MOTOROLA

DragonBall Operation

Order this document by
TBD

Date: 11/12/98

Preliminary

Application Note

16bit SRAM Interface

INTRODUCTION

This note describes the method of interfacing a 16bit SRAM to DragonBall-EZ (MC68EZ328). The interface is simply constructed by discrete logic gates.

In this note, the EZ68328ADS was used as a development platform and the TOSHIBA TC554161FTL-85 16bit Static Ram was used as an example. The TC554161FTL is 4,194,304 bits static random access memory organized as 262,144 words by 16 bits.

HARDWARE

Figure 1 shows a functional block diagram of the 16 bit SRAM interface.

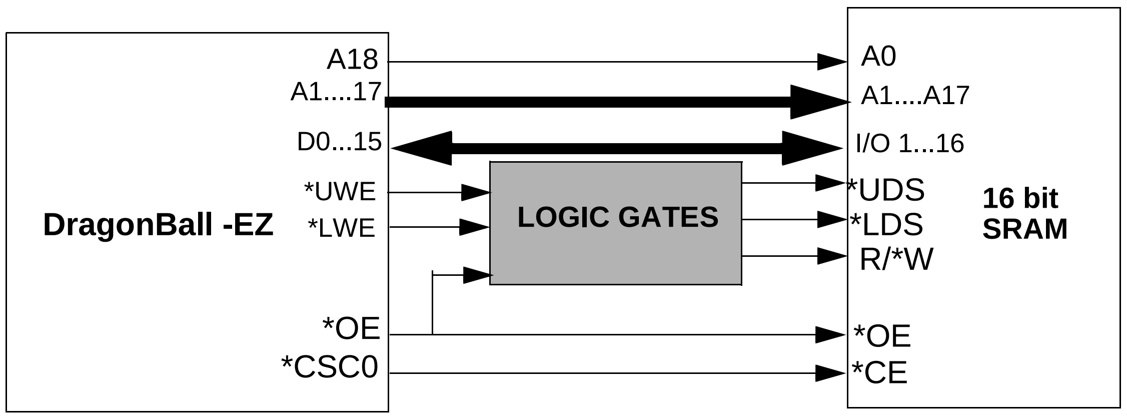


Figure 1. Block Diagram of the 16 bit SRAM interface

The logic implemented between the DragonBall -EZ and 16 bit SRAM is shown as following tables:

*UWE	*LWE	***UDS**	
0	0	**0**	***Write word***
0	1	**0**	***Write Upper Byte***
1	0	**1**	***Write Lower Byte***
1	1	**0**	***Read***

*Table 1. Logic table for *UDS*

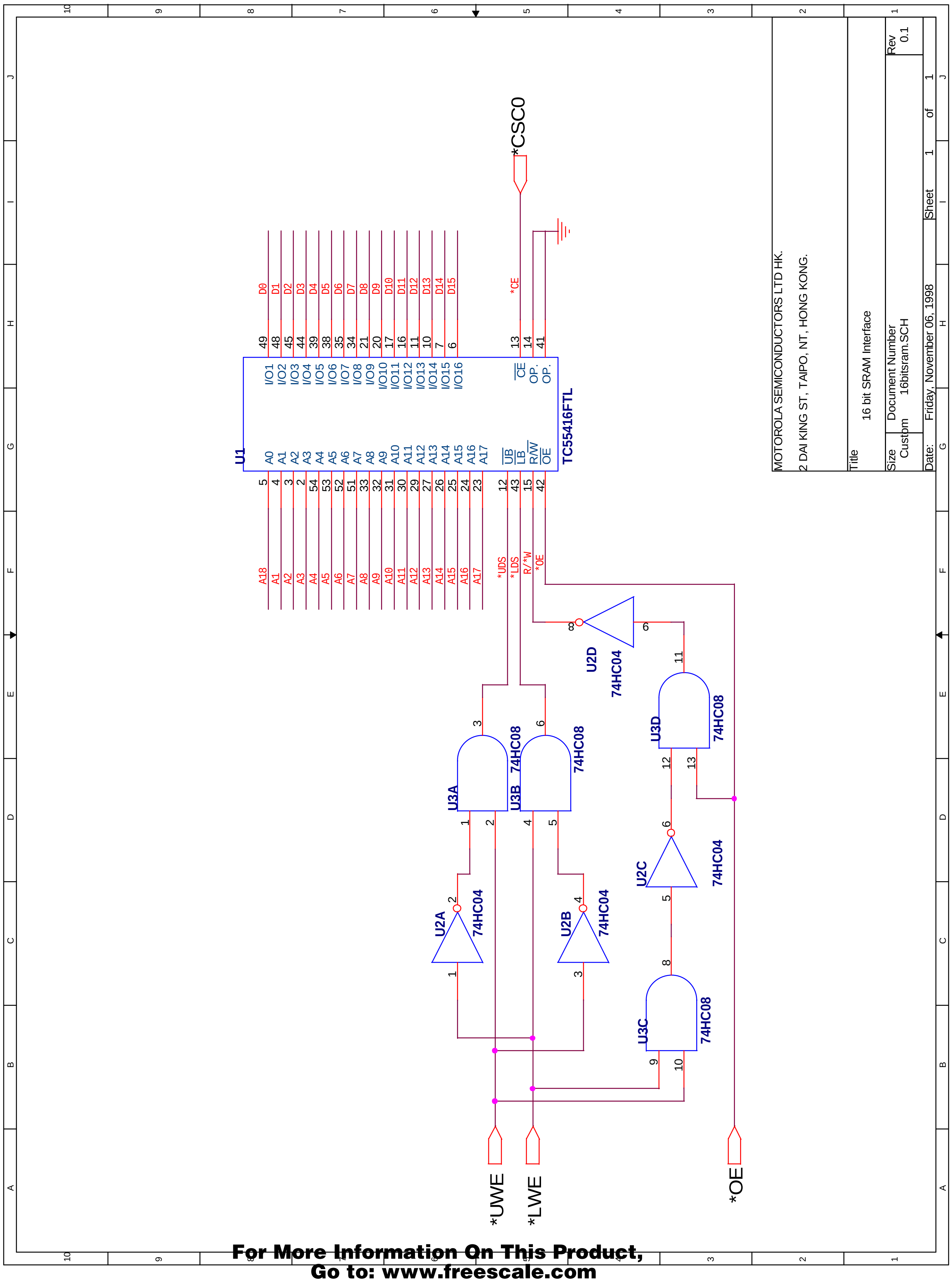
*UWE	*LWE	***LDS**	
0	0	**0**	***Write word***
0	1	**1**	***Write Upper Byte***
1	0	**0**	***Write Lower Byte***
1	1	**0**	***Read***

