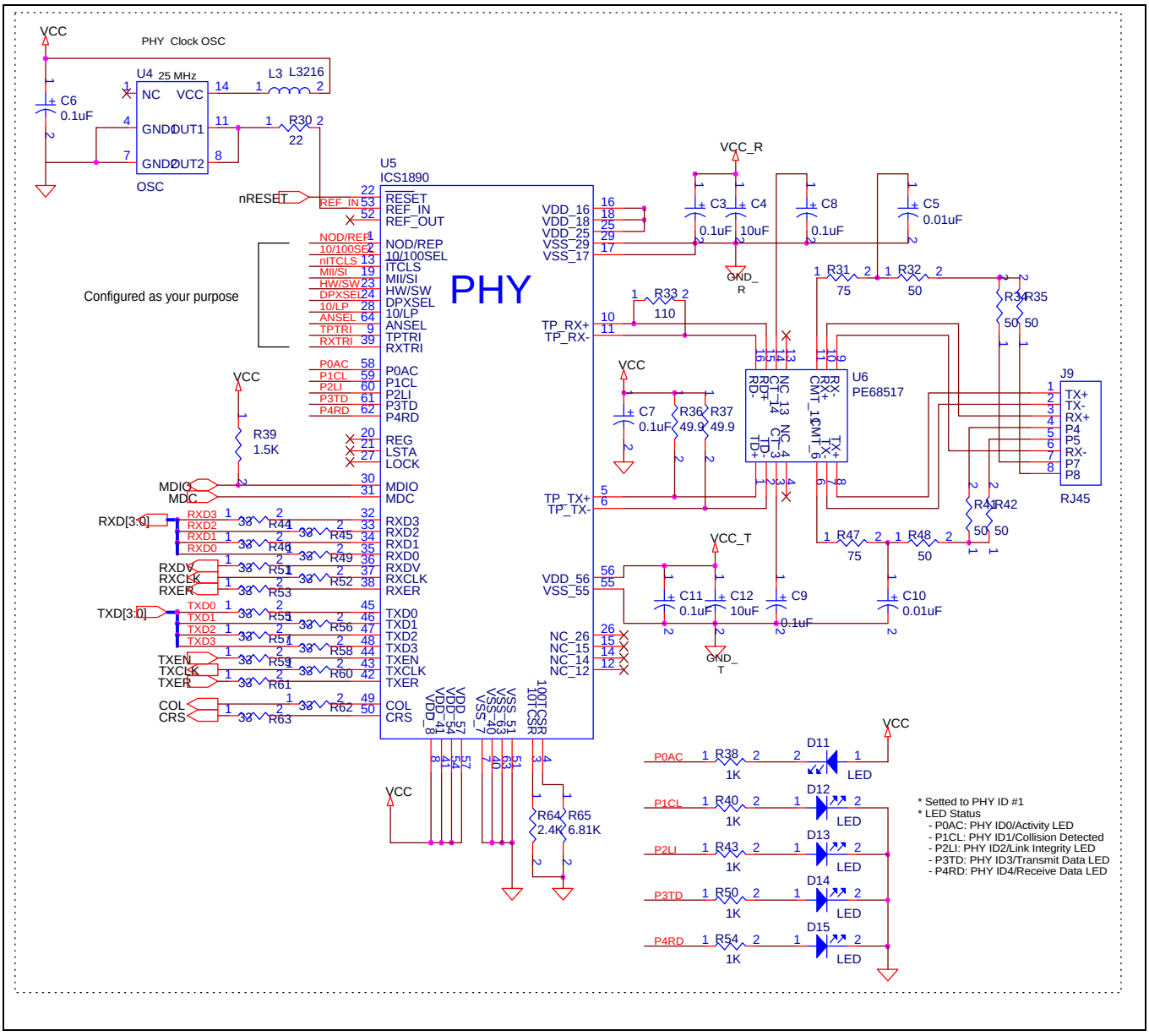


Figure 4-22. Sample Design with ICS1890 PHY

2. Sample Design with 10/100Mbps MII Interface PHY : LXT970

You can refer LXT970 datasheet for the detail specification.

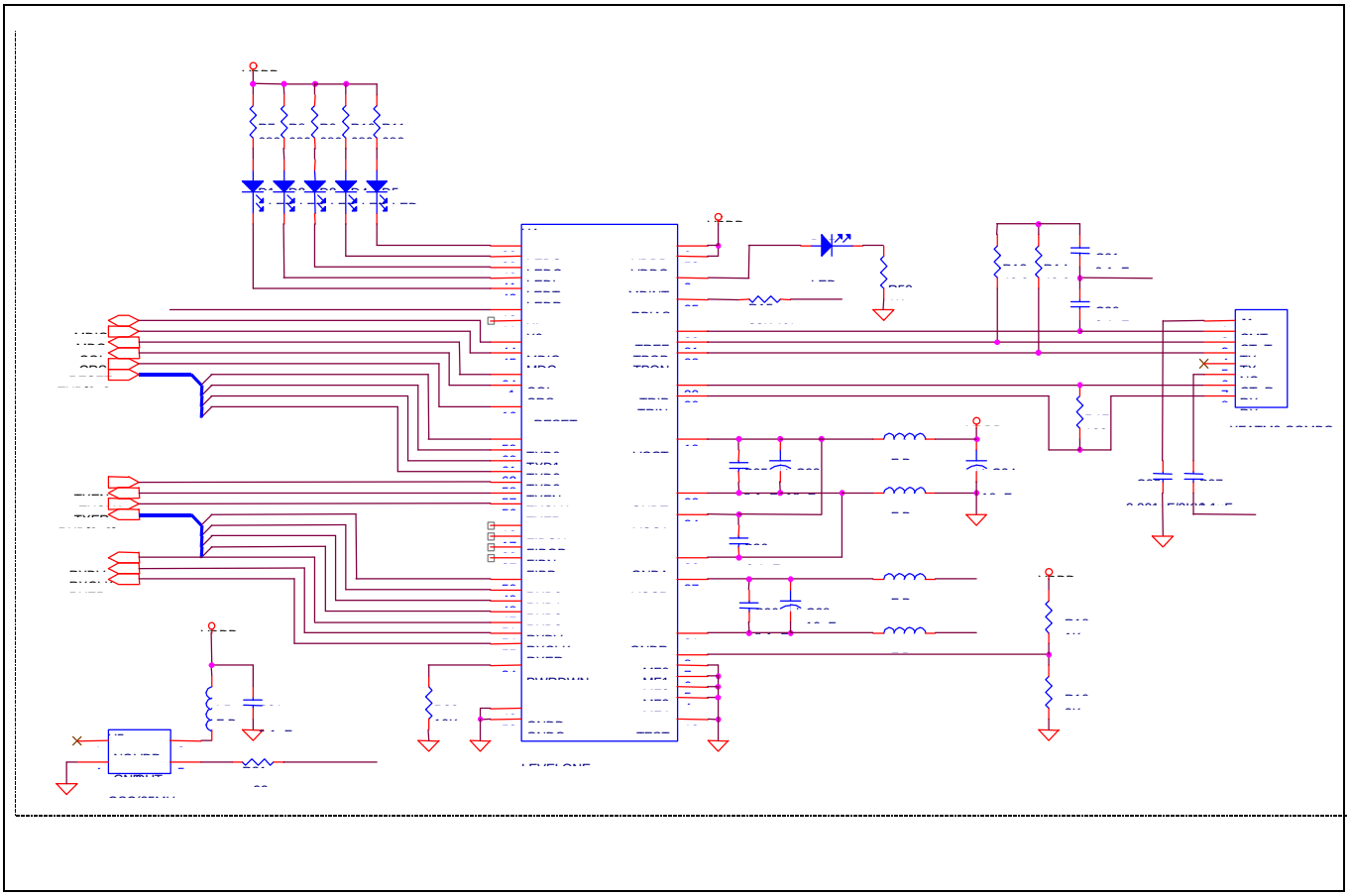


Figure 4-23. Sample Design with LXT970 PHY

3. Sample Design with 10/100Mbps MII Interface to RIC : LXT980

You can refer LXT980 datasheet for the detail specification.