*Table 2. Logic table for *LDS*

*UWE	*LWE	*OE	**R/*W**
0	0	0	**1**
0	1	0	**1**
1	0	0	**1**
1	1	0	**1**
0	0	1	**0**
0	1	1	**0**
1	0	1	**0**
1	1	1	**1**

*Table 3. Logic table for R/*W*

4 AND gates and 4 NOT gates are used to implement the above logic.
The following schematic shows how to use one 74HC04 and one 74HC08 to implement the logic.


*CSC0
*CE
D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 D10 D11 D12 D13 D14 D15
I/O1 I/O2 I/O3 I/O4 I/O5 I/O6 I/O7 I/O8 I/O9 I/O10 I/O11 I/O12 I/O13 I/O14 I/O15 I/O16
CE OP. OP.
U1
TC55416FTL
A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12 A13 A14 A15 A16 A17
UB LB R/W OE
A18 A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12 A13 A14 A15 A16 A17
*UDS *LDS R/*W *OE
U2A 74HC04
U2B 74HC04
U2C 74HC04
U2D 74HC04
U3A 74HC08
U3B 74HC08
U3C 74HC08
U3D 74HC08
*UWE
*LWE
*OE
MOTOROLA SEMICONDUCTORS LTD HK.
2 DAI KING ST, TAIPO, NT, HONG KONG.
Title
16 bit SRAM Interface
Size Custom
Document Number 16bitsram.SCH
Rev 0.1
Date: Friday, November 06, 1998
Sheet 1 of 1

Note:

Form the schematics and block diagram, we can see that A0 from the SRAM is connected to A18 from the DragonBallEZ. This connection is because of word addressing in 16 bit SRAM.

	D15.....D8	D7.........D0
.......00	ab	cd
........02	....	....
........04	....	....
........06	....	....
	....	....

DragonBall-EZ

	D15.............D0
0......00	abcd
0.......01	....
0.......02	....
0.......03	....
0.......04	....

16 bit SRAM

As shown in the above, the addressing in DragonBall-EZ and 16 bit SRAM is different. Connecting A0 to A18 can solve this problem. By connecting A0 to A18, the address is always even in 16 bit SRAM. Upper Byte Write and Lower Byte Write only depends on the *UDS and *LDS. When write to lower Byte, *LDS asserts while *UDS asserts when writing upper byte. For the read operation, *UDS and *LDS always assert since reading Lower Byte or Upper Byte depends on DragonBall-EZ only. DragonBall can select to read D0-D7 or D8-D15 itself.

Let us consider the following example, after connecting A0 to A18, the addressing in SRAM becomes:

	D15.............D0
0.......00	abcd
0.......02	....
0.......04	....
0.......06	....
....	....

16 bit SRAM

As shown in the above table, there is no odd address, therefore the address line A0-A17 in SRAM is the same when accessing upper byte (0xab)and lower byte(0xcd). But the difference is when write to upper byte *UDS asserts while *LDS asserts when write to lower byte.

INITIALIZATION

Finally here is the source code for initialization.

SOURCE CODE

```
***************************************
*
*This program is to initialize 16 bit SRAM
*
***************************************

M328_GRPBASEC      EQU    $fff104
M328_CSC           EQU    $fff114
M328_CSD           EQU    $fff116
M328_PCSEL         EQU    $fff40B
M328_DRAMCONT      EQU    $fffc02
M328_EXTWE         EQU    $fff150

      SECTION code
START:

      MOVE.W #$400,M328_GRPBASEC    ;Set CSB Group Base Address
      MOVE.W #$89,M328_CSC          ;Set Chip Select C Register
      MOVE.W #$9d,M328_CSD          ;Disable DRAM
      ANDI.B #$7f,M328_DRAMCONT     ;Disable DRAM controller
      ANDI.B #$fb,M328_PCSEL        ;Enable Chip Select CSC0
      ANDI.B #$7f,M328_EXTWE        ;Extend *LWE and *UWE

      END
```