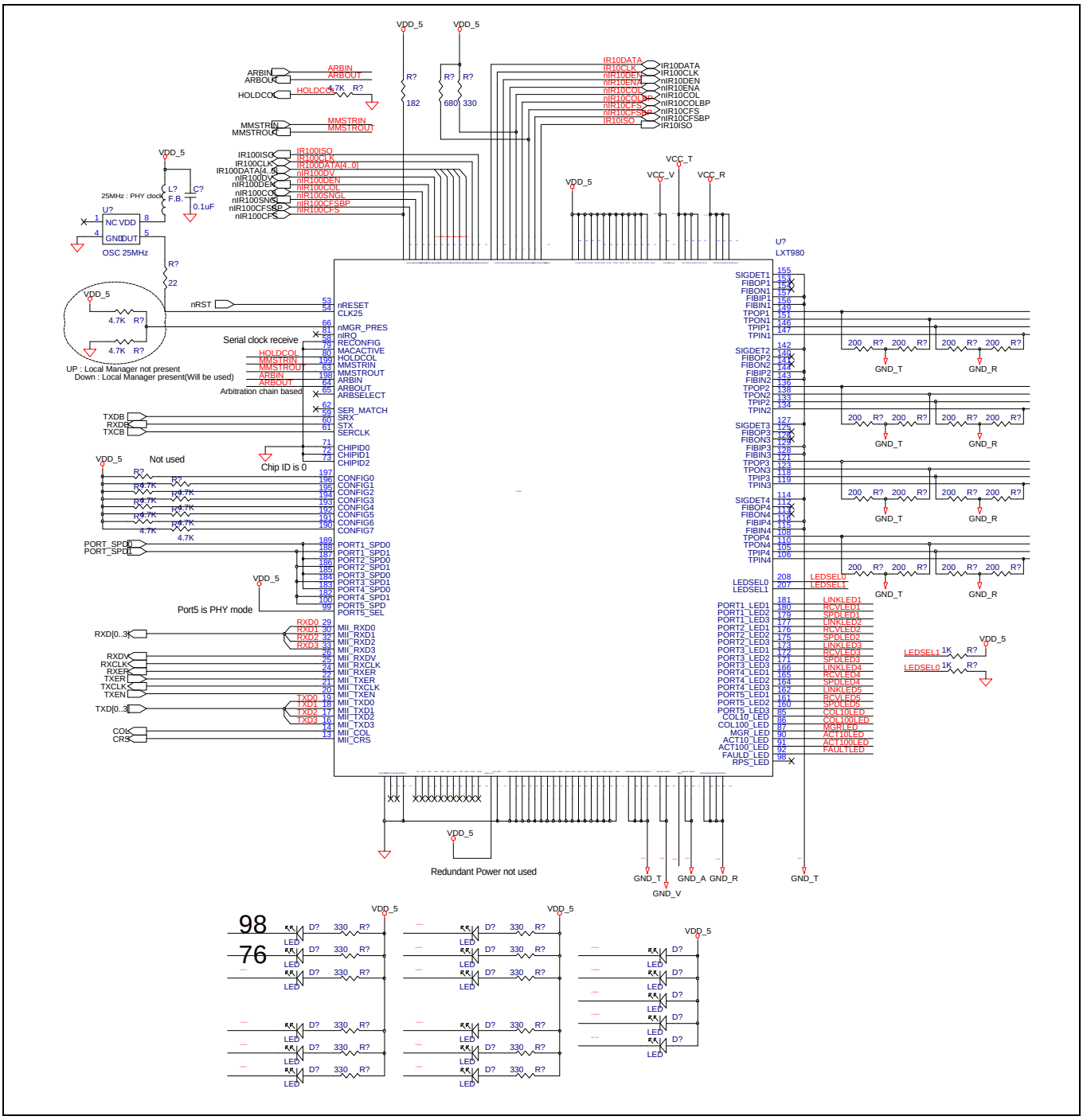


Figure 4-24. Sample Design with LXT980 4-port RIC

10M 7-wire Interface

7-wire interface only for 10M ethernet has control, data, and clock signal for the communication between ethernet mediag access controller (MAC) and PHY.

The standard 7-wire interface between MAC and PHY is shown in Figure 4-25.

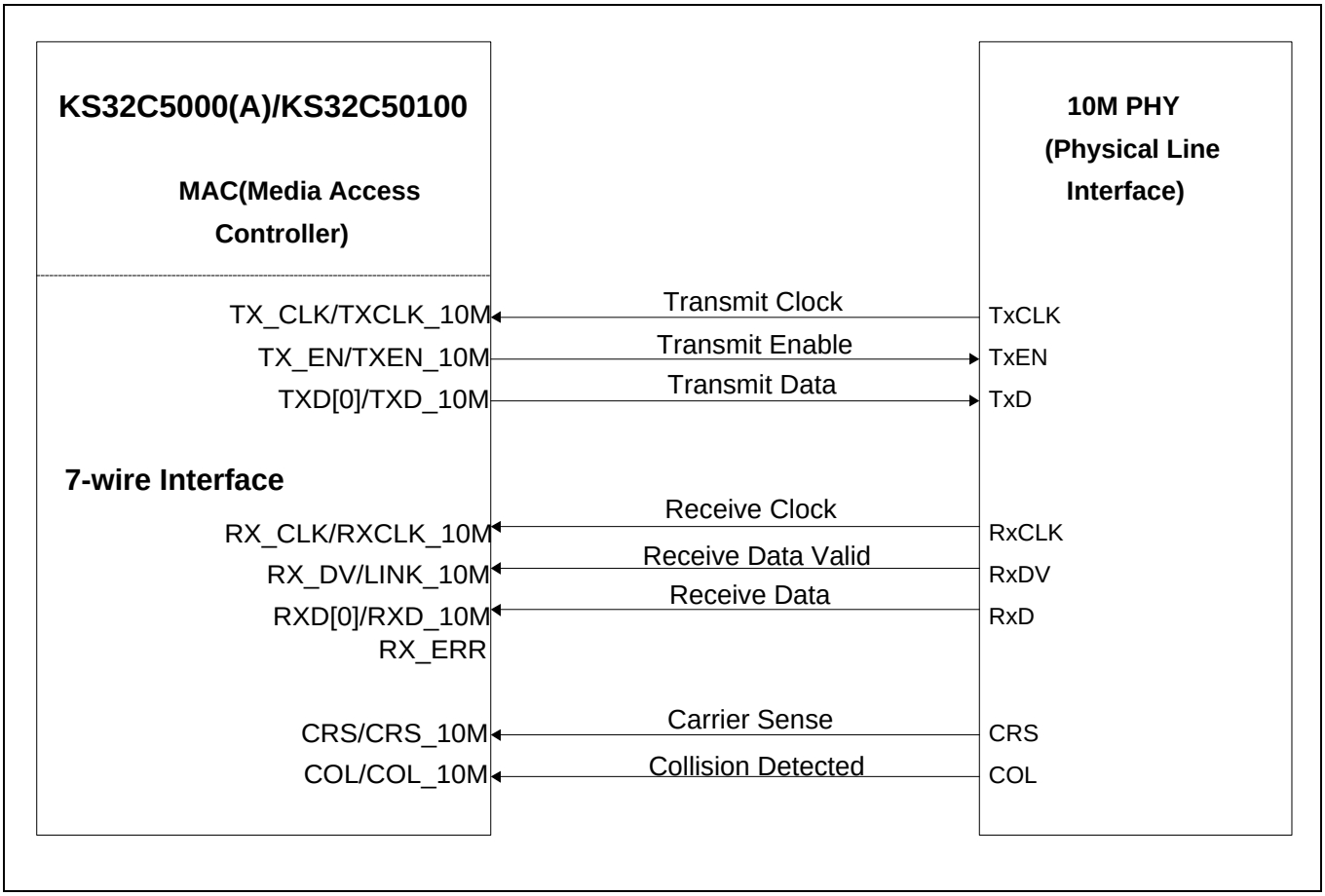


Figure 4-25. 7-Wire Interface with 10M PHY

1. Sample Design with 10Mbps 7-wire Interface: LXT901A/970A

You can refer LXT901/LXT970A datasheet for the detail specification.

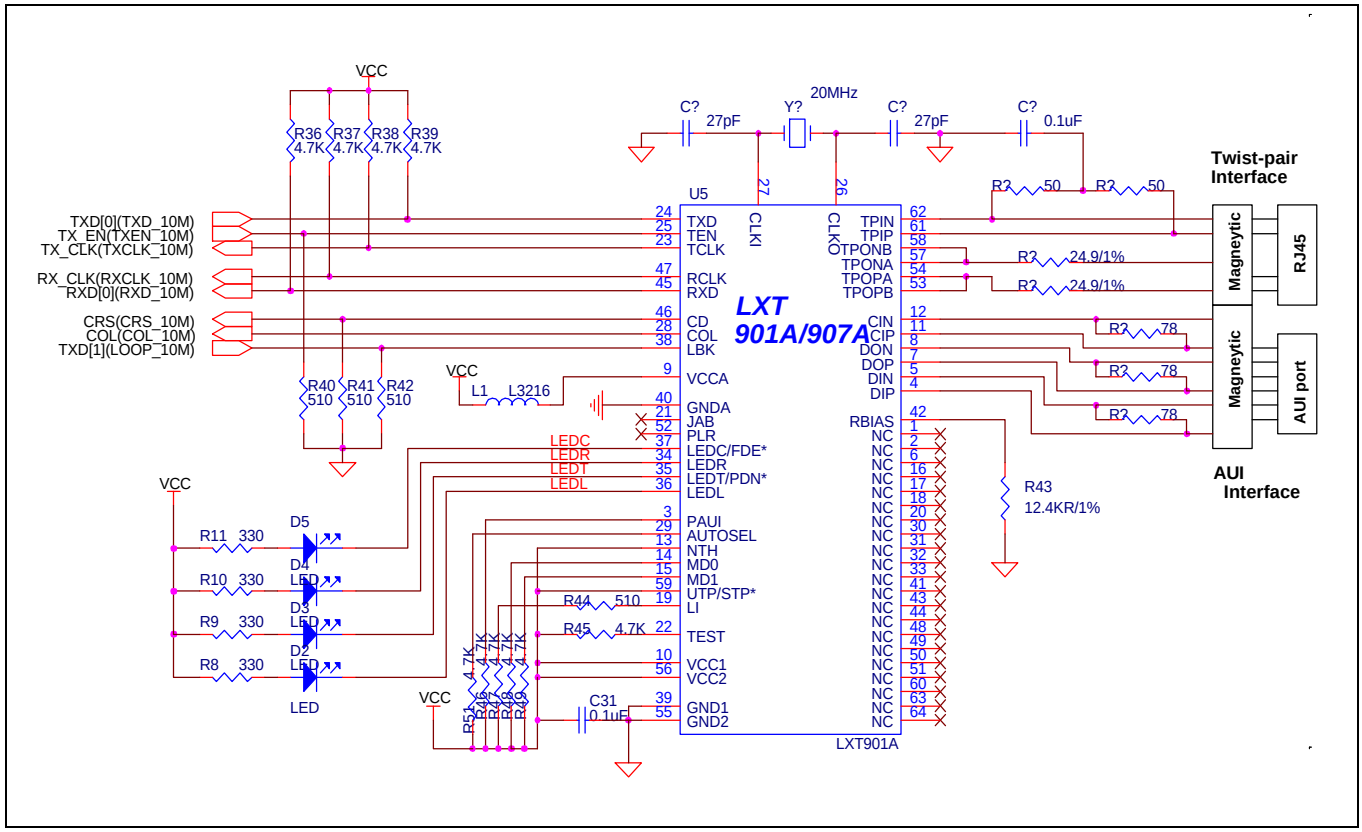


Figure 4-26. Sample Design with LXT901A/970A (Auto Port Select Support)

SYSTEM DESIGN WITH DEBUGGER SUPPORT

EmbeddedICE Macrocell and EmbeddedICE Interface

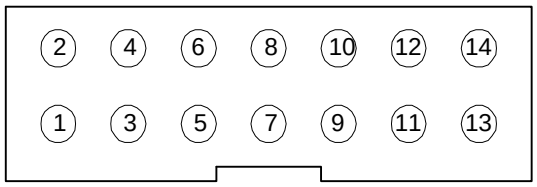
The KS32C5000(A)/KS32C50100 has an EmbeddedICE macrocell that provides debug support fro ARM cores. The EmbeddedICE macrocell is programmed in serial using the TAP(Test Access Port) controller on the KS32C5000(A)/KS32C50100. The EmbeddedICE interface is a JTAG protocol conversion unit. It translates a debug protocol message generated by the debugger into a JTAG signal which is sent to the built-in serial and parallel ports.

JTAG port for EmbeddedICE Interface

When you build a system with the KS32C5000(A)/KS32C50100 EmbeddedICE interface. You should design a JTAG port for EmbeddedICE interface. Usually, the interface connector is a 14-way box header, and this plug is connected to the EmbeddedICE interface module using 14-way IDC cable.

The JTAG port signals, nTRST,TDI,TMS, are internally pulled high, while TDO is internally pulled low which makes pull-up, and pull-down resisters unnecessary.

The pin configuration and a sample design are described in Figure 4-27, 4-28, respectively.



Pin	Name	Function
1,13	SPU	Connected to VDD through 33 or 0 ohm resister*
3	nTRST	Test reset, active low
5	TDI	Test data in
7	TMS	Test mode select
9	TCK	Test clock
11	TDO	Test data out
12	nICERST	Connected to VDD through 10K ohm resister
2,4,6,8,10,14	VSS	System Ground

* 33 ohm for EmbeddedICE, 0 ohm for Multi-ICE

Figure 4-27. EmbeddedICE Interface JTAG Connector

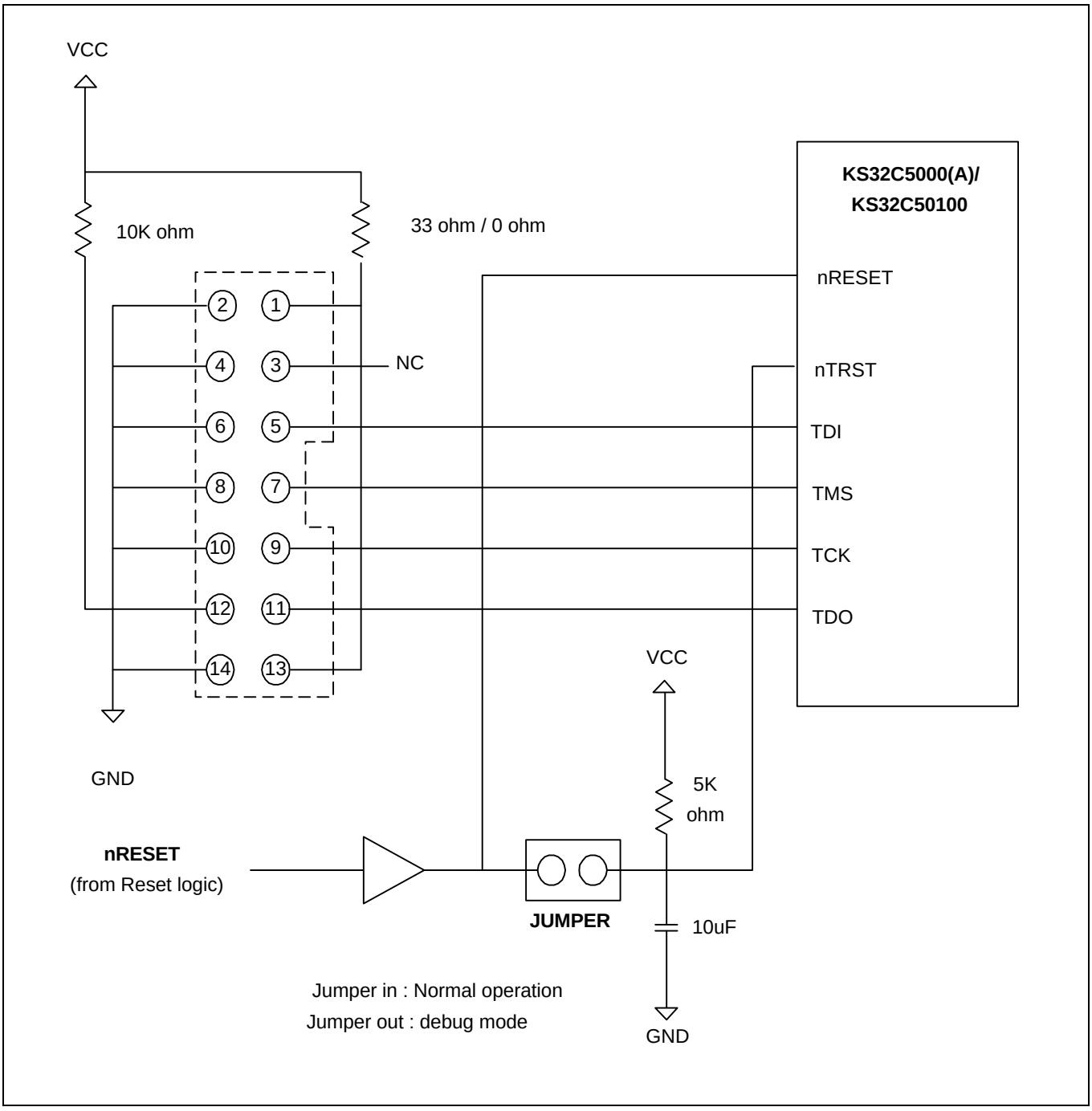


Figure 4-28. EmbeddedICE Interface Design Example

CHECK LIST FOR SYSTEM DESIGN WITH KS32C5000(A)/KS32C50100

When you design a system with the KS32C5000(A)/KS32C50100, you should check a number of items to build a good system. The check list is described below.

- The B0SIZE[1:0] signals are correctly configured to your boot ROM size.
- An nEWAIT signal has a pull-up resister.
- An ExtMREQ signal has a pull-down resister.
- Address bus interface to a memory system, that has a different internal and external address bus.
- There is a JTAG port for debugging.
- The reset logic for debugging interface. specially, nTRST and nRESET pin.
- PLL logic for KS32C50100

NOTES